Università degli studi di
Modena e Reggio Emilia

Dipartimento di Scienze
Fisiche, Informatiche, Matematiche

Tesi di Laurea in

Informatica

Progettazione, realizzazione ed accessibilità di un Database biomolecolare sulle sequenze ultraconservate del genoma umano

Relatore

Prof. Riccardo Martoglia

Correlatori

Prof. Federica Mandreoli
Dott. Cristian Taccioli

Candidato

Vincenzo Lomonaco

Anno Accademico 2012/2013

Università degli studi di
Modena e Reggio Emilia

Dipartimento di Scienze
Fisiche, Informatiche, Matematiche

Tesi di Laurea in

Informatica

Progettazione, realizzazione ed accessibilità di un Database biomolecolare sulle sequenze ultraconservate del genoma umano

Relatore

Prof. Riccardo Martoglia

Correlatori

Prof. Federica Mandreoli
Dott. Cristian Taccioli

Candidato

Vincenzo Lomonaco

Anno Accademico 2012/2013

Dedicata a Elisabetta Giubilato e Francesco Lomonaco,
dall'amore sincero e riconoscente di un figlio
verso i propri genitori.

Ringrazio tutti coloro i quali mi sono stati vicino o mi hanno supportato durante gli studi, arricchendone inevitabilmente il percorso.
Ringraziamenti particolari vanno al Relatore Prof. Martoglia ed ai Correlatori Prof. Mandreoli e Dott. Taccioli, per la correttezza, responsabilità e capacità con la quale hanno condotto il loro ruolo.

Introduzione

L'emergere di computer ad alte prestazioni a basso costo e le tecnologie omiche ad alto throughput sono alla base di diversi progetti finalizzati alla costruzione di ***database bio-tecnologici****, quali* ***RCSB Protein Data Bank****,* ***Gene Ontology****,* ***CMap****,* ***GEO NCBI****. Attualmente, tali banche dati contengono informazioni di vitale importanza e in una grande varietà di forme (sequenze genetiche, annotazioni testuali, dati in forma di grafo, etc) ma possono essere interrogate solo in modo indipendente e incompleto, rendendo estremamente difficile per un ricercatore poter rilevare anche le relazioni più elementari tra i dati.* [1]

La tesi di ricerca, dunque, è volta a soddisfare un nuovo paradigma di studio, progettazione e sviluppo di tecniche che consentano l'interoperabilità delle banche dati biologiche, dirette, inoltre, a sostenere modalità efficaci ed efficenti per la loro interrogazione.

Nello specifico essa ruota attorno la progettazione e realizzazione di una banca dati sulle *sequenze nucleotidiche ultraconservate*[1] del genoma umano e dati biologici relativi già esistenti.
Nella voluta risoluzione di noti problemi d'indagine come la sin troppo accentuata entropia delle informazioni inerenti, il progetto offre, dunque, innovativi *environments* d'investigazione ed interrogazione dei dati che rappresentano, sotto certi punti di vista, un'avanguardia nel mondo biologico della ricerca.
Basata su soluzioni portabili ed efficenti per il recupero di dati, come su sorprendenti compromessi di semplicità e capacità espressiva nell'interrogazione e rappresentazione degli stessi, la tesi di ricerca, infatti, è volta a segnare con decisione un nuovo traguardo nel campo delle sequenze nucleotidiche ultraconservate sia in termini di interrogabilità dei dati quanto di integrità e completezza.

[1] vd. Appendice **Elementi ultraconservati nel genoma umano**

Premessa

Si tiene a precisare che lo scritto rappresenta il risultato di un lavoro frutto di sudate sfide ed intuizioni, demoralizzazioni e rallegramenti, nella collaborazione di due mondi e discipline, quali l'Informatica e la Biologia, così diverse ma così intrensicamente legate. Sia il lettore, dunque, così gentile da superare, in virtù di quanto pocanzi espresso, eventuali forme di imprecisione o imperfezioni sfuggite ora all'uno ora all'altro punto di vista.

Obiettivi della Tesi

Entrando più nel dettaglio, la tesi può essere articolata attorno alla realizzazione di quattro obiettivi principali:

- **Recupero dei dati** d'interesse online mediante *Web Services*[2] e loro razionalizzazione e filtraggio in locale.
- **Studio**, **progettazione** e **Realizzazione** di una base di dati, o meglio degli *script*[3] SQL [2] portabili per la realizzazione di una base di dati sulle sequenze nucleotidiche ultraconservate del genoma umano.
- **Recupero e collegamento di un ontologia biomedica** sulle patologie umane per fornire una più completa gamma di possibilità nelle operazioni di *Data Mining*[4]
- **Studio di accessibilità ed interrogabilità** dei dati **mediante un portale web** che supporti diverse modalità di investigazione e visualizzazione e **relativa realizzazione**.

Nel corso dell'elaborato ogni obiettivo verrà approfondito con accuratezza seguendo l'ordine temporale di sviluppo come rammentato nella sezione seguente: **Struttura della Tesi**.
Verranno inoltre evidenziati, con meticolosa attenzione ai particolari, gli strumenti *Open Source*[5] utilizzati ed attraverso quale modalità. Verranno, altresì sottolineati punti di debolezza e le relative e possibili migliorie effettuabili.

[2]Sistema software progettato per supportare l'interoperabilità tra diversi elaboratori attraverso il web.
[3]In informatica, il termine script designa un tipo particolare di programma contrasseganto da particolari caratteristiche di linearità e semplicità.
[4]Insieme di tecniche e metodologie che hanno per oggetto l'estrazione di un sapere o di una conoscenza a partire da grandi quantità di dati.
[5]Termine inglese che significa codice sorgente aperto.

Struttura della tesi

Se da un lato, nel corso dell'elaborato, si sviluppano gli obiettivi enucleati logicamente nel paragrafo precedente, dall'altro l'organizzazione dei capitoli sottolinea maggiormente una suddivisione di tipo *temporale* per efatizzare con maggior rilievo il tortuoso ed affascinante percorso effettuato per giungere ad una tesi di ricerca completa.

Capitolo 1 - Background. Lo stato dell'arte. Si risponderà ad alcune semplici domande: Quali sono i dati di partenza? Quali sono le tecnologie adoperate per interrogarli? Con quali risultati? È opportuno un'operazione di miglioria? In quali termini?

Capitolo 2 - Sviluppo. Lo svilluppo centrale della Tesi. Metodi, tecniche e sistemi sono valutati ed eventualmente progettati in questo capitolo: dai dati grezzi ai dati in forma normale, dalle problematiche di ridondanza a quelle di linking, dall'analisi visuale manuale a quella avanzata. Dall'analisi e progettazione si passa, poi, alla realizzazione concreta con le relative problematiche, le scelte di campo e nuove possibilità.

Capitolo 3 - Risultati. Presentazione dei risultati. Il prodotto finito e le ultime modifiche, eventuali rifiniture, possibilità di utilizzo.

Capitolo 4 - Conclusioni e sviluppi futuri. Conclusioni e possibili migliorie. Difetti e valutazioni tecnico-personali.

Appendice A - Glossario di base di biologia molecolare. Informazioni e termini di base riguardo il background biologico essenziale per affrontare con linearità l'argomentazione della tesi.

Appendice B - Elementi ultraconservati nel genoma umano. Dettaglio tecnico biologico sugli elementi ultraconservati e nello specifico sulle sequenze nucleotidiche ultraconservate del genoma umano.

Appendice C - Archivio dei Codici. Totalità del codice scritto nell'ambito del progetto di tesi, riportato in forma completa ed all'ultima versione utile per chiarimenti ed approfondimenti.

Indice

1 Background **11**
1.1 Tabelle xls e script in R 11
1.2 Problematiche dei dati grezzi 15
1.2.1 Ridondanza e rumore 15
1.2.2 Incoerenza 16
1.2.3 Analizzabilità dei dati 16
1.3 Nuove necessità 17

2 Sviluppo **19**
2.1 Progettazione e realizzazione del DB 20
2.1.1 diagramma E/R 20
2.1.2 Progetto logico 23
2.1.3 Sript SQL 24
2.1.4 Creazione DB in Mysql 26
2.2 Script Java per il recupero dati e Insert SQL 27
2.2.1 Perchè Java? 27
2.2.2 Il codice 27
2.2.3 Risultati 33
2.3 Inserimento e linking della HDO 34
2.3.1 Recupero e riduzione dell'ontologia 34
2.3.2 Adattamento della struttura del Database 40
2.3.3 Adattamento e creazione degli script Java 42
2.4 Progettazione e realizzazione dell'interfaccia Web 49
2.4.1 Struttura e modalità di accesso 49
2.4.2 Cenni sulla realizzazione 50

3 Risultati **57**

4 Conclusioni e sviluppi futuri **63**

A Glossario di base di biologia molecolare **65**

B Elementi ultraconservati nel genoma umano **68**

C Archivio dei Codici **70**
C.1 Uc_biomaRt_pathology2.java 70
C.2 Tabelle.sql 71
C.3 BiomartMain.java 73
C.4 GeneRecover.java 73
C.5 SplicingRecover.java 76
C.6 SnpRecover.java 79
C.7 PathologyRecover.java 81
C.8 TestCorrispondenze.java 85
C.9 OboEditAllFilter.java 88
C.10 PathologyObj.java 91
C.11 NTree.java 92
C.12 NTreeNode.java 95
C.13 TreeMaker.java 96
C.14 RiempiPatTable.java 98
C.15 Index.html 99
C.16 Uc_data_mining.php 102
C.17 Table.php 106
C.18 Sequence.php 109
C.19 Result_search_uc.php 110
C.20 Result_search_pat.php 110
C.21 Result_gene.php 111
C.22 Result.php 115
C.23 Related_works.html 120
C.24 Blastpat.php 123
C.25 Blastn.php 129
C.26 About_us.html 132
C.27 Style.css 134

Bibliografia **150**

Capitolo 1

Background

Al fine di rendere ben intellegibili le volontà che hanno spinto a profondere una tale quantità di energie, quali lo stesso studio di tesi o la redazione dell'elaborato in questione, si vuole illustrare con precisione lo stato tecnico operativo nel quale si versava inizialmente, prima dell'intervento di una vera collaborazione informatica nell'ambito biologico. Nei paragrafi successivi verrà dato ampio spazio, inoltre, alle problematiche ed alle incertezze introdotte da una scarsa accuratezza o formalità nell'archiviazione dei dati ed i danni che essa può produrre.

1.1 Tabelle xls e script in R

Nel seguente paragrafo si comincerà a fare riferimento oltre che a termini tecnici puramente informatici anche ad alcuni di matrice biologica. Si faccia riferimento, dunque, in caso di smarrimento, all'appendice **A - Glossario di base di biologia molecolare** per chiarimenti.
I dati grezzi, con i cui ci si poteva rapportare all'inizio del percorso, erano rappresentati da cinque tabelle nel formato *xls*[1].

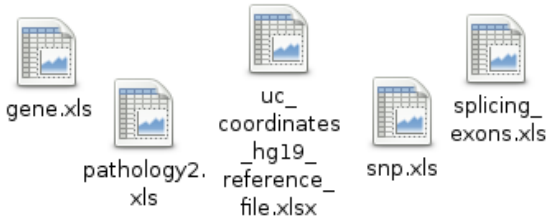


Fig. 1.1: Tabelle xls

[1]Estensione Microsoft Excel

Ognuno di questi file, seguendo per quanto possibile le intenzioni dell'ignoto progettista, avrebbe dovuto enucleare diversi concetti:

- **uc_coordinates_hg19_reference_file.xlsx**: File contenente le coordinate indispensabili ad ogni sequenza ultraconservata (identificatore della sequenza, numero cromosoma di appartenenza, coordinata d'inizio nel cromosoma, coordinata di fine, nome del filamento DNA).

- **gene.xls**: File contenente le informazioni circa i geni che contengono le sequenze ultraconservate (identificatore di sequenza coinvolta, nome del gene, identificatore univoco del gene, tipo di gene, nome del cromosoma di appartenenza, coordinata all'interno del cromosoma dell'inizio del gene, coordinata della fine del gene, banda cromosomica di appartenenza, nome del filamento di DNA in cui si trova il gene).

- **pathology2.xls**: File contenente le informazioni circa le patologie correlate ad un certo gene (identificatore ultraconservata coinvolta, identificatore gene correlato, descrizione *MIM*[2] circa la patologia genetica, nome della patologia, contenuto *GC*[3] del gene)

- **snp.xls**: File contenente informazioni circa le variazioni *SNP*[4] correlate ad una certa sequenza ultraconservata (identificatore di sequenza, id del polimorfismo, id dell'eventuale gene di appartenenza, *allele*[5] mutato, coordinata della variazione, nome del filamento di DNA sul quale è avvenuta la mutazione, Significatività clinica, descrizione fenotipica, eventuale validazione di un'ente).

- **plicing_exons.xls**: File contenente informazioni circa gli *Splicing*[6] correlate ad una certo gene (identificatore di sequenza, id del gene correlato, tipo di Splicing, codice di Splicing e nome esteso).

[2]Mendelian Inheritance in Man: è una banca dati che cataloga tutte le patologie aventi una componente genetica.
[3]Due delle cinque basi che compongono i nucleotidi del DNA e dell'RNA
[4]Single Nucleotide Polymorphism o SNP: polimorfismo a singolo nucleotide
[5]In genetica si definisce allele o fattore ogni variante di sequenza di un gene o di un locus genico
[6]In biologia molecolare e in genetica, splicing è una modificazione del nascente pre-mRNA

I files in questione sono stati ottenuti mediante interrogazione del web service offerto da *BioMart*[7] [10]. Come? attraverso degli obsoleti script in *R*[8]:

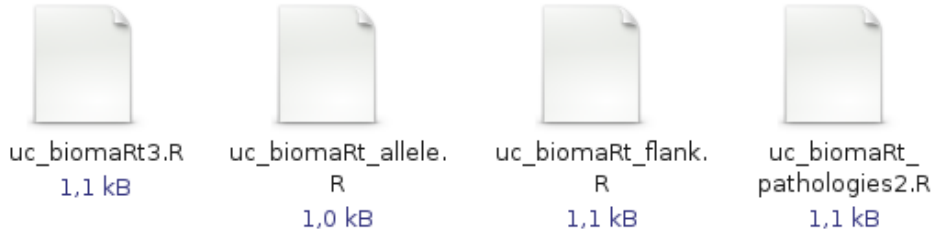


Fig. 1.2: Script in R

Tuttavia come facilmente notificabile dai titoli e dalla disparità numerica rispetto al conto dei fai xls, questi script non combaciano con la creazione degli stessi, ma offrono, tuttavia, un ottimo spunto per capire come effettivamente il web service lavora e come è possibile interrogarlo. Di seguito il codice, da considerare come un esempio, del file **uc_biomaRt_pathology2.R**:

```
1  library("biomaRt")
2  ensembl = useMart("ensembl",dataset="hsapiens_gene_ensembl")
3
4  options(width=120)
5  attributesR=c('ensembl_gene_id','mim_morbid_description',
6         'pathology','percentage_gc_content')
7  filtersR = c('chromosome_name','start','end')
```

Nella riga 1 importiamo il *package biomaRt*[9] [9]. Successivamente si costruiscono gli oggetti indispensabili all'interrogazione utilizzando le funzioni apposite che il pacchetto biomaRt offre. Si sceglie il centro di raccolta dei database (*ensembl*[10]) e il database da interrogare (in questo caso il database genetico dell'Homo Sapiens). Infine gli *attributi* ossia le informazioni che vogliamo recuperare ed i *filtri* secondo i quali effettuare la ricerca.

[7] BioMart è un sistema software per rendere le informazioni biologiche più accessibili ai biologi
[8] R è un linguaggio di programmazione libero per l'analisi statistica dei dati e molto di più
[9] Un intefaccia scritta in R ai database offerti da BioMart attraverso il suo web service.
[10] Ensembl è una banca dati bioinformatica allestita con lo scopo di fornire informazioni aggiornate sui principali genomi eucariotici

```
out.list<-list()
for (i in 1:length(uc2009[,1])){
      valuesR<-list(uc2009[i,2],uc2009[i,3],uc2009[i,4])
      out<-getBM(attributes=attributesR,filters=filtersR,
            values=valuesR, mart=ensembl,uniqueRows=F)
      if (nrow(out)!=0){
       out<-data.frame(cbind(uc2009[i,1],out))
         colnames(out)[1]<-"uc"
      }
      else{
        out.names<-colnames(out)
        nulli<-as.data.frame(matrix(rep(NA,ncol(out)),nrow=1))
        out<-data.frame(cbind(uc2009[i,1],nulli))
        colnames(out)<-c("uc",out.names)
      }

      if(i==1)
          write.table(out,"pathology.csv",append=FALSE,
           col.names=TRUE,quote=FALSE,row.names=FALSE,sep=";")
      else
          write.table(out,"pathology.csv",append=TRUE,
           col.names=FALSE,quote=FALSE,row.names=FALSE,sep=";")
      out.list[[i]]<-out
      print(uc2009[i,1])
      print(out)
}
```

La seconda parte di codice cicla su ogni elemento dell'array uc2009 (ossia le sequenze ultraconservate), creato in precedenza nel workspace, ed utilizzando specifiche funzioni offerte da R scrive, infine, un output tabellare in formato *cvs*[11]
Il risultato finale, se importiamo questi file csv in un software per l'elaborazione di fogli elettronici come ad esempio Open Office Calc, è modesto sia dal punto di vista estetico che da quello pramatico d'indagine, ma questa intuizione verrà argomentata nel paragrafo successivo.

[11] comma-separated values (abbreviato in CSV) è un formato di file basato su file di testo utilizzato per l'importazione ed esportazione di una tabella di dati.

1.2 Problematiche dei dati grezzi

Appreso il funzionamento del web service e la grezza archiviazione dei dati apportata allo stato attuale, sono evidenziabili fin da subito problemi di elevata caratura per quanto concerne l'affidabilità e l'analizzabilità dei dati, fine ultimo per un biologo.

1.2.1 Ridondanza e rumore

La più evidente delle mancanze è senza dubbio la ridondanza dei dati presente sui più e più file.

	A	B	C	D
1	**Id**	**wikigene_name**	**ensembl_gene_id**	**gene_biotype**
2	uc.1	PEX14	ENSG00000142655	protein_coding
3	uc.2	CASZ1	ENSG00000130940	protein_coding
4	uc.3	CASZ1	ENSG00000130940	protein_coding
5	uc.4	CASZ1	ENSG00000130940	protein_coding
6	uc.5	CASZ1	ENSG00000130940	protein_coding
7	uc.6	CASZ1	ENSG00000130940	protein_coding
8	uc.7	CASZ1	ENSG00000130940	protein_coding
9	uc.8	CASZ1	ENSG00000130940	protein_coding

	A	B	C	D	E
1	**Id**	**refsnp_id**	**ensembl_gene_stable_id**	**allele**	**chrom_start**
2	uc.1	rs190053770	ENSG00000142655	G/A	10597738
3	uc.1	rs190053770	ENSG00000142655	G/A	10597738
4	uc.1	rs190053770	ENSG00000142655	G/A	10597738
5	uc.1	rs190053770	ENSG00000142655	G/A	10597738
6	uc.1	rs190053770	ENSG00000142655	G/A	10597738
7	uc.1	rs181426894	ENSG00000142655	A/C	10597822
8	uc.1	rs181426894	ENSG00000142655	A/C	10597822
9	uc.1	rs181426894	ENSG00000142655	A/C	10597822
10	uc.1	rs181426894	ENSG00000142655	A/C	10597822
11	uc.1	rs181426894	ENSG00000142655	A/C	10597822

	A	B	
1	**Id**	**ensembl_gene_id**	**mim_morbid_description**
2	uc.1	ENSG00000142655	PEROXISOME BIOGENESIS FACTOR 14
3	uc.1	ENSG00000142655	PEROXISOME BIOGENESIS FACTOR 14
4	uc.1	ENSG00000142655	PEROXISOME BIOGENESIS FACTOR 14
5	uc.1	ENSG00000142655	PEROXISOME BIOGENESIS FACTOR 14
6	uc.1	ENSG00000142655	PEROXISOME BIOGENESIS FACTOR 14

Fig. 1.3: Ridondanza nel Formato tabellare

È subito ravvisabile nelle sue diverse forme: dalla ridondanza dovuta ai join tebellari effettuati lato server da BioMart a quelli introdotti dagli script lo-

cali in una mancata progettazione di archiviazione in forma normale dei dati stessi (si vedano le informazioni ripetute su più file).
Non meno evidenti le proporzioni di rumore (mancanza di dati informativi) presenti nei file.

310	uc.293	NA	NA	NA
311	uc.294	NA	NA	NA
312	uc.295	NA	NA	NA
313	uc.296	NA	NA	NA
314	uc.297	NA	NA	NA
315	uc.298	NA	NA	NA

Fig. 1.4: Rumore di fondo nel formato tabellare

Ridondanza e rumore accreccoscono le dimensioni dei file e diminuiscono la leggibilità dei dati ma introducono sopratutto i problemi elencati di seguito.

1.2.2 Incoerenza

Nel momento in cui si volesse aggiornare o modificare i dati sopracitati, ad esempio modificando l'identificatore di una sequenza ultraconservata, nulla potrebbe essere più drammatico: per ciascun file bisognerebbe manualmente trovare l'identificatore precedente e sostituirlo con il nuovo per ogni sua occorrenza. Non è molto difficile immaginare, come in un clima del genere, sia possibile incappare in grossolani errori di incoerenza a seguito di numerose modifiche o correzioni.

1.2.3 Analizzabilità dei dati

Quando ci si trova in situazioni analoghe la quantità di dati in presenza di rumore e ridondanze cresce vertiginosamente:

9947	uc.483	ENSG00000131374	NA
9948	uc.483	ENSG00000131374	NA
9949	uc.483	ENSG00000131374	NA
9950	uc.483	ENSG00000131374	NA
9951	uc.483	ENSG00000131374	NA
9952	uc.483	ENSG00000131374	NA

Fig. 1.5: Effetti ridondanza

Il danno è tutto a carico dell'analizzatore dei dati, il biologo in questo caso, che si ritrova costretto ad estrarre delle informazioni in condizioni sempre più difficoltose.

1.3 Nuove necessità

Realizzate le problematiche evidenziate durante il capitolo, risulta dunque comprensibile la nuova volontà di un'evoluzione in una direzione tecnico-informatica per far fronte alle nuove necessità della biologia molecolare moderna: analizzare e gestire agevolmente grandi quantità di dati ma sopratutto estrarre **informazioni** in modo agevole, funzionale e creativo. In questa direzione sono sicuramente gli obiettivi tracciati e sicuramente raggiunti della tesi di ricerca: Un database per garantire le proprietà *ACID*[12] e tutte le cure necessarie per salvaguardare l'integrità e l'affidabilità dei dati; Un Portale web per rendere tali dati accessibili e navigabili agevolmente, senza nient'altro che un browser; La possibilà di analizzarli in senso creativo e trasversale: attraverso l'inserimento nel database di un ontologia biomedica sulle patologie umane. Più avanti, nel capitolo successivo, sarà chiaro quanto a fronte dei dati grezzi tutte queste tecniche possano beneficiare ad un biologo, ricercatore o chiunque interessato a reperire informazioni sulle sequenze ultraconservate del genoma umano.

[12] Atomicità, Coerenza, Isolamento e Durabilità (ACID): Sono le proprietà logiche che devono avere le transazioni in un database moderno.

Capitolo 2

Sviluppo

Effettuata un'analisi accurata delle problematicità e necessità derivanti dall'uso di dati grezzi, analizzati nel capitolo precedente, è finalmente possibile passare alla fase creativa di sviluppo. Nel corso del capitolo verranno illustrate, seguendo un ordine temporale, le operazioni che porteranno dei dati sparpagliati, problematici e ridondanti a costituire una banca dati strutturata organicamente e funzionale all'interrogazione efficente.

2.1 Progettazione e realizzazione del DB

Il primo e forse più importante passo per la buona riuscita del progetto di migliorie d'apportare è la progettazione del Database: Esso costituisce il punto nevralgico di organizzazione logica dei dati e il pilastro fondamentale per la garanzia di efficacia che il progetto vorrebbe rilanciare.

2.1.1 diagramma E/R

Si agisce innanzitutto isolando le entità che comporranno il nostro modello logico, ossia i concetti fondamentali che vorremmo rappresentare, e successivamente arricchendole con gli attribuiti, ossia le informazioni relative che vorremmo memorizzare per ciascuna entità:

- **Sequenza ultraconservata**
  - **chr**: Identificativo del cromosoma di appartenenza
  - **Start**: Numero di *bp*[1] dall'inizio del cromosoma che identifica l'inizio della sequenza.
  - **End**: Numero di bp che identifica la fine della sequenza.
  - **Sequence**: Sequenza reale delle basi azotate ATGC .
  - **Strand**: Identificatore filamento di DNA dell'ultraconservata.
  - **Uc_name**: Nome mnemonico per identificare una sequenza.
- **Gene**
  - **Gene_start_pos**: Numero bp identificante l'inizio del gene.
  - **Gene_end_pos**: Numero bp identificante la fine del gene.
  - **Ensebl_gene_id**: Identificativo univoco assegnato da *Ensembl*.
  - **Band**: identificativo della banda cromosomica di appartenenza.
  - **Strand**: Identificatore filamento di DNA di registrazione del gene.
  - **Wikigene_name**: Nome mnemonico per identificare un gene.
  - **%G/C**: Percentuale delle basi azotate G e C all'interno del gene.
  - **Gene_biotype**: Tipo biologico di gene.

[1] Le coppie di basi o paia di basi (abbreviate come pb o, dall'inglese base pair, bp o bps) sono comunemente utilizzate come misura della lunghezza fisica di sequenze di acidi nucleici a doppio filamento

- **Patologia**
  - **Id**: Identificativo univoco della patologia.
  - **Nome**: Nome univoco esteso per definire la patologia.
  - **...**: Si vedrà nella sezione di linking dell'ontologia come migliorare gli attributi di questa entità.
- **Splicing**
  - **Event_Type_cod**: Codice univoco identificativo del tipo di splicing.
  - **Event_Type_name**: Nome esteso identificativo del tipo di splicing.
- **SNP**
  - **Refsnp_id**: Codice univoco identificativo dell'SNP.
  - **Start**: Identificativo numerico codificante la bp di inizio.
  - **Allele**: Allele mutato.
  - **Minor_allele_freq**: Frequenza di occorrenza dell'allele meno comune.
  - **Phenotipe_desc**: Descrizione fenotipica.
  - **Strand**: Identificativo del filamento di DNA su cui è avvenuta la mutazione.
  - **Validation**: Ente che ha effettuato validazione.
  - **Clinical_Significance**: Significatività clinica.

Lo step successivo è il delineamento delle relazioni tra le sopracitate entità. Di seguito si propone il diagramma E/R[2] del Progetto:

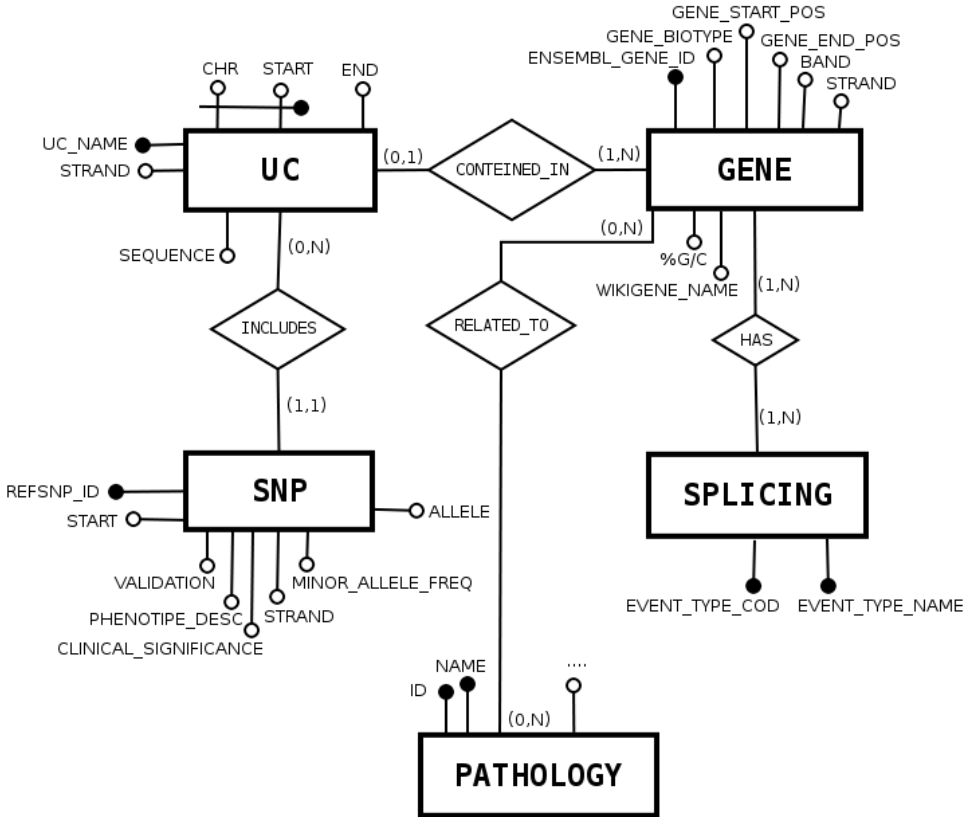


Fig. 2.1: Diagramma E/R

Si presti particolare attenzione alle cardinalità delle relazioni e le chiavi delle entità. Come facilmente ravvisabile, sono già scomparse le ridondanze rilevate nei deti grezzi in formato tabellare. Nel prossimo paragrafo verrà sviluppato il primo passaggio per la traduzione del modello logico in un vero e proprio database: il progetto logico.

[2]Il modello entità-relazione in informatica è un modello per la rappresentazione concettuale dei dati ad un alto livello di astrazione.

2.1.2 Progetto logico

Il progetto logico rappresenta la traduzione schematica di quanto proposto sottoforma di E/R. Esso costituisce, in sunto, un valido canovaccio per la stesura del codice *SQL*[3] per la creazione del database. Di seguito, più esplicativo di mille parole, il risultato:

```
UC (UC_NAME, CHR, START,END, STRAND, SEQUENCE, ENSAMBL_GENE_ID)
        AK: UC_NAME
        FK: ENSAMBL_GENE_ID REFERENCES GENE

SNP (REFSNP_ID, START, ALLELE, VALIDATION, MINOR_ALLELE_FREQ,
         PHENOTIPE_DESC, CLINICAL_SIGNIFIANCE, STRAND, CHR, START)
        FK: CHR, START REFERENCES UC NOT NULL

SPLICING (EVENT_TYPE_COD,EVENT_TYPE_NAME)
        AK: EVENT_TYPE_NAME

HAS (EVENT_TYPE_COD, ENSEMBL_GENE_ID)
        FK: EVENT_TYPE_COD REFERENCES SPLICING
        FK: ENSEMBL_GENE_ID REFERENCES GENE

PATHOLOGY (ID, NAME, ...)
        AK: NAME

RELATED_TO (ID, ENSAMBL_GENE_ID)
        FK: ID REFERENCES PATHOLOGY
        FK: ENSAMBL_GENE_ID REFERENCES GENE

GENE (ENSEMBL_GENE_ID, GENE_BYOTIPE, GENE_START_POS,
        GENE_END_POS, BAND, STRAND, WIKIGENE_NAME, G/C_PERC)
```

Effettuata la traduzione dal modello allo schema logico non resta nient'altro da fare che tradurre quanto proposto in codice SQL, che viene proposto nel paragrafo successivo.

[3] In informatica SQL (Structured Query Language) è un linguaggio standardizzato per database basati sul modello relazionale

2.1.3 Sript SQL

Di seguito viene riportato il codice SQL per la creazione delle tabelle. Si rammenta che si tratta di un codice portabile in un qualsiasi *DBMS*[4] relazionale moderno.

```sql
CREATE TABLE GENE
(
  ENSAMBL_GENE_ID VARCHAR(20),
  GENE_BYOTIPE VARCHAR(30),
  GENE_START_POS INTEGER,
  GENE_END_POS INTEGER,
  BAND VARCHAR(10),
  STRAND INTEGER,
  WIKIGENE_NAME VARCHAR(30),
  G_C_PERC VARCHAR(10),
  PRIMARY KEY(ENSAMBL_GENE_ID)
);

CREATE TABLE UC
(
  UC_NAME VARCHAR(10) UNIQUE NOT NULL,
  CHR VARCHAR(5),
  START INTEGER,
  ENDING INTEGER,
  STRAND INTEGER,
  SEQUENCE VARCHAR(1000) NOT NULL,
  ENSAMBL_GENE_ID VARCHAR(20),
  PRIMARY KEY (CHR,START),
  FOREIGN KEY (ENSAMBL_GENE_ID) REFERENCES GENE
    ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE SNP
(
  REFSNP_ID VARCHAR(30),
  START INTEGER,
  ALLELE VARCHAR(10),
  VALIDATION VARCHAR(30),
  MINOR_ALLELE_FREQ FLOAT(10),
  PHENOTIPE_DESC VARCHAR(30),
  CLINICAL_SIGNIFIANCE VARCHAR(30),
  STRAND INTEGER,
  CHR VARCHAR(5),
  START_UC INTEGER,
  PRIMARY KEY (REFSNP_ID),
```

[4]In informatica, un Database Management System (abbreviato in DBMS) è un sistema software progettato per consentire la creazione e la manipolazione (da parte di un amministratore) e l'interrogazione efficiente (da parte di uno o più utenti) di database.

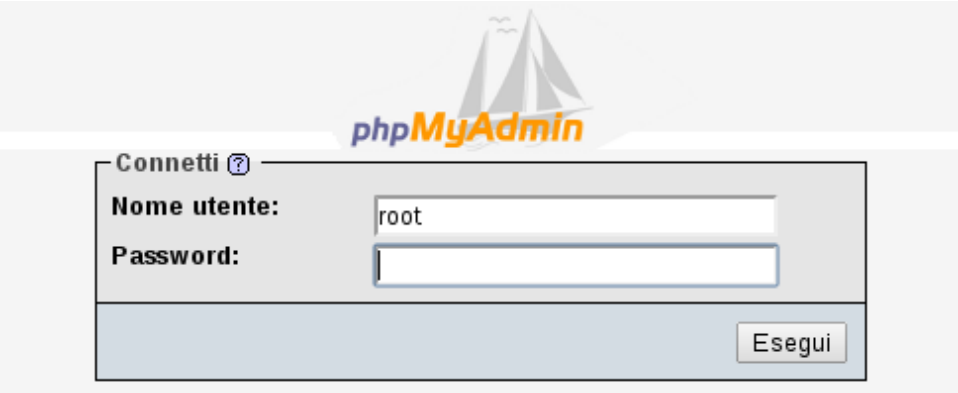
```
  FOREIGN KEY(CHR, START_UC) REFERENCES UC (CHR,START)
    ON DELETE CASCADE ON UPDATE CASCADE
);


CREATE TABLE SPLICING
(
  EVENT_TYPE_COD VARCHAR(8),
  EVENT_TYPE_NAME VARCHAR(30) UNIQUE NOT NULL,
  PRIMARY KEY(EVENT_TYPE_COD)
);

CREATE TABLE HAS
(
  EVENT_TYPE_COD VARCHAR(8),
  ENSAMBL_GENE_ID VARCHAR(20),
  PRIMARY KEY(EVENT_TYPE_COD, ENSAMBL_GENE_ID),
  FOREIGN KEY(EVENT_TYPE_COD) REFERENCES SPLICING
    ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY(ENSAMBL_GENE_ID) REFERENCES GENE
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE PATHOLOGY
(
  ID VARCHAR(15),
  NAME VARCHAR(100) NOT NULL,
  PRIMARY KEY(ID)
);

CREATE TABLE RELATED_TO
(
  ID VARCHAR(15),
  ENSAMBL_GENE_ID VARCHAR(20),
  PRIMARY KEY(ID, ENSAMBL_GENE_ID),
  FOREIGN KEY(ID) REFERENCES PATHOLOGY
    ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY(ENSAMBL_GENE_ID) REFERENCES GENE
    ON DELETE CASCADE ON UPDATE CASCADE

);
```

Si rammenta che per la tabella Pathology verranno apportate ulteriori migliorie nel paragrafo relativo all'inserimento e linking dell'ontologia biomedica.

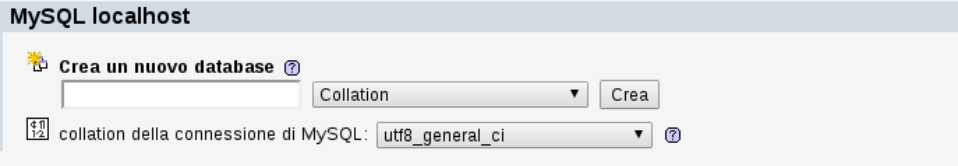
2.1.4 Creazione DB in Mysql

Ultimo step per la creazione effettiva del database è l'importazione nel DBMS dello script SQL visto nel paragrafo precedente: di seguito viene illustrata in massima sintesi la procedura adottata sul DBMS *Mysql* attraverso l'interfaccia scritta in *php*[5] *phpmyAdmin*.

Innanzitutto si accede tramite "localhost/phpmyadmin" inserendo opportunamente le credenziali di accesso.



Succesivamente si accede alla procedura guidata per creare un nuovo database.



Infine si utilizza la scheda *importa* per importare un file sql da eseguire.

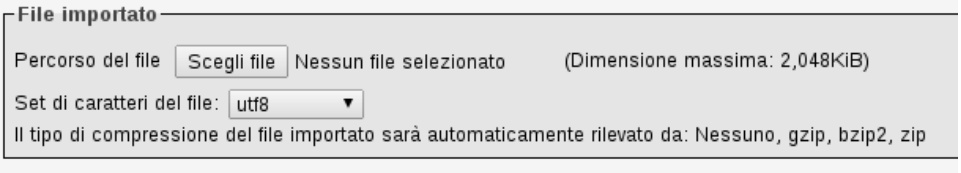


Fig. 2.2: Phpmyadmin

[5]PHP (acronimo ricorsivo di "PHP: Hypertext Preprocessor") è un linguaggio di programmazione interpretato, originariamente concepito per la programmazione Web

2.2 Script Java per il recupero dati e Insert SQL

La sezione seguente si sviluppa nella direzione di realizzare gli script SQL di inserimento per il riempimento delle tabelle create in precedenza. Ci si rifà, in senso procedurale, a quanto detto riguardo il reperimento dei dati attraverso gli script in R e l'interrogazione del web service offerto da BioMart. In questo caso, tuttavia, verrà utilizzato il linguaggio Java e gli script saranno impostati per ottenere i dati da biomart e per creare *direttamente* gli script SQL, in modo da enucleare al meglio, a livello concettuale, queste funzionalità.

2.2.1 Perchè Java?

La scelta di Java forse non risulta essere la più indicata per la prototipizzazione rapida non essendo un linguaggio prettamente dinamico. Tuttavia era stata presa in considerazione ed infine scelta perchè, stando alla documentazione biomart, erano offerti dei framework di lavoro per l'interrogazione del web service attraverso *SOAP API*[6] (ma che in realtà per scelte successive non verranno utilizzate). Inoltre risultava un linguaggio molto conosciuto e dalla indubbia portabilità, lasciando aperta ogni possibilità su future migliorie nel recupero *automatico* dei dati, ad esempio fornendo un'agevole *UI*[7].

2.2.2 Il codice

In ambiente di sviluppo *Eclipse*[8] sono stati scritti 5 script principali:

Fig. 2.3: Script java

[6]In informatica SOAP (inizialmente acronimo di Simple Object Access Protocol) è un protocollo leggero per lo scambio di messaggi tra componenti software.
[7]L'interfaccia utente, anche conosciuta come UI (dall'inglese User Interface), è ciò che si frappone tra la macchina e l'utente, consentendo l'interazione tra i due.
[8]Eclipse è un ambiente di sviluppo integrato multi-linguaggio e multipiattaforma.

Nello specifico essi rappresentano:

- **BiomartMain.java**: Il main che inizializza le classi e avvia i metodi
- **GeneRecover.java**: Lo Script che recupera le informazioni inerenti le sequenze ultraconservate da file e quelle sui geni eventualmente correlati dal web service biomart. Quindi crea i uno script SQL per il riempimento delle tabelle UC e GENE.
- **SplicingRecover.java**: Lo Script che recupera le informazioni inerenti lo splicing dei geni e riempie le tabelle sql SPLICING ed HAS.
- **SNPRecover.java**: Lo Script che recupera le informazioni inerenti le SNP e quindi riempie la tabelle SNP.
- **PathologyRecover.java**: Lo Script che recupera le informazioni inerenti le Pathology e quindi riempie la tabelle PATHOLOGY e RELATED_TO (lo script verrà rivisto poi alla luce delle migliorie introdotte a seguito del linking dell'ontologia biomedica).

A seguire si analizzerà solo uno di questi script, poichè essi agiscono tutti pressochè in modo analogo[9]. Si prende in analisi lo script **GeneRecover.java** analizzandone solo gli stralci principali, privi di controlli di robustezza per semplificarne la comprensione.
Nella primissima parte vengono importati gli oggetti di base per lo sviluppo. Successivamente si descrive la classe e gli attributi indispensabili al processo.

```
1  import java.io.*;
2  import java.net.URL;
3  import java.net.URLEncoder;
4  import java.util.StringTokenizer;
5
6  public class GeneRecover {
7    //dati per recupero info uc
8    private final int num_lines = 481;
9    private String[] chr = new String[num_lines];
10   private String[] bp_start = new String[num_lines];
11   private String[] bp_end = new String[num_lines];
12   private String[] strand_uc = new String[num_lines];
13   private String[] sequence = new String[num_lines];
14    strand_uc[lineNumber] = st.nextToken();
15   //dati per tabelle
16   private String ensambl_gene_id, gene_byotipe, gene_start_pos,
17           gene_end_pos, band, strand, wikigene_name,
18           g_c_perc, sql_gene, sql_uc;
```

[9]Si rimanda all'appendice **Archivio dei Codici** per approfondimenti

Tutto il lavoro viene svolto nel costruttore in modo da non dover chiamare nessun metodo e rendere il main il più semplice possbibile. Vengono creati ed indirizzati opportunamente i descrittori di input ed output. Il file **uc_coordinates_hg19_reference_file.csv** rappresenta i metadati relativi a ciascuna sequenza nel formato:

```
"Id","Chromosome","Start","End","Strand"
"uc.1","1",10597697,10597903,1
"uc.2","1",10732543,10732749,1
```

```
public GeneRecover()throws Exception{
  //leggo dal file le info
  try{
  //tsv file containing data
    String strFile_uc =
      "/home/uc_coordinates_hg19_reference_file.csv";
    String strFile_seq =
      "/home/ultraconservate/UC.txt";

    /* Si creano i descrittori per scrivere i file sql
       delle tabelle "UC" e "GENE". */
    BufferedReader br =
      new BufferedReader( new FileReader(strFile_uc));
    BufferedReader br2 =
      new BufferedReader( new FileReader(strFile_seq));
    FileOutputStream sql_file =
      new FileOutputStream("/home/uc_gene_insert.sql");
    PrintStream out =
      new PrintStream(sql_file);
```

Vengono rcuperate le sequenze reali dal file UC.txt formattato come quanto di seguito:

```
>uc.1
TCCACCGACAATGACCAGTTAGTCCTCATTTTCTCTCAA...
>uc.2
GCCCGCCCCCCTCCCCGGGCCCAATCTGTTTTCAAAGTG...
```

```
    //recupero sequenze uc
    int i = 1, j=0, k=0;
    while(i <= num_lines*2 ){
      strLine = br2.readLine();
      if(i%2 == 0) {sequence[j] = strLine; j++;}
      else {uc_name[k] = strLine.substring(1); k++;}
      i++;
    }
```

Si da avvio al ciclo principale che itera sul file delle uc recuperando i metadati per l'interrogazione del web service e inserendoli negli attributi della classe secondo l'indice *lineNumber* incrementato ad ogni ciclo.

```
1   StringTokenizer st = null;
2   int lineNumber = 0;
3   //la prima riga contenente meta-info viene scartata
4   br.readLine();
5   /* ciclo sulle sequenze
6   nel file uc\_coordinates\_hg19\_reference\_file.csv */
7   while( (strLine = br.readLine()) != null ){
8      //break comma separated line using ","
9      st = new StringTokenizer(strLine, ",");
10     String uc = st.nextToken();
11     chr[lineNumber] = st.nextToken();
12     //togliamo fastidiosi apici
13     chr[lineNumber] = chr[lineNumber].replace("\"", "");
14     bp_start[lineNumber] = st.nextToken();
15     bp_end[lineNumber] = st.nextToken();
16     strand_uc[lineNumber] = st.nextToken();
```

Si definisce, finalmente, il metodo di interrogazione: Se nel secondo capitolo, si era utilizzato il pacchetto *biomaRt* che metteva a disposizione i suoi metodi per l'interrogazione del web service, in questo caso si è scelto di effettuare la richiesta nel modo più semplice e naturale possibile, ossia tramite *REST*[10] API. Viene definito, dunque, un file xml per la strutturazione della query: Si scelgono in sequenza, il dataset su cui effettuare l'indagine, i filtri per la richiesta e gli attributi desiderati. Ad ogni ciclo, ovviamente, i filtri cambiano producendo il risultato atteso.

```
17  String myxml = "<Query virtualSchemaName = \"default\"
18        formatter = \"CSV\" header = \"1\" "
19  +"uniqueRows = \"0\" count = \"\"
20         datasetConfigVersion = \"0.6\" >"
21  +"<Dataset name = \"hsapiens_gene_ensembl\"
22           interface = \"default\" >"
23  +"<Filter name = \"chromosome_name\"
24           value = \""+chr[lineNumber]+"\"/>"
25  +"<Filter name = \"start\"
26           value = \""+bp_start[lineNumber]+"\"/>"
27  +"<Filter name = \"end\"
28           value = \""+bp_end[lineNumber]+"\"/>"
29  +"<Attribute name = \"wikigene_name\" />"
30  +"<Attribute name = \"ensembl_gene_id\" />"
```

[10]Representational state transfer (REST) è un tipo di architettura software per i sistemi di ipertesto distribuiti come il World Wide Web.

```
+"<Attribute name = \"gene_biotype\" />"
+"<Attribute name = \"start_position\" />"
+"<Attribute name = \"end_position\" />"
+"<Attribute name = \"band\" />"
+"<Attribute name = \"strand\" />"
+"<Attribute name = \"percentage_gc_content\" />"
+"</Dataset>"
+"</Query>";
```

Una volta definita la stringa xml, bastano poche righe di codice per inoltrare la richiesta al web service ed ottenere risposta nel BufferedReader.

```
String encoded = URLEncoder.encode(myxml, "utf-8");
URL url = new URL("http://www.biomart.org/biomart/martservice?
    query="+encoded);
InputStream response = url.openStream();
BufferedReader reader = new BufferedReader(new InputStreamReader
    (response));
```

All'interno del ciclo principale viene nidificato un'altro ciclo per l'analisi iterativa dei diversi record restituitici da biomart. Se non viene restituito nulla, si scrive su file solo il codice SQL per effettuare l'inserimento della sequenza ultraconservata, altrimenti, lo si accompagna al codice per l'inserimento del gene e le sue informazioni relative.

```
 String line;
 /*nel caso non ci sia nessuna risposta
  devo comunque inserire l'uc*/
 while (True){
    if((line = reader.readLine()) == null){
       sql_uc = "INSERT IGNORE INTO UC VALUES('"+
            uc_name[lineNumber]+"','"+ chr[lineNumber]+
            "',"+ bp_start[lineNumber]+
                 ","+ bp_end[lineNumber]+","
                 +strand_uc[lineNumber]+
                 ",'"+sequence[lineNumber]+"',null);";
       out.println(sql_uc);
       break;
    }
    else{
      //estrapoliamo le info dal record restituitoci
      st = new StringTokenizer(line, ",");
      wikigene_name = st.nextToken();
      if(!wikigene_name.equals("null"))
        wikigene_name = "'"+wikigene_name+"'";
      ensambl_gene_id = st.nextToken();
```

```
22          if(!ensambl_gene_id.equals("null"))
23            ensambl_gene_id = "'"+ensambl_gene_id+"'";
24          gene_byotipe = st.nextToken();
25          if(!gene_byotipe.equals("null"))
26            gene_byotipe = "'"+gene_byotipe+"'";
27          gene_start_pos = st.nextToken();
28          gene_end_pos = st.nextToken();
29          band = st.nextToken();
30          if(!band.equals("null"))
31            band = "'"+band+"'";
32          strand = st.nextToken();
33          if(!strand.equals("null"))
34            strand = "'"+strand+"'";
35          g_c_perc = st.nextToken();
36
37          sql_gene = "INSERT IGNORE INTO GENE VALUES("+
38                ensambl_gene_id+","+gene_byotipe+","+
39                gene_start_pos+","+gene_end_pos+","+
40                band+","+strand+","+wikigene_name+","
41                +g_c_perc+");";
42          //scriviamo su file
43          out.println(sql_gene);
44
45          sql_uc = "INSERT IGNORE INTO UC VALUES('"+
46              uc_name[lineNumber]+"','"+ chr[lineNumber]+"',"+
47              bp_start[lineNumber]+
48                  ","+ bp_end[lineNumber]+","+strand_uc[lineNumber]+
49                  ",'" +sequence[lineNumber]+"',"
50                  + ensambl_gene_id +");";
51                  //scriviamo su file
52          out.println(sql_uc);
53        }
54    }
```

Nel main bisognerà semplicemente stanziare un oggetto della classe descritta, così come per le altre costruite seguendo lo stesso scheletro logico, per ottenere tutti gli script SQL per il riempimento della banca dati.

```
1  public class BiomartMain {
2    public static void main(String[] args) throws Exception {
3
4      GeneRecover gene = new GeneRecover();
5      SnpRecover snp = new SnpRecover();
6      SplicingRecover spli = new SplicingRecover();
7      PathologyRecover path = new PathologyRecover();
8    }
```

2.2.3 Risultati

Eseguendo il main, otteniamo 4 script SQL per il riempimento automatico delle tabelle:

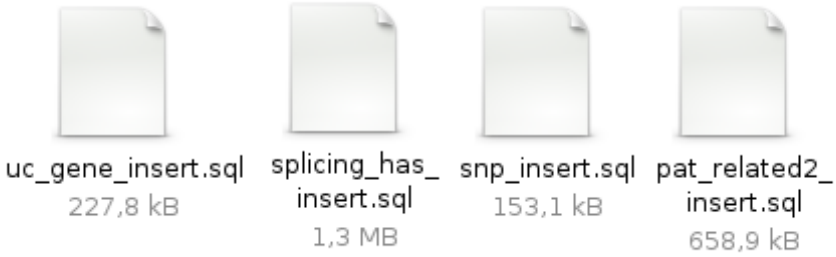


Fig. 2.4: Script SQL

Ciascuno dei quali avrà, all'incirca, lo stesso formato:

```
INSERT IGNORE INTO GENE VALUES('ENSG00000142655','
    protein_coding',10532345,10690815,'p36.22','1','PEX14',45.03);
INSERT IGNORE INTO UC VALUES('uc.1','1',10597697,10597903,1,
    'TCCACCGA...','ENSG00000142655');
INSERT IGNORE INTO GENE VALUES('ENSG00000130940','
    protein_coding',10696661,10856707,'p36.22','-1','CASZ1',56.73);
INSERT IGNORE INTO UC VALUES('uc.2','1',10732543,10732749,1,
    'GCCCGCCCCCCTC...','ENSG00000130940');
INSERT IGNORE INTO GENE VALUES('ENSG00000130940','
    protein_coding',10696661,10856707,'p36.22','-1','CASZ1',56.73);
........
```

Notare che sarà lo stesso DB (oltre che lo script nella sua interezza), grazie alla sua struttura, a filtrare le ridondanze ed il rumore nei dati. La clausola IGNORE viene qui utilizzata per trasformare gli *errors* dovuti alla presenza di record duplicati in *warnings* offrendoci un rendiconto generale sui problemi riscontrati ma senza interrompere il processo di inserimento.
Successivamente per ottenere un Database funzionante ed interrogabile basterà caricare gli script sopracitati allo stesso modo di quanto visto per l'importazione degli script per la creazione delle tabelle.

2.3 Inserimento e linking della HDO

Allo stato delle cose, il database risulta completo e funzionante. Tuttavia, si vorrebbe ottenere informazioni più dettagliate circa le patologie e sopratutto poter interrogare la basi di dati effettuando query strategiche che sappiano recuperare anche contenuti meno apparenti. L'idea è quella di introdurre nel Database una vera e propria otologia medica sulle patologie umane, o quantomeno parte di essa, completa nelle informazioni gerarchiche tra le patologie. In questo modo si renderanno possibili interrogativi del tipo: *ottenere tutte le sequenze ultraconservate che hanno una qualche relazione con il cancro o sue sottopatologie.*

2.3.1 Recupero e riduzione dell'ontologia

Tra le tante ontologie accessibili, come la celeberrima *SNOMED* [14], la scelta è ricaduta sulla *Human disease ontology* consultabile e scaricabile gratuitamente grazie all'esperimento collaborativo dell'*OBO Foundry* [15]. Il file in formato *obo*[11] segue la formattazione seguente:

```
[Term]
id: DOID:7479
name: duodenal somatostatinoma
synonym: "duodenal delta cell somatostatin producing tumor" EXACT []
xref: NCI:C27407
xref: UMLS_CUI:C1333320
is_a: DOID:10021 ! duodenum cancer

[Term]
id: DOID:7480
name: large cell carcinoma with rhabdoid phenotype
synonym: "large cell carcinoma with rhabdoid phenotype
         (morphologic abnormality)"
         EXACT [SNOMEDCT_2005_07_31:128629005]
synonym: "large cell lung carcinoma with Rhabdoid Phenotype"
         EXACT [NCI2004_11_17:C6876]
xref: NCI:C6876
xref: SNOMEDCT_2010_1_31:128629005
xref: UMLS_CUI:C1265997
is_a: DOID:4556 ! lung large cell carcinoma
```

Le enormi dimesioni dell'ontologia rappresentano, tuttavia, più un'ostacolo al progetto che un valore aggiunto, poichè solo una piccola parte delle

[11] Open Biomedical Ontology: formato adoperato per rappresentare esplicitamente significato e semantica di termini con vocabolari e relazioni tra gli stessi.

patologie presenti risulta essere effettivamente interessante allo scopo. È opportuno, dunque, estrarre solamente le patologie suggeriteci da Biomart in relazione a ciascun gene e **tutte quelle antenate sino alla radice**. Quello che si estrae, quindi, è un sotto-grafo minimale di **201** patologie a fronte **6393** elementi validi ed una centinaia di elementi considerati obsoleti del grafo originale.

Come è stato possibile compiere questa meticolosa estrazione? Innanzitutto tramite un nuovo script java, che per brevità e per la sua funzione strumentale non verrà mostrato nel dettaglio, si esaminano tutte le patologie recuperate da biomart e le si confronta con tutti i termini dell'ontologia nel formato precedentemente illustato. Se riusulta esserci un match tra in nome della patologia ed il nome del termine od un suo sinomimo allora quel termine viene rilevato dall'ontologia. Il secondo passo è l'indispensabile linking manuale delle patologie che biomart correla a determinati geni ma che non sono stati rilevati nell'ontologia o perchè troppo generici o perchè troppo particolarizzanti. Quello che dal punto informatico è possibile implementare è una serie di *suggerimenti* per il biologo tenuto ad effettuare il linking[12]. I file dei suggerimenti è ottenuto mediante l'algoritmo per calcolare la *Levenshtein distance*[13] di cui si è recuperata ed utilizzata la versione in java:

```
1  public static int computeLevenshteinDistance(CharSequence str1,
2                 CharSequence str2) {
3    int[][] distance =
4        new int[str1.length() + 1][str2.length() + 1];
5
6    for (int i = 0; i <= str1.length(); i++)
7       distance[i][0] = i;
8    for (int j = 1; j <= str2.length(); j++)
9       distance[0][j] = j;
10
11   for (int i = 1; i <= str1.length(); i++)
12      for (int j = 1; j <= str2.length(); j++)
13         distance[i][j] = minimum(
14            distance[i - 1][j] + 1,
15            distance[i][j - 1] + 1,
16            distance[i - 1][j - 1]
17            + ((str1.charAt(i - 1) == str2.charAt(j - 1)) ? 0 : 1));
18
19    return distance[str1.length()][str2.length()];
20 }
```

[12]Si veda *TestCorrispondenze.java* nell'archivio dei codici per approfondimenti

[13]la distanza di Levenshtein, o distanza di edit, è una misura per la differenza fra due stringhe.

Per ottenere l'elenco dei suggerimenti basterà, in maniera piuttosto semplice, ciclare su tutte le patologie di cui ancora non è stato trovato il corrispettivo nell'ontologia effettuando un controllo del tipo:

```
ris= computeLevenshteinDistance(do_name.get(k), patmim_name.get(i));
//se le differenze non sono piu' della meta' dei caratteri
if(ris < patmim_name.get(i).length()/2)
  out_file.println
  (
    patmim_name.get(i) +
    " somiglia a: " + do_name.get(k)
  );
```

Come risultato si otterrà un un file di testo con il quale un biologo esperto potrà agevolmente completare il lavoro di linking:

```
leukaemia somiglia a: leukemia
leukaemia somiglia a: leukopenia
lymphocytic somiglia a: lymphocele
lymphocytic somiglia a: lymphopenia
goitre somiglia a: goiter
........
```

Ma il lavoro per l'estrapolazione del sub-graph non è ancora concluso: bisognerà trovare un algoritmo o delle tecniche efficenti per venire a conoscenza di tutte quelle patologie coinvolte nei cammini (anche più di uno) che colleghino quelle di cui già si è a conoscenza alla radice dell'ontologia stessa. Solo in questo modo si potrà mantenere inalterata l'informazione gerarchica dell'ontologia. Si consideri l'esempio sottostante:

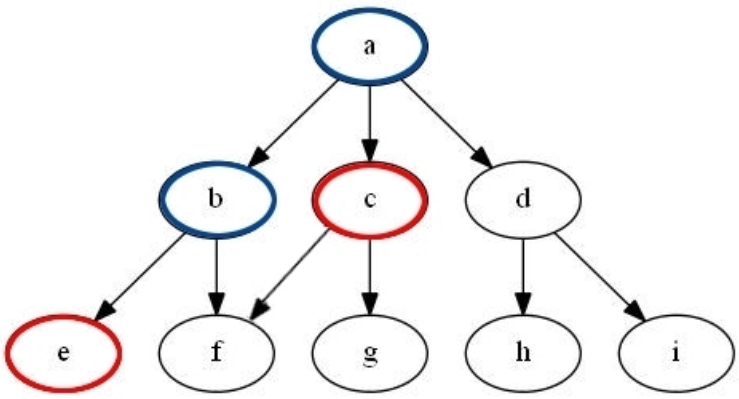


Fig. 2.5: Grafo d'esempio

Nell'immagine in figura si consideri in rosso *(c, e)* le patologie suggerite da biomart di cui si è ricondotto il termine corrispondente nell'ontologia. Ovviamente non si vuole prendere in cosiderazione il ramo del nodo *d* poichè non costituisce informazione rilevante ed in una futura interrogazione restituirebbe solo risultati vuoti circa le possibili sequenze o geni corelati. Saranno solo nodi cerchiati in blu *(a, b)* ad essere inseriti addizionalmente nel sotto-grafo, invece, perchè utili per generalizzare interrogazioni ottenendo informazioni sempre maggiori. Nell'esplorazione delle tecniche migliori per ottenere con il minor sforzo possibile gli effetti desiderati, risulta essere opportuno l'utilizzo del software open source *OBO-Edit* [19], un editor e visualizzatore di ontologie scritto in Java. Nello specifico sarà utilizzato il suo motore di ricerca ed il suo navigatore visuale per navigare il sottografo finale ed assicurasene la correttezza. Nella figura sottostante viene riportata la modalità semi-strutturata di interrogazione che verrà utilizzata: si specifica, ad esempio, che si vuole ottenere tutti i nodi dell'ontologia che hanno come antenato il nodo root *disease*.

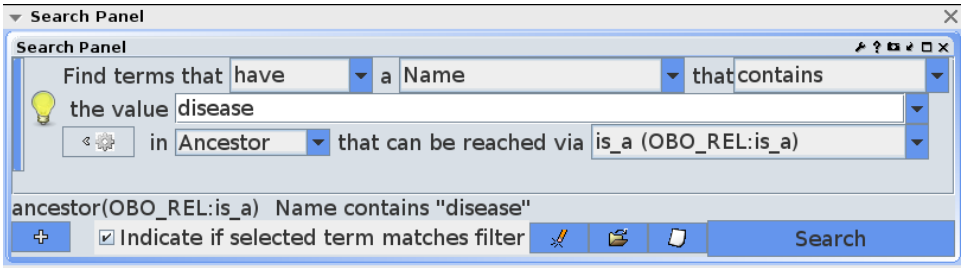


Fig. 2.6: Obo-Edit: Search panel

Tuttavia, compiere a mano questo lavoro, risulterebbe lungo e tedioso, ma fortunatamente, anche se non contemplata da OBO-edit nella sua documentazione è possibile un'altra soluzione: Il software sopracitato mette a disposizione un metodo per il salvataggio ed il caricamento delle query sottomesse al suo motore di ricerca; fortunatamente esse sono salvate in un intelleggibile xml che segue pressocchè la seguente struttura:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <java version="1.6.0_18" class="java.beans.XMLDecoder">
3   <object class="org.obo.filters.CompoundFilterImpl">
4    <void property="booleanOperation">
5     <int>1</int>
6    </void>
7    <void property="filters">
8     <void method="add">
9      <object class="org.obo.filters.ObjectFilterImpl">
10      <void property="aspect">
```

```
11        <object class="org.obo.filters.DescendantSearchAspect"/>
12       </void>
13       <void property="comparison">
14        <object id="EqualsComparison0" class="org.obo.filters.
             EqualsComparison"/>
15       </void>
16       <void property="criterion">
17        <object id="NameSynonymSearchCriterion0" class="org.obo.
             filters.NameSynonymSearchCriterion"/>
18       </void>
19       <void property="reasoner">
20        <object id="RuleBasedReasoner0" class="org.obo.reasoner.rbr.
             RuleBasedReasoner">
21         <void property="properties">
22          <object class="java.util.HashSet"/>
23         </void>
24        </object>
25       </void>
26       <void property="traversalFilter">
27        <object class="org.obo.filters.LinkFilterImpl">
28         <void property="filter">
29          <void property="comparison">
30           <object class="org.obo.filters.EqualsComparison"/>
31          </void>
32          <void property="criterion">
33           <object class="org.obo.filters.IDSearchCriterion"/>
34          </void>
35          <void property="value">
36           <string>OBO_REL:is_a</string>
37          </void>
38         </void>
39        </object>
40       </void>
41       <void property="value">
42        <string>aniridia</string>
43       </void>
44      </object>
45     </void>
46     <void method="add">
47      <object class="org.obo.filters.ObjectFilterImpl">
48       <void property="comparison">
49        <object idref="EqualsComparison0"/>
50       </void>
51       <void property="criterion">
52        <object idref="NameSynonymSearchCriterion0"/>
53       </void>
54       <void property="reasoner">
55        <object idref="RuleBasedReasoner0"/>
56       </void>
```

```
57        <void property="value">
58         <string>aniridia</string>
59        </void>
60       </object>
61      </void>
62     </void>
63    </object>
64   </java>
```

Il file xml come intuibile rappresenta la seguente query:

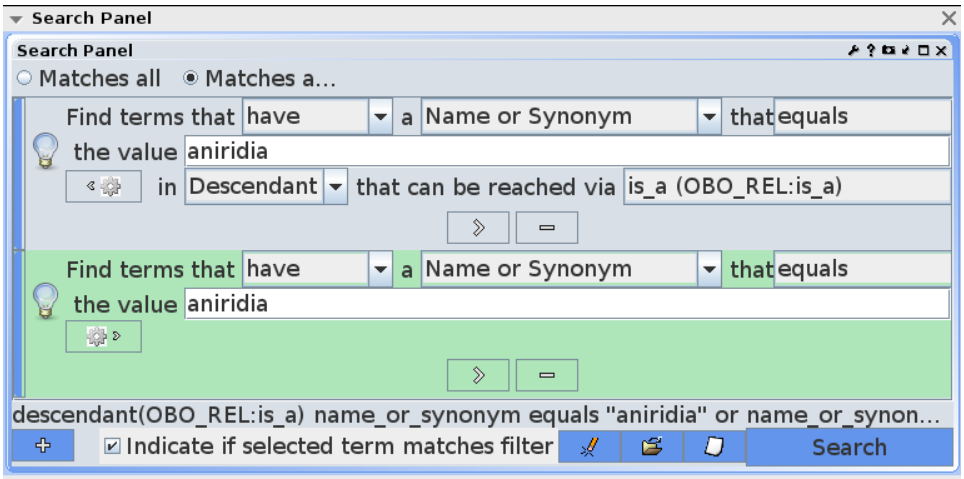


Fig. 2.7: Obo-Edit: Search panel

Ossia chiede che vengano restituiti tutti quei nodi che hanno tra i discendenti il nodo con nome e sinonimo "aniridia" o siano il nodo stesso. Anche se non propriamente strutturata per l'intellegibilità e modificabilità risulta chiaro l'utilizzo dei filtri e come sarebbe possibile aggiungere al file nuovi criteri di ricerca. Basterà dunque, attraverso un semplice script Java[14], modificare il file in modo che, per ogni patologia tra quelle recuperate e collegate all'ontologia, venga introdotta una struttura analoga.
Il risultato costituirà il sottografo che si vorrebbe introdurre nel database, di cui si propone la rappresentazione grafica per coglierne l'ingenza: Costruito il sottografo di interesse viene esportata da OBO-Edit la lista completa dei nodi che lo costituiscono (nome e ID).

[14]Si veda *OboEditAllFilter.java* nell'alrchivio codici per approfondimenti

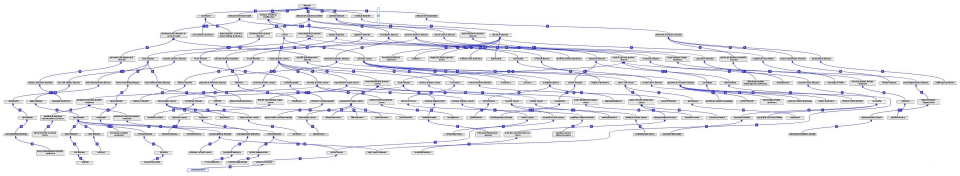


Fig. 2.8: Grafo della patologie estratte

2.3.2 Adattamento della struttura del Database

A seguito della scelta dell'ontologia è finalmente definibile con chiarezza la struttura di quanto lasciato precedentemente incompleto riguardo il modello logico e quindi l'implementazione del database, in particolare la definizione dell'entità patgology e dei suoi attributi. Di seguito la modifica del diagramma E/R ed del relativo progetto logico:

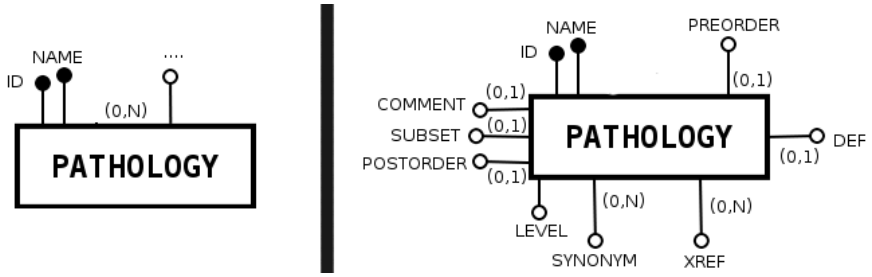


Fig. 2.9: Confronto migliorie

Oltre alle informazioni reperite dall'ontologia tradotte in attributi nel modello, si noti la modellazione della informazione gerachica[15] espressa negli attributi: level, preorder e postorder.
Perchè la scelta di questi attributi aggiuntivi?

"Given a node v [..] with pre(v) and post(v) ranks, the following properties are important towards our objectives: - all nodes x with pre(x) < pre(v) are the ancestors or preceding nodes of v;
- all nodes x with pre(x) > pre(v) are the descendants or following nodes of v;
- all nodes x with post(x) < post(v) are the descendants or preceding nodes of v;
- all nodes x with post(x) > post(v) are the ancestors or following nodes of v;

[15] Si rammenta che, per brevità, nel diagramma E/R l'entità pathology è direttamente riportata a seguito del collasso verso l'alto delle entità figlie.

- for any v [..], we have pre(v) - post(v) + size(v) = level(v);
- if pre(v) = 1, v is the root, if pre(v) = n, v is a leaf. [..]" [20]

Ossia, per riuscire a sapere se una patologia y è una sottopatologia di una patologia x basterà verificare che l'attributo preorder di y sia maggiore di quello di x e che **contemporaneamente** l'attributo di postorder di y sia minore di quello di x. Tuttavia questa logica è applicabile solo sui nodi di un *albero*; Si spiegherá più in avanti in che modo nel nostro caso. Momentaneamente ci si cocentri sulla struttura del database la cui modifica del progetto logico, dunque, risulta:

```
......

PATHOLOGY (ID, NAME, DEF, SUBSET, LEVEL, PREORDER,POSTORDER, COMMENT )
        AK: NAME

SYNONYM (ID, NAME)
        FK: ID REFERENCES PATHOLOGY

XREF (ID, REF)
        FK: ID REFERENCES PATHOLOGY
......
```

Infine tradotto in SQL:

```sql
.......

CREATE TABLE PATHOLOGY
(
  ID VARCHAR(15),
  NAME VARCHAR(100) NOT NULL,
  COMMENT VARCHAR(200),
  DEFINITION VARCHAR(500),
  SUBSET VARCHAR(100),
  PREORDER INTEGER,
  POSTORDER INTEGER,
  LEVEL INTEGER,
  PRIMARY KEY(ID)
);

CREATE TABLE SYNONYM
(
  ID VARCHAR(15),
  NAME VARCHAR(100),
  PRIMARY KEY(ID, NAME),
  FOREIGN KEY(ID) REFERENCES PATHOLOGY
```

```
136     ON DELETE CASCADE ON UPDATE CASCADE
137 );
138
139 CREATE TABLE XREF
140 (
141   ID VARCHAR(15),
142   REF VARCHAR(100),
143   PRIMARY KEY(ID, REF),
144   FOREIGN KEY(ID) REFERENCES PATHOLOGY
145     ON DELETE CASCADE ON UPDATE CASCADE
146 );
147 .......
```

2.3.3 Adattamento e creazione degli script Java

L'obiettivo successivo è quello di modificare gli script Java per la creazione delle INSERT SQL relativi alla nuova struttura dell'entità pathology pocanzi illustrata e crearne di nuovi. Nello specifico si dovrà:

1. Recuperare tutte le informazioni circa ogni singolo nodo del sub-graph, di cui attualmente si conosce solo nome ed id.

2. Strutturare le patologie a formare un *albero* per recuperare gli attributi di preorder, postorder e level, e successivamente inserire le stesse nel database.

3. Collegare ed inserire nel database le pathology indicate da biomart con particolare attenzione alla creazione, nel database, delle patologie di cui non si è trovato il corrispettivo nell'ontologia.

Effettuati questi punti si potrebbe dire concluso il procedimento di inserimento dell'ontologia. Si riportano i tratti salienti di ciascun punto:

1. Per il recupero delle informazioni circa una patologia viene creata una classe che prende in ingresso un id e cercando oppurtunamente nel file testuale costituente l'ontologia (la cui struttura è mostrata nel paragrafo sulla riduzione dell'ontologia) restituisce un oggetto patologia completo degli attributi di interesse. Si estrae di seguito il core principale[16].

```
1 public class PathologyObj {
2
3   public String id="";
4   public String name="";
5   public String def="";
6   public String subset="";
```

[16]Si rimanda al codice completo nell'appendice *PathologyObj.java*

```
7    public String comment="";
8    public ArrayList<String> synonym = new ArrayList<String>();
9    public ArrayList<String> xref = new ArrayList<String>();
10   public String is_a="";
11   public int pre = 0;
12   public int post = 0;
13   public int level = 0;
14
15   public PathologyObj(String id) throws IOException{
16       BufferedReader br =
17          new BufferedReader( new FileReader("/home/HumanDO.txt"));
18       String strLine;
19       while( (strLine = br.readLine()) != null ){
20        if(strLine.startsWith("id: "+id)){
21          this.id = strLine;
22          while( !(strLine = br.readLine()).startsWith("[Term]")){
23           if(strLine.startsWith("name:"))
24            this.name = strLine;
25           else if(strLine.startsWith("def:"))
26            this.def = strLine;
27           else if(strLine.startsWith("subset:"))
28            this.subset = strLine;
29           else if(strLine.startsWith("synonym:"))
30            this.synonym.add(strLine);
31           else if(strLine.startsWith("xref:"))
32            this.xref.add(strLine);
33           else if(strLine.startsWith("is_a:")) {
34             StringTokenizer st;
35             st = new StringTokenizer(strLine, "!");
36             String tmp = st.nextToken();
37             is_a = tmp.substring(0,tmp.length()-1);
38           }
39           }
40           break;
41         }
42       }
43   }
```

2. Questo punto rappresenta la parte più laboriosa e delicata, anzitutto perchè non si sta trattando con un albero bensì con un grafo. Poiché, però, solamente 2 dei 201 nodi d'interesse detiene più di una relazione di discendenza, si è deciso di poter introdurre una lieve ridondanza all'interno del database a fronte di un'indubbia semplificazione nella gestione delle informazioni gerarchiche considerando il grafo come un'albero. In questo caso, quindi, come è possibile verificare nello scorcio di grafo riportato in figura, la ridondanza è circoscritta ad i due nodi con due padri ed ad i relativi sotto-grafi: in tutto solo 4 nodi vengono ripetuti.

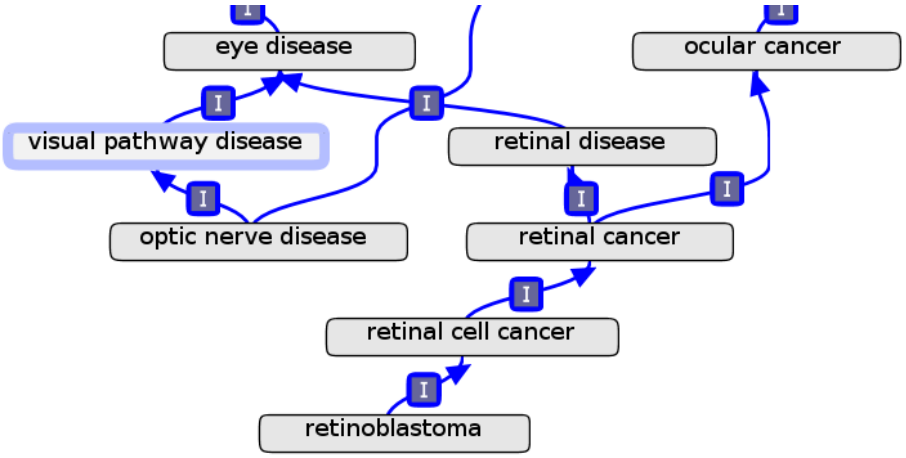


Fig. 2.10: Nodi con due padri

Per recuperare gli attributi, in ultima analisi, sarà sufficiente creare un programa che utilizzi la classe PathologyObj per il recupero delle informazioni circa ogni patologia di nostro interesse e che costruisca l'albero per poi effettuare le *visite in preordine e postordine*[17]. Nella pratica sono stati creati i seguenti file:

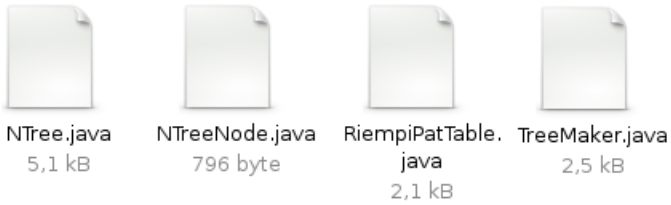


Fig. 2.11: Script Java per la gestione dell'albero

- **NTree.java**: Costituisce la classe dell'albero di patologie con le relative funzioni di visita.
- **NTreeNode.java**: Costituisce la classe per la definizione degli oggetti che costituiranno i nodi dell'albero.
- **TreeMaker.java**: Costituisce il Main per l'utilizzo dell'albero definito nelle classi precedenti (Si rammenta anche l'utilizzo della classe PathologyObj definita in precedenza).

[17] Preorder e Postorder sono due diversi algoritmi di attraversamento e visita dei nodi di un'albero.

- **RiempiPatTable.java** : Costituisce lo script per il riempimento, nel database, delle tabelle Pathology, Xref e Synonim.

Brevemente, si propongono alcuni scorci di codice di ciascuna classe per capirne meglio la funzione[18]:
Il file **NTreeNode.java** costituisce solo la struttura di ogni si nodo dell'albero memorizzandone l'elemento associato ma anche il riferimento al primo figlio ed al fratello.

```
1  public class NTreeNode
2  {
3      protected NTreeNode firstSon; //primo figlio
4      protected NTreeNode brother; //fratello
5      protected PathologyObj element; //elemento del nodo
6      public NTreeNode() //costruttore
7      {
8              this(null,null, null);
9      }
10     public NTreeNode(PathologyObj element, NTreeNode firstSon)
11     {
12         this(element, firstSon,null);
13     }
14
15     .......
```

Mentre, la classe descritta in **NTree.java** realizza l'albero con tutti i metodi indispensabili ad inserimenti, cancellazioni e modifiche oltre che le visite in preordine e postordine. Si evita di riproporre l'interezza del codice costituisce la normale gestione di un'albero e si ripropone solo l'implementazione dei metodi *set_preorder* e *set_postorder* che rappresentano l'unico elemento di novità assegnando a ciascun oggetto PathologyObj il relativo attributi di preordine e postordine ossia il numero progressivo di ordine di visita.

```
1  public class NTreeNode
2  {
3  public class NTree //classe pubblica albero
4  {
5          protected NTreeNode root; //radice dell'albero
6          public ArrayList<PathologyObj> array = new ArrayList<
               PathologyObj>(); //per dopo.. lista
7          public NTree() //costruttore
8          {
9                  root=null;
10         }
```

[18]Si rimanda all'**appendice C** per approfondimenti sul codice

```java

        public void setRoot(NTreeNode root){
          this.root = root;
        }

        ............

        public void set_preorder(){
            set_preorder(root,1);
        }
        public int set_preorder(NTreeNode p, int num){
            if(p!=null){
                p.element.pre = num;
                    NTreeNode t=p.firstSon;
                    while(t!=null){
                            num = set_preorder(t, ++num);
                            t=t.brother;
                    }
            }
            return num;
        }

        public void set_postorder(){
            set_postorder(root,1);
        }
        public int set_postorder(NTreeNode p, int num){
            if(p!=null){
                    NTreeNode t=p.firstSon;
                    while(t!=null){
                            num = set_postorder(t, num);
                            num++;
                            t=t.brother;
                    }
            p.element.post = num;
            }
      return num;
      }
```

Nel file **TreeMaker.java**, invece, non si fa altro che utilizzare le classi messe a disposizione in precedenza creando l'albero ed inserendo ciclicamente tutti le patologie (ogni ciclo aggiunge al minimo un livello) recuperandole dal file *allnodesfromobo.txt* che rappresenta l'insieme estratto precedentemente da Obo-Edit (Si rammenta che si è evitato di mostrare, ma é presente nello stesso file l'inserimento "manuale" delle patologie "duplicate" per la resa della gerarchia in forma d'albero):

```java
public class TreeMaker {
  NTree tree;

  public TreeMaker() throws IOException {
    BufferedReader br = new BufferedReader(
      new FileReader("/home/allnodesfromobo.txt"));
    ArrayList<PathologyObj> list =
      new ArrayList<PathologyObj>();
    StringTokenizer st = null;
    String strLine,id;

    //while per la creazione della lista di oggetti
    while((strLine = br.readLine()) != null ){
      st = new StringTokenizer(strLine, " ");
      id = st.nextToken();
      //System.out.println(id);
      list.add(new PathologyObj(id));
    }

    //creiamo l'albero e colleghiamo i nodi
    tree = new NTree();
    for(int i=0; i<list.size(); i++)
      if(list.get(i).id.equals("id: DOID:4")) {
        tree.setRoot(
          new NTreeNode(list.get(i),null,null));
        list.remove(i);
      }

    while(!list.isEmpty()){
      for(int i=0; i<list.size(); i++){
        //attenzione agli obsoleti! sono 3!
        if(list.get(i).is_a.equals(""))
           list.remove(i);
        String parent_id = "id: "+ list.get(i).is_a.substring(6);
        if(tree.insert(parent_id, new NTreeNode(list.get(i),null,
            null)))
          list.remove(i);


      }
    }
```

Infine, in **RiempiPatTable.java**, di cui si riporta solo la parte fontamentale, dopo l'aggiunta "manuale" delle patologie obsolete (estranee alla gerarchia), si cicla sulla lista delle patologie recuperata tramite uno specifico metodo della classe NTree, ed dopo avere effettuato specifici controlli di tipo e formato circa i sigoli attributi si scrive il file SQL per l'inserimento nel database:

```
1
2     for(int i=0; i<list.size(); i++){
3       //System.out.println(list.size());
4       tmp = list.get(i);
5
6       .........
7
8       String sql = "INSERT INTO PATHOLOGY VALUES(
9       '"+tmp.id.substring(4)+"','"+
10      "'"+tmp.name.substring(6).replace("'", "\"")+
11      "'"+comment+def+subset+pre+post+","+tmp.level+");";
12      out.println(sql);
13
14      for(int j=0; j<tmp.synonym.size(); j++){
15        sql = "INSERT INTO SYNONYM VALUES (
16        '"+tmp.id.substring(4)+"','"+
17        tmp.synonym.get(j).substring(9).replace("'", "\"")+"');";
18        out.println(sql);
19      }
20
21      for(int j=0; j<tmp.xref.size(); j++){
22        sql = "INSERT INTO XREF VALUES (
23        '"+tmp.id.substring(4)+"','"+
24        tmp.xref.get(j).substring(6).replace("'", "\"")+"');";
25        out.println(sql);
26      }
27
28    }
```

3. Per concludere, si attuano le modifice opportune al file **PathologyRecover.java** Introdotto nella sezione **Progettazione e realizzazione del DB**. Il codice, che, poichè concettualmente semplice, non viene mostrato, si occupa della già affrontata interrogazione del web service offerto da Biomart e nella scrittura, a differenza dello script originale, delle sole insert per la tabella RELATED_TO[19]. Solo nel caso in cui si trattasse di patologie di cui non si è trovata corrispondenza nell'ontologia ma di cui comunque si vuol tener traccia (es Unclassificable Pathology) allora si preoccupa anche di inserire il loro record nella tabella PATHOLOGY. Inoltre, da non dimenticare, vi è anche la gestione a parte dei nodi "replicati" per i quali ancora "manualmente" è stato effettuata l'aggiunta nella tabella RELATED_TO. Si ritiene, dunque, conclusa la progettazione e sviluppo del database nella sua interezza e funzionalità. Nella prossima sezione verranno ampiamente discusse, invece, le modalità di accesso allo stesso mediante il web ed il loro relativo sviluppo.

[19]Si rimanda all'**Appendice C** per approfondimenti

2.4 Progettazione e realizzazione dell'interfaccia Web

Se da un lato lo sviluppo del database realizza e concretizza la ricerca efficente dei dati, da non sottovalutare è l'importanza della componente funzionale ma sopratutto di agevolezza introdotta senz'altro da un'interfaccia grafica. Tra le tante e possibili interfacce che potrebbero essere accompagnate allo sviluppo di un database la più facile ed accessibile al giorno d'oggi è rappresentata da quella web. La scelta effettuata per il progetto, infatti, è ricaduta proprio su quest'ultima, in virtù della sua più semplice accessibilità remota mediante browser, dell'assente necessità di librerie o componenti esterni preinstallati, e poichè ricade su paradigmi noti e consolidati in termini di scelte grafiche e modalità di interazione con l'utente. Inoltre non vi sono particolari problematiche legate alla privacy del database e/o di sicurezza poichè in conflitto con lo scopo stesso del progetto: rendere possibile un più semplice ed agevole accesso ai dati e favorire l'interoperabiltá.

2.4.1 Struttura e modalità di accesso

Per garantire un contesto all'interrogazione dei dati vera e propria, si è scelto di accennare la creazione un vero e proprio portale Web, per eventuali e possibili sviluppi futuri. Il portale sarà costituito da diverse sezioni elencate di seguito:

- **Home**: La sezione che offre la visione iniziale del portale con un'introduzione ed una lista di suggerimenti e nozioni fondamentali per l'utente piú spaesato.
- **UC Data Mining**: La sezione che offre le diverse modalità di interrogazione ed un'introduzione sommaria sulla struttura del database. Il *core* del portale nonché centro degli sforzi ad interesse esclusivo del progetto di tesi.
- **Related Works**: La sezione che sottopone l'insieme navigabile dei lavori correlati circa la sequenze e gli elementi ultraconservati.
- **About Us**: La sezione che enuclea le informazioni circa i gruppi di ricerca convolti nel progetto, ed altre informazioni di reperibilità.

Nel proseguo dello scritto non si scenderà nel dettaglio per descrivere ogni singola pagina del portale ed i tecnicismi informatici grazie i quali è stato possibile ottenere certi risultati poichè non strettamente inerenti e rilevanti al progetto di tesi. Si approfondiscono, invece, la struttura e le modalità scelte per l'interrogazione nella sezione **UC Data Mining**. La sezione, come pocanzi accennato, introdurrá innanzitutto la struttura del database per rendere il piú chiara possibile l'interazione col database e l'indagine più seria

e conscia, successivamente offrirà due gradi categorie di interrogazione: la prima, per gli utenti meno esperti, mediante una serie di *query prestrutturate*, la seconda secondo un'accesso completo al database mediante la possibilità di somministrare *query libere* al sistema. Per la prima categoria si è pensato di introdurre le seguenti *query prestrutturate* e di quindi altrettanti *forms*:

1. Possibilitá di conoscere tutte le informazioni relative una sequeza uc specificata.

2. Possibilitá di conoscere tutte le informazioni relative un gene specificato.

3. Possibilitá di conoscere tutte sequenze uc rilevate su di uno specifico cromosoma e comprese tra diverse *bp* specificate.

4. Possibilitá di conoscere tutte patologie correlate ad una sequeza uc **e tutti i suoi sottotipi** (Si notino le potenzialità introdotte dall'inserimento dell'ontologia).

5. Possibilitá di confrontare secondo *matching approssimato*[20] una sequenza nucleotidica con quelle delle uc presenti nel database ed ottenere una lista di possibili risultati elencati per *scores*.

6. Possibilitá di confrontare secondo *matching approssimato* una sequenza nucleotidica con quelle delle uc presenti nel database ed ottenere una lista di possibili risultati elencati per *scores* e filtrati secondo una patologia (ossia che siano necessariamente correlati a quella specifica patologia o ad un suo sottotipo).

Per la seconda categoria verrà invece introdotto un unico *form* la somministrazione delle query libere in SQL .

2.4.2 Cenni sulla realizzazione

Per la realizzazione del progetto di accessibilità esposto nel paragrafo precedente, si rende opportuna la conoscenza e l'utilizzo del linguaggio **html** per la strutturazione dei contenuti, **css** per la gestione del *leyout*, e del **php** per la gestione degli eventi lato server e la comunicazione con il database. Senza scendere troppo nel dettaglio si mettono in evidenza solo le strategie fondamentali per la realizzazione di alcuni dei punti elencati precedentemente.

Si prenda in considerazione, ad esempio, la possibilità di conoscere tutte patologie correlate ad una sequeza uc e tutti i suoi sottotipi. Sfruttando gli attributi delle patologie che rendono possibile la gerachizzazione, come già spiegato nei paragrafi inerenti il collegamento della patologia (nell'esempio "disease"), si potrebbe giungere al seguente SQL:

[20] Per matching approssimato si intende con possibilitá di errori

```sql

SELECT UC_NAME, NAME, MIN( LEVEL ) AS LEVEL
FROM UC
INNER JOIN
(

 SELECT ENSAMBL_GENE_ID, NAME,
 LEVEL FROM RELATED_TO
 INNER JOIN
 (

   SELECT ID, NAME, LEVEL
   FROM PATHOLOGY
   WHERE PREORDER >=
     (
      SELECT PREORDER
      FROM PATHOLOGY
      WHERE NAME = 'disease'
     )
   AND POSTORDER <=
     (
      SELECT POSTORDER
      FROM PATHOLOGY
      WHERE NAME = 'disease'
     )
   ORDER BY LEVEL

 ) AS B
 WHERE RELATED_TO.ID = B.ID
) AS C
WHERE UC.ENSAMBL_GENE_ID = C.ENSAMBL_GENE_ID
GROUP BY uc_name, name
ORDER BY LEVEL
```

In seguito alla somministrazione nel form da parte dell'utente del nome della patologia quindi, verrà inoltrata una richiesta *GET*[21] con i relativi parametri ad una pagina di risposta che li recupererà e connettendosi al database fornirà le risposte desiderate. Si prenda visione della sequenza indispensabile dei comandi in php presenti nella pagina di risposta:

```php

<?php
//recupero del paramentro pat presente nella richiesta GET
$pat = filter_input(INPUT_GET,"pat", FILTER_SANITIZE_STRING);

```

[21] Il metodo GET è un metodo del protocollo HTTP usato per ottenere il contenuto della risorsa indicata come URI

```
6     .... //codice di controllo, html etc..
7
8   //connessione al database
9   $con = mysql_connect("localhost","client","password");
10  if (!$con)
11    {
12    die('Could not connect: ' . mysql_error());
13    }
14  mysql_select_db("UCbase", \$con);
15
16    .... //codice di controllo, html etc..
17
18  //si stanzia ed esegue la query
19  $query = "SELECT UC_NAME,NAME,MIN(LEVEL)AS LEVEL
20  FROM UC
21  INNER JOIN (SELECT ENSAMBL_GENE_ID,NAME,LEVEL
22  FROM RELATED_TO INNER JOIN
23  (SELECT ID,NAME,LEVEL FROM PATHOLOGY WHERE PREORDER >=
24  (SELECT PREORDER FROM PATHOLOGY WHERE NAME='".\$pat."')
25  AND POSTORDER
26  <=(SELECT POSTORDER FROM PATHOLOGY WHERE NAME='".\$pat."')
27   ORDER BY LEVEL)AS B
28  WHERE RELATED_TO.ID = B.ID)AS C
29  WHERE UC.ENSAMBL_GENE_ID = C.ENSAMBL_GENE_ID
30  group by uc_name, name ORDER BY LEVEL";
31
32  $result = mysql_query($query);
33
34  //se ci sono risultati si crea la tabella
35  while($row = mysql_fetch_array($result, MYSQL_ASSOC))
36    {
37      if($i%2 == 0) echo "<tr class ='alt'>";
38      else echo "<tr>";
39
40      if($i==0){
41        foreach($row as $key => $value)
42        {
43        echo "<th>";
44        echo $key;
45        echo "</th>";
46        }
47        echo "</tr>";
48      }
49
50      if($i==0) echo "<tr>";
51      foreach($row as $x=>$x_value)
52      {
53        echo "<td>";
54        if($x == "UC_NAME"){
```

```
55          $uc_name = $x_value;
56          echo "<a href='result.php?uc_id=".
57            $uc_name."'>".$x_value."</a>";
58        }
59        else if($x == "UC_NAME"){
60          $uc_name = $x_value;
61          echo $x_value;
62        }
63        else if($x == "DEFINITION" && $x_value !=""){
64          $x_value = preg_replace(
65            '!(http|ftp|scp)(s)?:\/\/[a-zA-Z0-9.?&_=/]+!',
66            "<a target='_blank' href=\"\\0\">\\0</a>",$x_value);
67                echo $x_value;
68        }
69        else if($x == "SEQUENCE") {
70          echo strtolower(substr($x_value, 0, 13)."...");
71          echo "<a class='button' href='sequence.php?uc_id=".
72              $uc_name."'> Get Sequence!</a>"; }
73        else
74          echo $x_value;
75        echo "</td>";
76
77      }
78      echo "</tr>";
79      $i++;
80    }
81    .... //codice finale, di controllo, html etc..
82  ?>
```

Con lo stesso meccanismo, potranno essere realizzati anche i primi 3 punti. Per quanto riguarda invece gli ultimi due, non si implemeterà un algoritmo di confronto ma si utilizzerà il collaudato e rinomato *BLAST*[22][28] nello specifico la sua versione specializzata per il confronto di sequenze nucleotidiche *Blastn*.

Prima di ogni altra cosa bisogna creare ed indicizzare il piccolo database delle sequenze a partire dal file nel formato *FASTA*[23] usando il comando *makeblastdb* offerto sempre dal pacchetto scaricabile dal sito dell'NCBI [27].

```
./makeblastdb -in UC.fasta -dbtype nucl
-parse_seqids -out UCdb -title "Human UC"
```

[22]BLAST (Basic Local Alignment Search Tool, ovvero strumento di ricerca di allineamento locale) è un algoritmo usato per comparare le informazioni contenute nelle strutture biologiche primarie

[23]In bioinformatica, il formato FASTA è un formato testuale per la rappresentazione di sequenze nucleotidiche e peptidiche.

Successivamente si potrá utilizzare direttamente il comando blastn indicando il file contenente la query sempre nel formato FASTA ed il file di output:

```
./blastn -db UCdb -query query.fasta
-task blastn-short -out results.out
```

Il file di output potrà ad esempio essere:

```
Database: Human UC
           481 sequences; 126,007 total letters

Query= prova

Length=29
                                                              Score    E
Sequences producing significant alignments:                   (Bits)  Value

lcl|uc.0                                                      58.0   8e-12
lcl|uc.23                                                      20.3   1.8
.........                                                      ....   ....

>lcl|uc.0
Length=207

 Score = 58.0 bits (29),  Expect = 8e-12
 Identities = 29/29 (100%), Gaps = 0/29 (0%)
 Strand=Plus/Plus

Query  1   TCCACCGACAATGACCAGTTAGTCCTCAT  29
           |||||||||||||||||||||||||||||
Sbjct  1   TCCACCGACAATGACCAGTTAGTCCTCAT  29


>lcl|uc.23
Length=235

 Score = 20.3 bits (10),  Expect = 1.8
 Identities = 16/18 (89%), Gaps = 0/18 (0%)
 Strand=Plus/Minus

Query  12  TGACCAGTTAGTCCTCAT  29
           ||||||| | ||||||||
Sbjct  64  TGACCAGATGGTCCTCAT  47

........
```

Sia per la realizzazione del punto **4.** che del punto **5.** basterà recuperare per eventualmente poi rielaborare questi contenuti in php con una semplice riga di codice che inserisce nella variabile output un array di stringhe che costituscono l'output visto in precedenza:

```
exec( "
    cd /var/www/UCbase/ncbi-blast-2.2.27+/bin/;
    ./blastn -db UCdb -query query.fasta
    -task blastn-short 2>&1
    ", $output
  );
```

Si rammenta che, ovviamente, per il punto **5.** la rielaborazione dell'output non sarà solamente in termini grafici, ma si dovrà effettuare il filtraggio e l'ordinamento delle uc in funzione di quelle restituite da una query analoga a quanto visto per il risolvimento del punto **3.** : meccanicamente si eliminarà quelle sequenze uc non correlate a nessuna patologia e laddove presente la si indicherà; A parità di score (indicato da blastn), inoltre, la sequenza uc correlata alla patologia il cui livello è piú affine a quello della patologia indicata dovrà risultare prioritaria nell'ordinamento dei risultati. (Data la linearità concettuale, anche in questo caso, si rimanda all'archivio dei codici per approfondimenti)

Capitolo 3

Risultati

Concluso e raffinato il codice per la definizione del layout di ogni pagina, ed a seguito di ulteriori migliorie ed aggiunte applicate al modello di accessibilità descritto nel capitolo precedente, si presentano i risultati raggiunti. Si tratta di un portale dal design semplice,leggero e molto asciutto. In primis la homepage, accessibile attualmente all'indirizzo
http://www.dsb.unimo.it/ucbase:

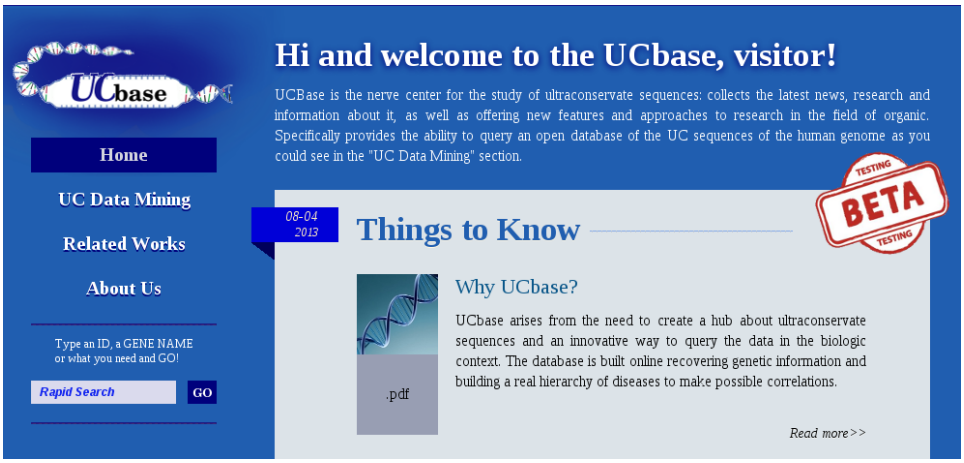


Fig. 3.1: UCbase: Pagina principale

Presenti in alto a sinistra i link fondamentali per il proseguo della navigazione. Si prenda visione delle altre pagine:

Fig. 3.2: UCbase: Related works

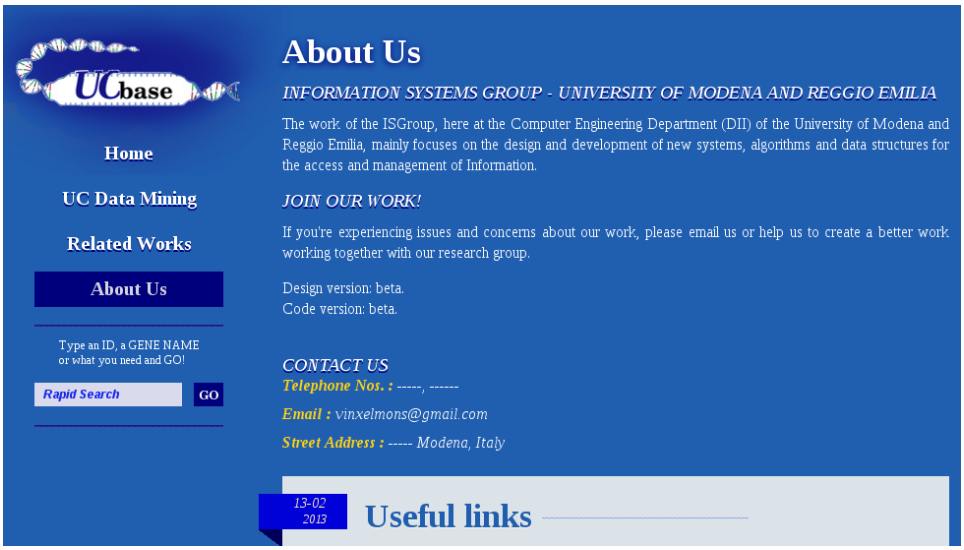


Fig. 3.3: UCbase: About us

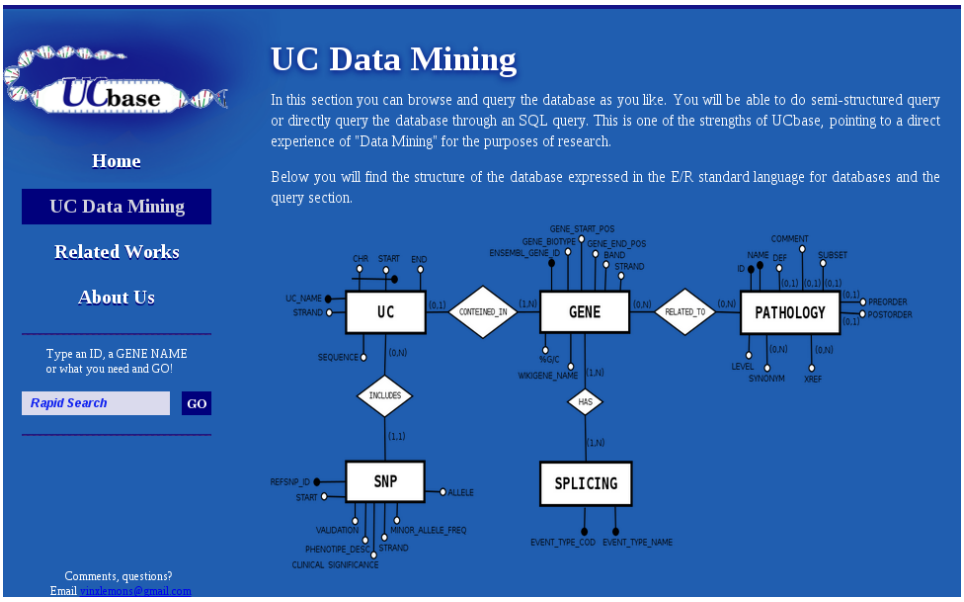


Fig. 3.4: UCbase: Uc data mining

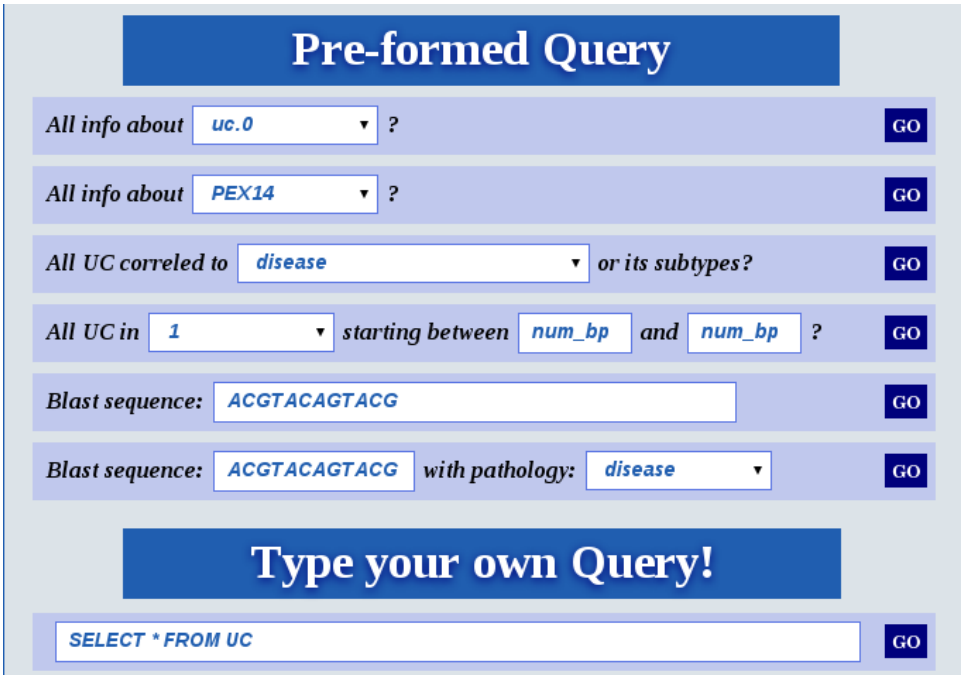


Fig. 3.5: UCbase: Query

Search Result

SELECT UC_NAME,NAME,MIN(LEVEL)AS LEVEL FROM UC INNER JOIN (SELECT ENSAMBL_GENE_ID,NAME,LEVEL FROM RELATED_TO INNER JOIN (SELECT ID,NAME,LEVEL FROM PATHOLOGY WHERE PREORDER >= (SELECT PREORDER FROM PATHOLOGY WHERE NAME='disease') AND POSTORDER <=(SELECT POSTORDER FROM PATHOLOGY WHERE NAME='disease') ORDER BY LEVEL)AS B WHERE RELATED_TO.ID = B.ID)AS C WHERE UC.ENSAMBL_GENE_ID = C.ENSAMBL_GENE_ID group by uc_name, name ORDER BY LEVEL

UC_NAME	NAME	LEVEL
uc.284	disease of cellular proliferation	1
uc.76	disease of cellular proliferation	1
uc.321	disease of cellular proliferation	1
uc.323	disease of cellular proliferation	1
uc.3	disease of cellular proliferation	1
uc.63	disease of cellular proliferation	1
uc.108	disease of cellular proliferation	1
uc.144	disease of cellular proliferation	1
uc.34	disease of cellular proliferation	1

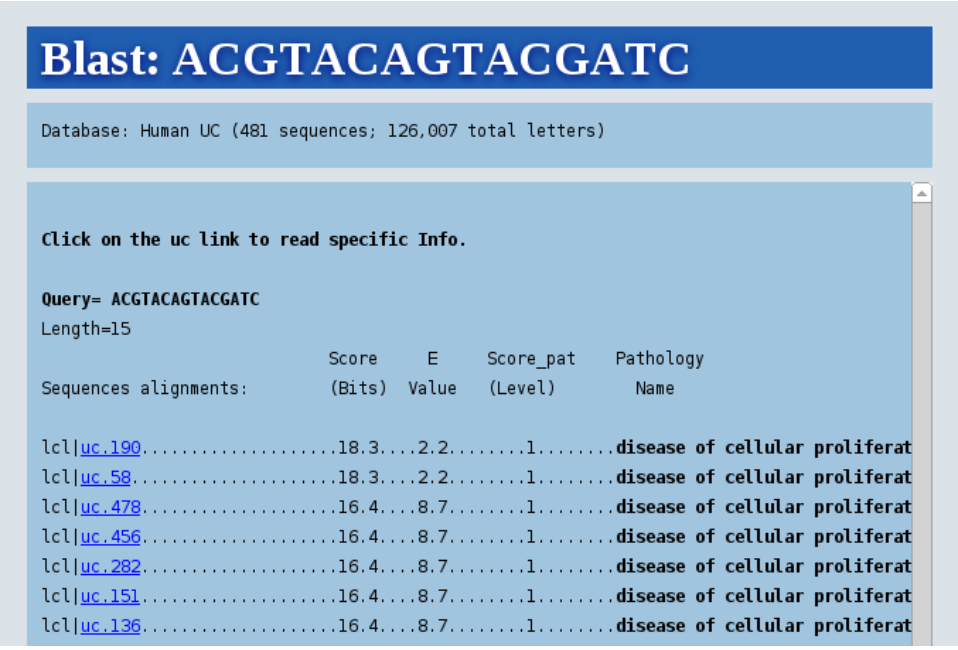
Fig. 3.6: UCbase: Risultati in formato tabellare

Search Result

uc.0

Uc_name	uc.0
Chr	1
Start	10597697
Ending	10597903
Strand	1
Sequence	TCCACCGACAATGACCAGTT... Get Sequence!
Ensambl_gene_id	ENSG00000142655

Fig. 3.7: UCbase: Risultati in formato schedario



Blast: ACGTACAGTACGATC

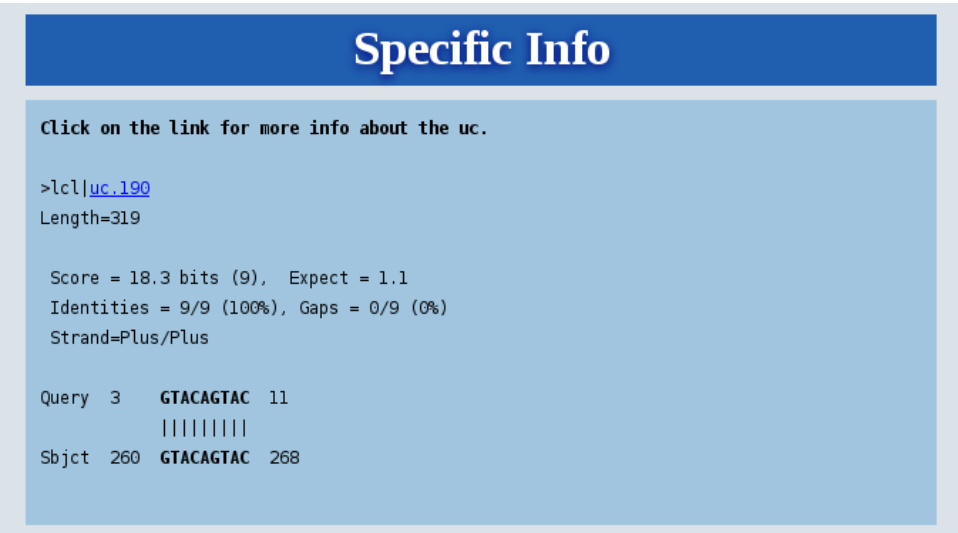
Database: Human UC (481 sequences; 126,007 total letters)

Click on the uc link to read specific Info.

Query= ACGTACAGTACGATC
Length=15

Sequences alignments:	Score (Bits)	E Value	Score_pat (Level)	Pathology Name
lcl|uc.190	18.3	2.2	1	**disease of cellular proliferat**
lcl|uc.58	18.3	2.2	1	**disease of cellular proliferat**
lcl|uc.478	16.4	8.7	1	**disease of cellular proliferat**
lcl|uc.456	16.4	8.7	1	**disease of cellular proliferat**
lcl|uc.282	16.4	8.7	1	**disease of cellular proliferat**
lcl|uc.151	16.4	8.7	1	**disease of cellular proliferat**
lcl|uc.136	16.4	8.7	1	**disease of cellular proliferat**

Fig. 3.8: UCbase: Risultati blastn



Specific Info

Click on the link for more info about the uc.

```
>lcl|uc.190
Length=319

 Score = 18.3 bits (9),  Expect = 1.1
 Identities = 9/9 (100%), Gaps = 0/9 (0%)
 Strand=Plus/Plus

Query  3    GTACAGTAC  11
            |||||||||
Sbjct  260  GTACAGTAC  268
```

Fig. 3.9: UCbase: Informazioni specifiche

La pagina che raggruppa i contenuti relativi agli elementi ultraconservati *Related works* si presenta come mostrato in **Fig.3.2**. Segue lo stesso layout grafico anche la pagine *About Us* (**Fig.3.3**).
Si prenda visione, infine, della sezione più importante per la ricerca esplorativa sui dati: la pagina *UC Data Mining* (**Fig.3.4**). La pagina d'apprima illustra la struttura del database poi fornisce come una serie di form compilabili le opzioni di investigazione illustrate nel capitolo precedente (**Fig.3.5**). Sono stati implementate, poi, 3 grandi tipologie di layout per l'output dei dati: La prima in formato tabellare per rispondere ad una query generica come quelle provenienti dal form delle query libere oppure dal form 3. e 4. dal gruppo delle query prestrutturate. (**Fig.3.6**) La seconda, invece, in un formato un po' più da "schedario" per le informazioni circa i campi di un singolo record, quindi per rispondere alle query prestrutturate 1. e 2. (**Fig.3.7**)
La terza ed ultima categoria di visualizzazione utilizzate per le query prestrutturate 5. e 6., infine, in cui viene lasciato inalterato lo scheletro di output di blastn, ma aggiunta, come nel caso mostrato in figura (**Fig.3.8**), qualche informazione aggiuntiva.
Si rimanda nuovamente al sito *http://www.dsb.unimo.it/ucbase* per scoprire ed approfondire appieno le potenzialità dell'interfaccia.

Capitolo 4

Conclusioni e sviluppi futuri

A conclusione dell'elaborato, successivamente alla realizzazione del database ed all'interfaccia web per la sua accessibilità, risulta quantomeno appropriato fornire sommarie conclusioni circa l'effettivo raggiungimento degli obiettivi che ci si era posti all'inizio del percorso di tesi, le difficoltà incontrate durante il percorso e consapevoli quanto umili commenti circa le riconosciute mancanze e grossolanità del progetto sicuramente migliorabili in futuro. In primis si premette che, a fronte di un risicato tempo di studio, si é realizzato un progetto dalle dimensioni rilevanti, le cui basi tematiche necessitano, perl'altro, di ulteriori e necessari approfondimenti. Per cui si ritenga l'elaborato frutto di un attento e laborioso lavoro di riassunto, che non lascia evincere le difficoltà con le quali ci si é scontrati, né i tentativi e scelte esplorative effettuate nelle più disparate direzioni e mai più realizzate. Tuttavia, la totalità degli obiettivi enunciati nel primo capitolo, sono stati portati a termine e dunque possono essere considerate risolte le problematiche che quest'ultimi erano tesi a dirimere. Nonostante questo, numerosi sono gli aspetti migliorabili del progetto di cui si è a consapevolezza e si rimanda in un futuro:

- **Il linguaggio java**: Una delle prime pecche risulta essere la scelta del linguaggio, per il quale si sarebbe dovuto prediligere un linguaggio per la prototipizzazione veloce come un linguaggio dinamico (ad esempio il Python).

- **Ridondanza nel database e linking semi-automatico delle patologie**: Un importante e nuovo livello di pulizia dei dati riguarderebbe la riprogettazione di un linking automatico con l'ontologia di riferimento (attualmente si ricorda che il processo richiede l'intervento di un esperto in materia per le patologie il cui nome non combacia direttamente con quello dell'ontologia) e di quella per evitare la ridondanza all'interno del database (introdotta attualmente per semplificare il processo di gestione delle gerarchie attraverso l'uso un'albero anzichè di un grafo)

- **Impossibilità di aggiornamento dei dati**: Attualmente il progetto non prevede la possibilità di aggiornare dati già esistenti ma solo di creare gli script indispensabili alla creazione ed all'inserimento di dati in un database ex-novo. Si suggerisce anche questo punto come una delle possibili migliorie effettuabili.

- **Semplicità dell'interfaccia Web**: Infinitamente migliorabile, infine, l'interfaccia web, sia in termini di design sia in termini di funzionalità offerte.

Sottolineate le mancanze e possibili migliorie, non rimane che riavvalorare, in antitesi, quanto, invece, risulta essere un punto di forza del progetto:

- **Open Database e linking dell'HDO**: Nell'ambito degli elementi e delle sequenze nucleotidiche ultraconservate, il servizio offerto per l'interrogazione del database, costituisce uno dei pochissimi esempi di libero accesso ed investicazione in ambito biologico e rappresenta un avanguardia sia per modalitá di interrogazione mediante SQL, sia per la struttura stessa del database che collega direttamente numerose informazioni (comprese quelle di tipo **gerarchico**) sulle patologie correlate ad un gene e quindi ad alcuni elementi ultraconservati.

- **L'interfaccia Web**: Offrire il sopracitato servizio mediante il Web risulta un'ulteriore agevolazione rivolta all'utenza, attraverso un'interfaccia semplice e veloce, funzionale ed efficente, che non comporta nessun prerequisito o difficoltà iniziale.

- **Recupero automatico dei dati**: Con poche e semplici operazioni (e pochi file di supporto) è possibile, anche per i poco esperti, ottenere informazioni **aggiornate** sulle metainformazioni inerenti le sequenze ultraconservate e, quindi, costruire il database, senza il minimo sforzo ed in tempi brevi.

Appendice A

Glossario di base di biologia molecolare

Di seguito vengono riportate alcune informazioni utili (citazioni e definizioni) inerenti.

Biologia Molecolare: *" La biologia molecolare è una branca della biologia che studia gli esseri viventi a livello dei meccanismi molecolari alla base della loro fisiologia, concentrandosi in particolare sulle interazioni tra le macromolecole, ovvero proteine e acidi nucleici (DNA e RNA). Per biologia molecolare si intendono spesso una serie di tecniche che consentono la rilevazione, l'analisi, la manipolazione, l'amplificazione (PCR) e la copia (clonaggio) degli acidi nucleici."* [23]

DNA e RNA: *" L'acido deossiribonucleico (DNA) costituisce, con l'acido ribonucleico (RNA), la classe di polimeri informazionali definita 'acidi nucleici', componenti fondamentali delle strutture viventi. Tra tutte le molecole biologiche soltanto gli acidi nucleici possiedono la potenzialità di autoduplicazione che permette la replicazione e la trascrizione dell'informazione chimica in essi contenuta. Altri polimeri costituenti fondamentali delle strutture biologiche, quali le proteine e i polisaccaridi, sono privi di tale capacità e, pur dotati di comparabile complessità polimerica strutturale e informazionale, di ricchissima informazione chimica e di grande flessibilità funzionale, non sono in grado di rivestire un ruolo genetico. Questo ruolo è legato alla particolare organizzazione strutturale degli acidi nucleici e alle proprietà chimiche, fisiche e topologiche sia dei suoi costituenti che delle macromolecole considerate nel loro insieme. "* [24]

Cromosoma: *" Nome dato da W. Waldeyer nel 1888 ai piccoli corpi intensamente colorabili, in genere di forma bastoncellare, visibili nel nucleo della cellula durante la mitosi. Secondo la teoria cromosomica , dimostrata*

da T.H. Morgan con ricerche su Drosophila melanogaster, i cromosomi sono le strutture essenziali dell'eredità. " [24]

Banda Cromosomica: *" La denaturazione o la digestione enzimatica della cromatina, seguite dall'incorporazione di un colorante specifico per il DNA, fa sì che nei cromosomi mitotici di organismi complessi si individui una serie di bande alternantesi in senso trasversale fino a un totale di 400, 550 o 850 visibili al microscopio ottico. Le si ottengono sottoponendo i cromosomi a digestione con tripsina prima della colorazione con Giemsa; si formano bande scure (bande G) e chiare (bande G negative). "* [24]

Base azotata: *" In biochimica, per base azotata si intende una delle cinque basi che compongono i nucleotidi del DNA e dell'RNA. Si distinguono in purine e pirimidine. Purine: Adenina e Guanina Pirimidine: Citosina, Timina e Uracile Nel DNA si trovano entrambe le purine e, tra le pirimidine, citosina e timina. Nell'RNA la timina è sostituita dall'uracile. Nel DNA le basi si accoppiano a due a due con legami a idrogeno, mentre nell'RNA, essendo questo una catena singola, non sono legate tra loro. DNA: adenina-timina e citosina-guanina. RNA: adenina-uracile e citosina-guanina. In chimica, una base azotata è un qualsiasi composto che manifesta proprietà basiche per via della presenza di un doppietto elettronico non condiviso su un atomo di azoto, ad esempio l'ammoniaca e le ammine. "* [23]

Sequenza Nucleotidica: *In biologia molecolare, la successione ordinata di nucleotidi del DNA. Ogni nucleotide è formato da una molecola di desossiribosio, un gruppo fosfato e una base azotata. In particolare, sono le basi azotate a determinare la sequenza nucleotidica specifica dell'informazione genetica."* [24]

Gene: *" Nome proposto da W. Johannsen (1909) per indicare l'unità ereditaria scoperta da G. Mendel; è materializzato nella cellula da un segmento di una molecola di DNA, in cui l'ordine di successione dei quattro tipi di nucleotidi determina, secondo la legge di corrispondenza espressa dal codice genetico, l'ordine di successione dei diversi tipi di amminoacido nel corrispondente polipeptide. Mentre nelle cellule procariotiche ogni gene è espresso da un segmento continuo di DNA, di regola i geni delle cellule eucariotiche sono in pezzi: segmenti di DNA portatori di informazione (esoni) si alternano a segmenti non informazionali (introni); la sintesi del polipeptide avviene in tal caso mediante la trascrizione di un tratto di molecola di DNA che comprende tutti gli esoni e i relativi introni. La lunga molecola di RNA così sintetizzata subisce quindi un processo di maturazione, che consiste nell'escissione degli introni, con la formazione della molecola di mRNA necessaria per l'ordinario processo di traduzione dell'informazione.[...] "* [24]

Base Pair: *" Le coppie di basi o paia di basi (abbreviate come pb o, dall'inglese base pair, bp o bps) sono comunemente utilizzate come misura della lunghezza fisica di sequenze di acidi nucleici a doppio filamento. Spesso, data la grande dimensione dei genomi, viene utilizzata l'unità kbp (k sta per kilo), pari a mille paia di basi. Il numero di paia di basi rappresenta il numero di coppie di basi azotate che contiene il filamento in analisi. Ogni paio di basi è tenuto insieme attraverso legami idrogeno che si instaurano tra le molecole, ricche di azoto. "* [23]

SNP: *" Un polimorfismo a singolo nucleotide (spesso definito in inglese Single Nucleotide Polymorphism o SNP, pronunciato snip) è un polimorfismo, cioè una variazione, del materiale genico a carico di un unico nucleotide, tale per cui l'allele polimorfico risulta presente nella popolazione in una proporzione superiore all'1%. Al di sotto di tale soglia si è soliti parlare di mutazione. Ad esempio, se le sequenze individuate in due pazienti sono AAGCCTA e AAGCTTA, è presente uno SNP che differenzia i due alleli C e T. "* [23]

Splicing: *" In biologia molecolare e in genetica, splicing è una modificazione del nascente pre-mRNA che avviene insieme o dopo la trascrizione (biologia), nella quale gli introni sono rimossi e gli esoni vengono uniti. La cosa è necessaria per il tipico RNA messaggero prima che possa essere usato per produrre una corretta proteina tramite la traduzione (biologia) o sintesi proteica. "* [23]

Allele: *" In genetica si definisce allele o fattore ogni variante di sequenza di un gene o di un locus genico (generalmente un gruppo di geni). Il genotipo di un individuo relativamente ad un gene è il corredo di alleli che egli si trova a possedere. In un organismo diploide, in cui sono presenti due copie di ogni cromosoma (i cosiddetti cromosomi omologhi), il genotipo è dunque costituito da due alleli. Se un organismo possiede per lo stesso gene due alleli identici allora si definisce omozigote per quel locus genico, mentre se sono differenti viene detto eterozigote. In realtà per dati geni si possono riscontrare diverse varianti alleliche tanto che per molti geni noti è stata messa in discussione la terminologia allele mutante perché le varietà alleliche non necessariamente portano ad uno svantaggio selettivo. "* [23]

Strand: *" In genetica, ciascuna delle due catene di una molecola di DNA, normalmente costituita da una coppia di filamenti fra loro complementari. Nel processo di sintesi del DNA si chiama filamento guida (leading strand) quello che è sintetizzato in maniera continua, in direzione 5'3', e filamento tardivo (lagging strand) quello sintetizzato in direzione opposta 3'5', in maniera discontinua sotto forma di brevi frammenti successivamente legati da legami covalenti. "* [24]

Appendice B

Elementi ultraconservati nel genoma umano

Di seguito vengono riportate alcune informazioni utili (citazioni e definizioni) inerenti.

Elemento ultraconservato:
Un elemento ultraconservato (UCE: ultraconserved element) è una regione di DNA che è identica in almeno due specie. Uno dei primi studi sugli UCEs mostra che sequenze della lunghezza di 200 nucleotidi o più del DNA umano si sono conservate interamente (sequenze identiche di acidi nucleici) in entrambi ratti e topi. Rispetto alla maggioranza di DNA non codificante, alcuni elementi ultraconservati sono stati rilevati come attivi a livello trascrizionale, originando molecole di RNA non codificante. [23]

Estratto tradotto dall'articolo originale Ultraconserved Elements in the Human Genome [22] :
Ci sono 481 segmenti più lunghi di 200 bp che sono conservati (100% di identità senza inserimenti né cancellazioni) tra regioni ortologhe del genoma umano, del topo e del ratto. Quasi tutti questi segmenti sono anche *ultraconservati* nel genoma del pollo e del cane con un 95% e 99% di identità, rispettivamente. Molti sono anche significativamente conservati nel pesce. Questi elementi **utraconservati** del genoma umano sono spesso situati o sovrapposti agli esoni nei geni coinvolti nel processamento dell'RNA o in introni o vicino a geni coinvolti nella regolazione della trascrizione e sviluppo. Insieme a più di 5.000 sequenze di oltre 100bp che sono conservate tra i tre mammiferi sequenziati, questi rappresentano una classe di elementi genetici le cui funzioni ed origini evolutive sono ancora da determinare, ma che sono più altamente conservate tra queste specie che le proteine, e sembrano essere essenziali per l'ontogenesi dei mammiferi e altri vertebrati.
Anche se solo circa l'1,2% del genoma umano appare codificare proteine, è

stato stimato che circa il 5% è più conservato di quanto previsto dall'evoluzione naturale dopo la divisione con i roditori. Studi hanno evidenziato segmenti non codificanti specifici nel genoma umano che sembrano essere in fase di selezione, con una conservazione del 70% o 80% di identità con il topo su più di 100bp. Uno studio di questi elementi sul cromosoma 21 dell'uomo ha scoperto che quelli che erano altamente conservati in più specie contenevano un numero significante di elementi non codificanti. Risultati simili sono stati trovati confrontando l'uomo, il topo ed il ratto in uno studio sulla regione cromosomica contenente il gene CFTR (1.8 Mbp) e in uno studio funzionale sul locus SIM2 in molti mammiferi.

Appendice C

Archivio dei Codici

Di seguito, vengono riportati, in ordine di apparizione, tutti i codici di cui, a volte, si è mostratosolo gli scorci più importanti. Si rammenta che per i codici modificati nel corso dell'elaborato verrà riportata solo l'ultima versione.

C.1 Uc_biomaRt_pathology2.java

```
library("biomaRt")
ensembl = useMart("ensembl",dataset="hsapiens_gene_ensembl")

options(width=120)
attributesR=c('ensembl_gene_id','mim_morbid_description','pathology','percentage_gc_content')
filtersR = c('chromosome_name','start','end')

out.list<-list()
for (i in 1:length(uc2009[,1])){
      valuesR<-list(uc2009[i,2],uc2009[i,3],uc2009[i,4])
      out<-getBM(attributes=attributesR,filters=filtersR,values=valuesR,mart=ensembl,uniqueRows=F)
      if (nrow(out)!=0){
       out<-data.frame(cbind(uc2009[i,1],out))
         colnames(out)[1]<-"uc"
      }
      else{
        out.names<-colnames(out)
        nulli<-as.data.frame(matrix(rep(NA,ncol(out)),nrow=1))
        out<-data.frame(cbind(uc2009[i,1],nulli))
        colnames(out)<-c("uc",out.names)
      }

      if(i==1)
          write.table(out,"pathology.csv",append=FALSE,col.names=TRUE,quote=FALSE,row.names=FALSE,sep=";")
      else
          write.table(out,"pathology.csv",append=TRUE,col.names=FALSE,quote=FALSE,row.names=FALSE,sep=";")
```

```
        out.list[[i]]<-out
        print(uc2009[i,1])
        print(out)
}
```

C.2 Tabelle.sql

```sql
CREATE TABLE GENE
(
  ENSAMBL_GENE_ID VARCHAR(20),
  GENE_BYOTIPE VARCHAR(30),
  GENE_START_POS INTEGER,
  GENE_END_POS INTEGER,
  BAND VARCHAR(10),
  STRAND INTEGER,
  WIKIGENE_NAME VARCHAR(30),
  G_C_PERC VARCHAR(10),
  PRIMARY KEY(ENSAMBL_GENE_ID)
);

CREATE TABLE UC
(
  UC_NAME VARCHAR(10) UNIQUE NOT NULL,
  CHR VARCHAR(5),
  START INTEGER,
  ENDING INTEGER,
  STRAND INTEGER,
  SEQUENCE VARCHAR(1000) NOT NULL,
  ENSAMBL_GENE_ID VARCHAR(20),
  PRIMARY KEY (CHR,START),
  FOREIGN KEY (ENSAMBL_GENE_ID) REFERENCES GENE
    ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE SNP
(
  REFSNP_ID VARCHAR(30),
  START INTEGER,
  ALLELE VARCHAR(10),
  VALIDATION VARCHAR(30),
  MINOR_ALLELE_FREQ FLOAT(10),
  PHENOTIPE_DESC VARCHAR(30),
  CLINICAL_SIGNIFIANCE VARCHAR(30),
  STRAND INTEGER,
  CHR VARCHAR(5),
  START_UC INTEGER,
  PRIMARY KEY (REFSNP_ID),
  FOREIGN KEY(CHR, START_UC) REFERENCES UC (CHR,START)
    ON DELETE CASCADE ON UPDATE CASCADE
);


CREATE TABLE SPLICING
```

```
47  (
48    EVENT_TYPE_COD VARCHAR(8),
49    EVENT_TYPE_NAME VARCHAR(30) UNIQUE NOT NULL,
50    PRIMARY KEY(EVENT_TYPE_COD)
51  );
52
53  CREATE TABLE HAS
54  (
55    EVENT_TYPE_COD VARCHAR(8),
56    ENSAMBL_GENE_ID VARCHAR(20),
57    PRIMARY KEY(EVENT_TYPE_COD, ENSAMBL_GENE_ID),
58    FOREIGN KEY(EVENT_TYPE_COD) REFERENCES SPLICING
59      ON DELETE CASCADE ON UPDATE CASCADE,
60    FOREIGN KEY(ENSAMBL_GENE_ID) REFERENCES GENE
61      ON DELETE CASCADE ON UPDATE CASCADE
62  );
63
64  CREATE TABLE PATHOLOGY
65  (
66    ID VARCHAR(15),
67    NAME VARCHAR(100) NOT NULL,
68    COMMENT VARCHAR(200),
69    DEFINITION VARCHAR(500),
70    SUBSET VARCHAR(100),
71    PREORDER INTEGER,
72    POSTORDER INTEGER,
73    LEVEL INTEGER,
74    PRIMARY KEY(ID)
75  );
76
77  CREATE TABLE RELATED_TO
78  (
79    ID VARCHAR(15),
80    ENSAMBL_GENE_ID VARCHAR(20),
81    PRIMARY KEY(ID, ENSAMBL_GENE_ID),
82    FOREIGN KEY(ID) REFERENCES PATHOLOGY
83      ON DELETE CASCADE ON UPDATE CASCADE,
84    FOREIGN KEY(ENSAMBL_GENE_ID) REFERENCES GENE
85      ON DELETE CASCADE ON UPDATE CASCADE
86
87  );
88
89  CREATE TABLE SYNONYM
90  (
91    ID VARCHAR(15),
92    NAME VARCHAR(100),
93    PRIMARY KEY(ID, NAME),
94    FOREIGN KEY(ID) REFERENCES PATHOLOGY
95      ON DELETE CASCADE ON UPDATE CASCADE
96  );
97
98  CREATE TABLE XREF
99  (
100   ID VARCHAR(15),
```

```
  REF VARCHAR(100),
  PRIMARY KEY(ID, REF),
  FOREIGN KEY(ID) REFERENCES PATHOLOGY
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

C.3 BiomartMain.java

```
import java.net.*;
import java.io.*;

public class BiomartMain {
  public static void main(String[] args) throws Exception {

    GeneRecover gene = new GeneRecover();
    SnpRecover snp = new SnpRecover();
    SplicingRecover spli = new SplicingRecover();
    PathologyRecover path = new PathologyRecover();
  }

}
```

C.4 GeneRecover.java

```
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.net.URL;
import java.net.URLEncoder;
import java.util.StringTokenizer;

public class GeneRecover {
  //dati per recupero info uc
  private final int num_lines = 481;
  private String[] chr = new String[num_lines];
  private String[] bp_start = new String[num_lines];
  private String[] bp_end = new String[num_lines];
  private String[] strand_uc = new String[num_lines];
  private String[] sequence = new String[num_lines];
  private String[] uc_name = new String[num_lines];
  //dati per tabelle
  private String ensambl_gene_id, gene_byotipe, gene_start_pos,
          gene_end_pos, band, strand, wikigene_name, g_c_perc, sql_gene,
              sql_uc;

  public GeneRecover()throws Exception{
    //leggo dal file le info
    try
        {
      //tsv file containing data
```

```
28            String strFile_uc = "/home/uc_coordinates_hg19_reference_file.
                  csv";
29            String strFile_seq = "/home/ultraconservate/UC.txt";
30
31
32            //create BufferedReader to read tsv file e per scrivere i file
                  sql
33            //delle tabelle "UC" e "GENE".
34            BufferedReader br = new BufferedReader( new FileReader(
                  strFile_uc));
35            BufferedReader br2 = new BufferedReader( new FileReader(
                  strFile_seq));
36            FileOutputStream sql_file = new FileOutputStream("/home/kira/
                  Scrivania/tesi/uc_gene_insert.sql");
37            PrintStream out = new PrintStream(sql_file);
38
39            String strLine = "";
40            //recupero sequenze uc
41            int i = 1, j=0, k=0;
42            while(i <= num_lines*2 ){
43              strLine = br2.readLine();
44              if(i%2 == 0) {sequence[j] = strLine; j++;}
45              else {uc_name[k] = strLine.substring(1); k++;}
46              i++;
47            }
48
49            StringTokenizer st = null;
50            int lineNumber = 0;
51            br.readLine(); //la prima riga contenente meta-info viene
                  scartata cosÃň
52            //read comma separated file line by line
53            while( (strLine = br.readLine()) != null )
54            {
55                //break comma separated line using ","
56                st = new StringTokenizer(strLine, ",");
57                String uc = st.nextToken();
58                chr[lineNumber] = st.nextToken();
59                //togliamo fastidiosi apici
60                chr[lineNumber] = chr[lineNumber].replace("\"", "");
61                bp_start[lineNumber] = st.nextToken();
62                bp_end[lineNumber] = st.nextToken();
63                strand_uc[lineNumber] = st.nextToken();
64                System.out.println("Sto recuperando "+uc+": "+chr[lineNumber
                      ]+" "+bp_start[lineNumber]+" "+bp_end[lineNumber]);
65
66
67                String myxml = "<Query virtualSchemaName = \"default\"
                      formatter = \"CSV\" header = \"1\" "
68                +"uniqueRows = \"0\" count = \"\" datasetConfigVersion =
                      \"0.6\" >"
69                +"<Dataset name = \"hsapiens_gene_ensembl\" interface = \"
                      default\" >"
70                +"<Filter name = \"chromosome_name\" value = \""+chr[
                      lineNumber]+"\"/>"
```

```
71                  +"<Filter name = \"start\" value = \""+bp_start[lineNumber]+"
                        \"/>"
72                  +"<Filter name = \"end\" value = \""+bp_end[lineNumber]+"\"/>
                        "
73                  +"<Attribute name = \"wikigene_name\" />"
74                  +"<Attribute name = \"ensembl_gene_id\" />"
75                  +"<Attribute name = \"gene_biotype\" />"
76                  +"<Attribute name = \"start_position\" />"
77                  +"<Attribute name = \"end_position\" />"
78                  +"<Attribute name = \"band\" />"
79                  +"<Attribute name = \"strand\" />"
80                  +"<Attribute name = \"percentage_gc_content\" />"
81                  +"</Dataset>"
82                  +"</Query>";
83                  //System.out.println(myxml);
84                  String encoded = URLEncoder.encode(myxml, "utf-8");
85                  //System.out.println(encoded.length());
86                  URL url = new URL("http://www.biomart.org/biomart/
                        martservice?query="+encoded);
87                  InputStream response = url.openStream();
88                  BufferedReader reader = new BufferedReader(new
                        InputStreamReader(response));
89                  i=0;
90                  for (String line; ;i++) {//nel caso non ci sia nessuna
                        risposta devo comunque inserire l'uc
91                    if((line = reader.readLine()) == null && (i == 1)) {
92                        sql_uc = "INSERT IGNORE INTO UC VALUES('"+uc_name[
                              lineNumber]+"','"+ chr[lineNumber]+"','"+ bp_start[
                              lineNumber]+
93                        ","+ bp_end[lineNumber]+","+strand_uc[lineNumber]+",'"
                              +sequence[lineNumber]+"',null);";
94                        System.out.println(sql_uc);
95                        out.println(sql_uc);
96                        break;
97                    }
98                    else if((i!=0) && (line != null)){//se ÃÍ la prima riga ->
                          scarto, altrimenti scrivo su file sql
99                      line.replace(",,",",null,");
100                     if(line.charAt(0) == ',') line = "null"+ line;
101                     System.out.println(line);
102                     st = new StringTokenizer(line, ",");
103
104                     wikigene_name = st.nextToken();
105                     if(!wikigene_name.equals("null")) wikigene_name = "'"+
                            wikigene_name+"'";
106                     ensambl_gene_id = st.nextToken();
107                     if(!ensambl_gene_id.equals("null")) ensambl_gene_id = "'"
                            +ensambl_gene_id+"'";
108                     gene_byotipe = st.nextToken();
109                     if(!gene_byotipe.equals("null")) gene_byotipe = "'"+
                            gene_byotipe+"'";
110                     gene_start_pos = st.nextToken();
111                     gene_end_pos = st.nextToken();
112                     band = st.nextToken();
```

```
                    if(!band.equals("null")) band = "'"+band+"'";
                    strand = st.nextToken();
                    if(!strand.equals("null")) strand = "'"+strand+"'";
                    g_c_perc = st.nextToken();

                    sql_gene = "INSERT IGNORE INTO GENE VALUES("+ensambl_gene_id+","+gene_byotipe+","+
                    gene_start_pos+","+gene_end_pos+","+band+","+strand+","+wikigene_name+","
                    +g_c_perc+");";
                    System.out.println(sql_gene);
                    out.println(sql_gene);

                      sql_uc = "INSERT IGNORE INTO UC VALUES('"+uc_name[lineNumber]+"','"+ chr[lineNumber]+"',"+ bp_start[lineNumber]+
                      ","+ bp_end[lineNumber]+","+strand_uc[lineNumber]+",'"+sequence[lineNumber]+"',"
                        + ensambl_gene_id +");";
                     System.out.println(sql_uc);
                     out.println(sql_uc);
                  }
                  else if(line == null) break;
                  System.out.println();

                }
                reader.close();
                lineNumber++;
            }
        }
        catch(Exception e)
        {
                System.out.println("Exception: " + e);
        }
  }

}
```

C.5 SplicingRecover.java

```
import java.io.BufferedReader;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.net.URL;
import java.net.URLEncoder;
import java.util.StringTokenizer;

public class SplicingRecover {
  private final int num_lines = 481;
  private String[] chr = new String[num_lines];
```

```
14    private String[] bp_start = new String[num_lines];
15    private String[] bp_end = new String[num_lines];
16
17    private String splicing_cod, splicing_name, ensambl_gene_id, sql_has,
          sql_splicing;
18
19    public SplicingRecover()throws Exception{
20      //leggo dal file le info
21      try
22          {
23        //tsv file containing data
24            String strFile = "/home/kira/Scrivania/tesi/
                uc_coordinates_hg19_reference_file.csv";
25            //sql output file
26            FileOutputStream sql_file = new FileOutputStream("/home/
                splicing_has_insert.sql");
27            PrintStream out = new PrintStream(sql_file);
28
29            //create BufferedReader to read tsv file
30            BufferedReader br = new BufferedReader( new FileReader(strFile
                ));
31            String strLine = "";
32            StringTokenizer st = null;
33            int lineNumber = 0;
34            br.readLine(); //la prima riga contenente meta-info viene
                scartata cosÃň
35            //read comma separated file line by line
36            while( (strLine = br.readLine()) != null )
37            {
38                //break comma separated line using ","
39                st = new StringTokenizer(strLine, ",");
40                String uc = st.nextToken();
41                chr[lineNumber] = st.nextToken();
42                chr[lineNumber] = chr[lineNumber].replace("\"", "");//
                    togliamo fastidiosi apici
43                bp_start[lineNumber] = st.nextToken();
44                bp_end[lineNumber] = st.nextToken();
45                System.out.println("Sto recuperando "+uc+": "+chr[lineNumber
                    ]+" "+bp_start[lineNumber]+" "+bp_end[lineNumber]);
46
47
48                String myxml = "<Query virtualSchemaName = \"default\"
                    formatter = \"CSV\" header = \"1\" "
49                +"uniqueRows = \"0\" count = \"\" datasetConfigVersion =
                    \"0.6\" >"
50                +"<Dataset name = \"hsapiens_gene_ensembl\" interface = \"
                    default\" >"
51                +"<Filter name = \"chromosome_name\" value = \""+chr[
                    lineNumber]+"\"/>"
52                +"<Filter name = \"start\" value = \""+bp_start[lineNumber]+"
                    \"/>"
53                +"<Filter name = \"end\" value = \""+bp_end[lineNumber]+"\"/>
                    "
54                +"<Attribute name = \"ensembl_gene_id\" />"
```

```
55                +"<Attribute name = \"splicing_event_type\" />"
56                +"<Attribute name = \"name_106\" />"
57                +"</Dataset>"
58                +"</Query>";
59                //System.out.println(myxml);
60                String encoded = URLEncoder.encode(myxml, "utf-8");
61                //System.out.println(encoded.length());
62                URL url = new URL("http://www.biomart.org/biomart/
                      martservice?query="+encoded);
63                InputStream response = url.openStream();
64                BufferedReader reader = new BufferedReader(new
                      InputStreamReader(response));
65                int i=0;
66                for (String line; (line = reader.readLine()) != null;) {
67                  if(i!=0){//se ÃÍ la prima riga -> scarto
68                    while(line != (line = line.replace(",,",",null,")));
69                    line = line.replace("'", "\"");
70                    if(line.charAt(0) == ',') line = "null"+ line;
71                    if(line.charAt(line.length()-1) == ',') line = line + "
                          null";
72                    st = new StringTokenizer(line, ",");
73                    System.out.println(line);
74
75                    ensambl_gene_id = st.nextToken();
76                    if(!ensambl_gene_id.equals("null")) ensambl_gene_id = "'"+
                          ensambl_gene_id+"'";
77                    splicing_cod = st.nextToken();
78                    if(!splicing_cod.equals("null")) splicing_cod = "'"+
                          splicing_cod+"'";
79                    splicing_name = st.nextToken();
80                    if(!splicing_name.equals("null")) splicing_name = "'"+
                          splicing_name+"'";
81
82                    //se sono nulli non li inserisco per nulla
83                    if(!splicing_cod.equals("null") && !splicing_name.equals("
                          null") ){
84                      sql_splicing = "INSERT INTO SPLICING VALUES("+
                            splicing_cod+","+splicing_name+") ON DUPLICATE KEY
                            UPDATE EVENT_TYPE_COD = EVENT_TYPE_COD;";
85                      System.out.println(sql_splicing);
86                      out.println(sql_splicing);
87
88                      sql_has = "INSERT INTO HAS VALUES("+splicing_cod+","+
                            ensambl_gene_id+") ON DUPLICATE KEY UPDATE
                            EVENT_TYPE_COD = EVENT_TYPE_COD;";
89                      System.out.println(sql_has);
90                      out.println(sql_has);
91                    }
92                  }
93                  i++;
94                }
95              lineNumber++;
96                reader.close();
97            }
```

```
        }
        catch(Exception e)
        {
                System.out.println("Exception while reading csv file: " + e);
        }
  }

}
```

C.6 SnpRecover.java

```
import java.io.BufferedReader;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.net.URL;
import java.net.URLEncoder;
import java.util.StringTokenizer;


public class SnpRecover {
  private final int num_lines = 481;
  private String[] chr = new String[num_lines];
  private String[] bp_start = new String[num_lines];
  private String[] bp_end = new String[num_lines];

  private String refsnp_id, allele, start, strand, clinical, phenotype,
      validated, allele_freq, sql;

  public SnpRecover()throws Exception{

    try
        {
      //tsv file containing data
            String strFile = "/home/uc_coordinates_hg19_reference_file.csv";
            //per output sql
            FileOutputStream sql_file = new FileOutputStream("/home/
                snp_insert.sql");
            PrintStream out = new PrintStream(sql_file);

            //create BufferedReader to read tsv file
            BufferedReader br = new BufferedReader( new FileReader(strFile
                ));
            String strLine = "";
            StringTokenizer st = null;
            int lineNumber = 0;
            br.readLine(); //la prima riga contenente meta-info viene
                scartata cosÃň
            //read comma separated file line by line
            while( (strLine = br.readLine()) != null )
            {
```

```
39                  //break comma separated line using ","
40                  st = new StringTokenizer(strLine, ",");
41                  String uc = st.nextToken();
42                  chr[lineNumber] = st.nextToken();
43                  chr[lineNumber] = chr[lineNumber].replace("\"", "");//
                        togliamo fastidiosi apici
44                  bp_start[lineNumber] = st.nextToken();
45                  bp_end[lineNumber] = st.nextToken();
46                  System.out.println("Sto recuperando "+uc+": "+chr[lineNumber
                        ]+" "+bp_start[lineNumber]+" "+bp_end[lineNumber]);
47
48
49                  String myxml = "<Query virtualSchemaName = \"default\"
                        formatter = \"CSV\" header = \"1\" "
50                  +"uniqueRows = \"0\" count = \"\" datasetConfigVersion =
                        \"0.6\" >"
51                  +"<Dataset name = \"hsapiens_snp\" interface = \"default\" >"
52                  +"<Filter name = \"chr_name\" value = \""+chr[lineNumber]+"
                        \"/>"
53                  +"<Filter name = \"chrom_start\" value = \""+bp_start[
                        lineNumber]+"\"/>"
54                  +"<Filter name = \"chrom_end\" value = \""+bp_end[lineNumber
                        ]+"\"/>"
55                  +"<Attribute name = \"refsnp_id\" />"
56                  +"<Attribute name = \"allele\" />"
57                  +"<Attribute name = \"chrom_start\" />"
58                  +"<Attribute name = \"chrom_strand\" />"
59                  +"<Attribute name = \"clinical_significance\" />"
60                  +"<Attribute name = \"phenotype_description\" />"
61                  +"<Attribute name = \"validated\" />"
62                  +"<Attribute name = \"minor_allele_freq\" />"
63                  +"</Dataset>"
64                  +"</Query>";
65                  //System.out.println(myxml);
66                  String encoded = URLEncoder.encode(myxml, "utf-8");
67                  //System.out.println(encoded.length());
68                  URL url = new URL("http://www.biomart.org/biomart/
                        martservice?query="+encoded);
69                  InputStream response = url.openStream();
70                  BufferedReader reader = new BufferedReader(new
                        InputStreamReader(response));
71                  int i=0;
72                  for (String line; (line = reader.readLine()) != null;) {
73                    //se ÃÍ la prima riga -> scarto
74                    if(i!=0) {
75                      int first = line.indexOf("\"");
76                      int last = line.lastIndexOf("\"");
77                      if(first != -1){
78                        String substring = line.substring(first,last);
79                        String substring_replaced = substring.replace(",", "-")
                              ;
80                        line = line.replace(substring, substring_replaced);
81                        line = line.replace("\"", "");
82                      }
```

```
                    while(line != (line = line.replace(",,",",null,")));
                    if(line.charAt(0) == ',') line = "null"+ line;
                    if(line.charAt(line.length()-1) == ',') line = line + "null";
                    st = new StringTokenizer(line, ",");
                    System.out.println(line);

                    refsnp_id = st.nextToken();
                    if(!refsnp_id.equals("null")) refsnp_id = "'"+refsnp_id+"'";
                    allele = st.nextToken();
                    if(!allele.equals("null")) allele = "'" + allele +"'";
                    start = st.nextToken();
                    if(!start.equals("null")) start = "'" +start + "'";
                    strand = st.nextToken();
                    clinical = st.nextToken();
                    if(!clinical.equals("null")) clinical = "'"+clinical+"'";
                    phenotype = st.nextToken();
                    if(!phenotype.equals("null")) phenotype = "'"+phenotype+"'";
                    validated = st.nextToken();
                    if(!validated.equals("null")) validated = "'"+validated+"'";
                    allele_freq = st.nextToken();

                    sql = "INSERT IGNORE INTO SNP VALUES("+ refsnp_id +","+start+","+allele+","+validated+
                    ","+allele_freq+","+phenotype+","+clinical+","+strand+","+"'"+chr[lineNumber]+"'"+
                    ","+bp_start[lineNumber]+");";
                    System.out.println(sql);
                    System.out.println();
                    out.println(sql);

                  }
                  i++;
                }
                reader.close();
                lineNumber++;
            }
        }
        catch(Exception e)
        {
                System.out.println("Exception while reading csv file: " + e);
        }
  }

}
```

C.7 PathologyRecover.java

```
import java.io.BufferedReader;
import java.io.FileOutputStream;
```

```
3  import java.io.FileReader;
4  import java.io.InputStream;
5  import java.io.InputStreamReader;
6  import java.io.PrintStream;
7  import java.net.URL;
8  import java.net.URLEncoder;
9  import java.util.StringTokenizer;
10
11 public class PathologyRecover {
12   private final int num_lines = 481;
13   private String[] chr = new String[num_lines];
14   private String[] bp_start = new String[num_lines];
15   private String[] bp_end = new String[num_lines];
16
17   public PathologyRecover()throws Exception{
18     //leggo dal file le info
19       //tsv file containing data
20           String strFile = "/home/uc_coordinates_hg19_reference_file.csv";
21
22           //reader per controllare id patologia
23           BufferedReader read = new BufferedReader( new FileReader("/
                home/MIM+PATHOLOGY_PULITO_ORDINATO+cri.csv"));
24           //file sql
25           FileOutputStream sql_file = new FileOutputStream("/home/
                pat_related2_insert.sql");
26           PrintStream out = new PrintStream(sql_file);
27           //create BufferedReader to read tsv file
28           BufferedReader br = new BufferedReader( new FileReader(strFile
                ));
29           String strLine = "";
30           StringTokenizer st = null;
31           int lineNumber = 0;
32           br.readLine(); //la prima riga contenente meta-info viene
                scartata cosÃň
33           //read comma separated file line by line
34           while( (strLine = br.readLine()) != null )
35           {
36               //break comma separated line using ","
37               st = new StringTokenizer(strLine, ",");
38               String uc = st.nextToken();
39               chr[lineNumber] = st.nextToken();
40               chr[lineNumber] = chr[lineNumber].replace("\"", "");//
                    togliamo fastidiosi apici
41               bp_start[lineNumber] = st.nextToken();
42               bp_end[lineNumber] = st.nextToken();
43               System.out.println("Sto recuperando "+uc+": "+chr[lineNumber
                    ]+" "+bp_start[lineNumber]+" "+bp_end[lineNumber]);
44
45
46               String myxml = "<Query virtualSchemaName = \"default\"
                    formatter = \"TSV\" header = \"1\" "
47               +"uniqueRows = \"0\" count = \"\" datasetConfigVersion =
                    \"0.6\" >"
```

```
48                  +"<Dataset name = \"hsapiens_gene_ensembl\" interface = \"
                        default\" >"
49                  +"<Filter name = \"chromosome_name\" value = \""+chr[
                        lineNumber]+"\"/>"
50                  +"<Filter name = \"start\" value = \""+bp_start[lineNumber]+"
                        \"/>"
51                  +"<Filter name = \"end\" value = \""+bp_end[lineNumber]+"\"/>
                        "
52                  +"<Attribute name = \"ensembl_gene_id\" />"
53                  +"<Attribute name = \"mim_morbid_description\" />"
54                  +"<Attribute name = \"pathology\" />"
55                  +"</Dataset>"
56                  +"</Query>";
57                  //System.out.println(myxml);
58                  String encoded = URLEncoder.encode(myxml, "utf-8");
59                  //System.out.println(encoded.length());
60                  URL url = new URL("http://www.biomart.org/biomart/
                        martservice?query="+encoded);
61                  InputStream response = url.openStream();
62                  BufferedReader reader = new BufferedReader(new
                        InputStreamReader(response));
63
64                  String line = reader.readLine(); //scarto indice colonne
65
66                  while ( (line = reader.readLine()) != null) { //per ogni
                        riga che mi restituiscono
67                    st = new StringTokenizer(line, " ");
68                    String gene_id = st.nextToken();
69                    String pat="", mim="";
70                    if(st.hasMoreTokens())
71                      mim = st.nextToken();
72                    if(st.hasMoreTokens())
73                      pat = st.nextToken();
74
75                    //qui scopri id patologia o iventalo
76
77                    if(mim.contains(";")){
78                      st = new StringTokenizer(mim, ";");
79                      mim = st.nextToken();
80                    }
81                    if(pat.contains(";")){
82                      st = new StringTokenizer(pat, ";");
83                      pat = st.nextToken();
84                    }
85
86                    if(!mim.equals("") && mim.startsWith(" "))
87                      mim = mim.substring(1); //il famoso spazio iniziale
88                    mim = mim.toLowerCase();
89                    if(!pat.equals("") && pat.startsWith(" "))
90                      pat = pat.substring(1); //il famoso spazio iniziale
91                    pat = pat.toLowerCase();
92                    System.out.println("cerco... :"+pat+ " "+mim);
93                    String id_pat="", id_mim="";
94
```

```
95                  while ( (line = read.readLine()) != null){
96                    if(!pat.equals("")){
97                      if (line.toLowerCase().startsWith(pat)){
98                        st = new StringTokenizer(line, ";");
99                        st.nextToken();
100                       id_pat = st.nextToken();
101                       System.out.println("trovato "+pat+" id:"+id_pat);
102                       break;
103                     }
104                   }
105                   else
106                     break;
107                 }
108                 //resetto read
109                 //riporto all'inizio file
110                 read =new BufferedReader( new FileReader("/home/MIM+
                        PATHOLOGY_PULITO_ORDINATO+cri.csv"));
111
112                 while ( (line = read.readLine()) != null){
113                   if(!mim.equals("")){
114                     if (line.toLowerCase().startsWith(mim)){
115                       st = new StringTokenizer(line, ";");
116                       st.nextToken();
117                       id_mim = st.nextToken();
118                       System.out.println("trovato "+mim+" id:"+id_mim);
119                       break;
120                     }
121                   }
122                   else
123                     break;
124                 }
125                 //resetto read
126                 //riporto all'inizio file
127                 read =new BufferedReader( new FileReader("/home/MIM+
                        PATHOLOGY_PULITO_ORDINATO+cri.csv"));
128
129                 if(mim.contains("'")){
130                   mim = mim.replace("'", "\"");
131                 }
132                 if(pat.contains("'")){
133                   pat = pat.replace("'", "\"");
134                 }
135
136                 if(!pat.equals("") && !id_pat.startsWith("DOID")){
137                   String sql1 = "INSERT IGNORE INTO PATHOLOGY VALUES('"+
                          id_pat+"','"+"'"+pat+
138                 "',null,null,null,null,null,-1);";
139                   out.println(sql1);
140                   System.out.println(sql1);
141                 }
142                 if(!mim.equals("") && !id_mim.startsWith("DOID")){
143                   String sql1 = "INSERT IGNORE INTO PATHOLOGY VALUES('"+
                          id_mim+"','"+"'"+mim+
144                 "',null,null,null,null,null,-1);";
```

```
                    out.println(sql1);
                    System.out.println(sql1);
                  }
                  //inserimento finale
                  if(!pat.equals("")){
                    String sql = "INSERT IGNORE INTO RELATED_TO VALUES('"+id_pat+"','"+gene_id+"');";
                    out.println(sql);
                    System.out.println(sql);

                    if(id_pat.equals("DOID:1891") || id_pat.equals("DOID:4645") || id_pat.equals("DOID:771") || id_pat.equals("DOID:768") ){
                      String sql_ = "INSERT IGNORE INTO RELATED_TO VALUES('"+id_pat+"bis','"+gene_id+"');";
                      out.println(sql_);
                      break;

                    }
                  }
                  if(!mim.equals("")){
                      String sql = "INSERT IGNORE INTO RELATED_TO VALUES('"+id_mim+"','"+gene_id+"');";
                      out.println(sql);
                      System.out.println(sql);

                    if(id_mim.equals("DOID:1891") || id_mim.equals("DOID:4645") || id_mim.equals("DOID:771") || id_mim.equals("DOID:768") ){
                      String sql_ = "INSERT IGNORE INTO RELATED_TO VALUES('"+id_mim+"bis','"+gene_id+"');";
                      out.println(sql_);
                    }
                    }
                }
                lineNumber++;
                reader.close();
            }
        }

}
```

C.8 TestCorrispondenze.java

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.PrintStream;
import java.util.ArrayList;


public class TestCorrispondenze {
```

```
10
11 //TESTA LE CORRISPONDENZE DIRETTE TRA LE PATOLOGIE BIOMART E QUELLE DELLA
       ONTOLOGIA
12   public static void main(String[] args) throws Exception {
13
14     extractSourceFileFromDO();
15     String strFile = "/home/kira/Scrivania/AllpathologyfromDO.txt";
16     BufferedReader br = new BufferedReader( new FileReader(strFile));
17     String strFile2 = "/home/kira/Scrivania/pat_mim.txt";
18     BufferedReader br2 = new BufferedReader( new FileReader(strFile2));
19     FileOutputStream file = new FileOutputStream("/home/result.txt");
20         PrintStream out = new PrintStream(file);
21     FileOutputStream file2 = new FileOutputStream("/home/suggests.txt");
22         PrintStream out_sugg = new PrintStream(file2);
23
24         ArrayList<String> do_name = new ArrayList<String>();
25         ArrayList<String> patmim_name = new ArrayList<String>();
26         String line = "";
27
28         //riempio le patologie dall'ontologia
29         while( (line = br.readLine()) != null ){
30             do_name.add(line.toLowerCase());
31         }
32         //riempio le patologie di biomart
33         while( (line = br2.readLine()) != null ){
34             patmim_name.add(line.toLowerCase());
35         }
36
37         for(int i=0; i< patmim_name.size(); i++){
38           if(do_name.contains(patmim_name.get(i))){
39             System.out.println(patmim_name.get(i));
40             out.println(patmim_name.get(i));
41           }
42           else{
43             int ris;
44         for(int k=0; k<do_name.size(); k++){
45               ris = computeLevenshteinDistance(do_name.get(k), patmim_name.
                     get(i));
46           if(ris < patmim_name.get(i).length()/2)//se le differenze non sono
                  piÃź della metÃă dei caratteri
47             out_sugg.println(patmim_name.get(i) + " somiglia a: " + do_name.
                   get(k) );
48         }
49           }
50         }
51
52   }
53
54   private static void extractSourceFileFromDO() throws Exception{
55     String strFile = "/home/kira/Scrivania/HumanDO.txt";
56     BufferedReader br = new BufferedReader( new FileReader(strFile));
57     FileOutputStream file = new FileOutputStream("/home/AllpathologyfromDO
           .txt");
58         PrintStream out = new PrintStream(file);
```

```
    ArrayList<String> do_name = new ArrayList<String>();
    String line = "";
    while( (line = br.readLine()) != null ){
      if(line.startsWith("name:") || line.startsWith("synonym:")){
        if(line.startsWith("name:"))
          line = line.substring(6);
        else{
          line = line.substring(9);
          int firstI, secondI;
          firstI=line.indexOf("\"");
          secondI=line.lastIndexOf("\"");
          line = line.substring(firstI+1, secondI);
        }
        if(!do_name.contains(line)) do_name.add(line);
      }
      else if(line.startsWith("[Typedef]"))
        break;
    }

    //stampa sul file i nomi puliti senza doppioni
    for(int i=0; i< do_name.size(); i++)
        out.println(do_name.get(i));

  }

  private static int minimum(int a, int b, int c) {
        return Math.min(Math.min(a, b), c);
  }

  public static int computeLevenshteinDistance(CharSequence str1,
              CharSequence str2) {
        int[][] distance = new int[str1.length() + 1][str2.length() + 1];

        for (int i = 0; i <= str1.length(); i++)
              distance[i][0] = i;
        for (int j = 1; j <= str2.length(); j++)
              distance[0][j] = j;

        for (int i = 1; i <= str1.length(); i++)
              for (int j = 1; j <= str2.length(); j++)
                    distance[i][j] = minimum(
                                  distance[i - 1][j] + 1,
                                  distance[i][j - 1] + 1,
                                  distance[i - 1][j - 1]
                                                + ((str1.charAt(i - 1) ==
                                                      str2.charAt(j - 1))
                                                      ? 0
                                                            : 1));

        return distance[str1.length()][str2.length()];
  }

}
```

C.9 OboEditAllFilter.java

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.PrintStream;
import java.util.StringTokenizer;


public class OboEditAllFilter {
  //classe per creare il file dei filtri per obo-edit per ottenere tutti i nodi
  //dell'albero che ci interessa, anche quelli intermedi

  public static void main(String[] args) throws Exception {
    String srcFile = "/home/MIM+PATHOLOGY_PULITO_ORDINATO+cri.csv";
    BufferedReader br = new BufferedReader( new FileReader(srcFile));
    FileOutputStream out_file = new FileOutputStream("/home/OBOAllfilter.xml");
    PrintStream out = new PrintStream(out_file);

    //parte iniziale
    out.println("<?xml version=\"1.0\" encoding=\"UTF-8\"?>" +
        "<java version=\"1.6.0_18\" class=\"java.beans.XMLDecoder\">" +
        "<object class=\"org.obo.filters.CompoundFilterImpl\">" +
        "<void property=\"booleanOperation\">" +
        "<int>1</int>" +
        "</void>" +
        "<void property=\"filters\">");

        StringTokenizer st = null;
        String strLine = "";
        while( (strLine = br.readLine()) != null ){
          st = new StringTokenizer(strLine, ";");
          String pat_name = st.nextToken();
          String id = st.nextToken();
          if(id.startsWith("DOID")){
            out.println("<void method=\"add\">" +
                "<object class=\"org.obo.filters.ObjectFilterImpl\">" +
                "<void property=\"comparison\">" +
                "<object id=\"EqualsComparison0\" class=\"org.obo.filters.EqualsComparison\"/>" +
                "</void>" +
                "<void property=\"criterion\">" +
                "<object id=\"IDSearchCriterion0\" class=\"org.obo.filters.IDSearchCriterion\"/>" +
                "</void>" +
                  "<void property=\"reasoner\">" +
                  "<object id=\"RuleBasedReasoner0\" class=\"org.obo.reasoner.rbr.RuleBasedReasoner\">"+
                  "<void property=\"properties\">" +
                  "<object class=\"java.util.HashSet\"/>" +
```

```
                "</void>" +
                "</object>"+
                "</void>"+
              "<void property=\"value\">" +
              "<string>"+id+"</string>" +
              "</void>" +
              "</object>" +
              "</void>" +
              //parte per richiedere padri raggiungibili con is_a
              "<void method=\"add\">"+
              "<object class=\"org.obo.filters.ObjectFilterImpl\">"+
              "<void property=\"aspect\"> " +
              "<object id=\"DescendantSearchAspect0\" class=\"org.obo.filters.DescendantSearchAspect\"/>"+
              "</void>"+
              "<void property=\"comparison\">"+
              "<object id=\"EqualsComparison0\" class=\"org.obo.filters.EqualsComparison\"/>"+
              "</void>"+
              "<void property=\"criterion\">"+
              "<object id=\"IDSearchCriterion0\" class=\"org.obo.filters.IDSearchCriterion\"/>"+
              "</void>"+
              "<void property=\"reasoner\">"+
              "<object id=\"RuleBasedReasoner0\" class=\"org.obo.reasoner.rbr.RuleBasedReasoner\">"+
              "<void property=\"properties\">"+
              "<object class=\"java.util.HashSet\"/>"+
              "</void>"+
              "</object>"+
              "</void>"+
              "<void property=\"traversalFilter\">"+
              "<object class=\"org.obo.filters.LinkFilterImpl\">"+
              "<void property=\"filter\">"+
              "<void property=\"comparison\">"+
              "<object class=\"org.obo.filters.EqualsComparison\"/>"+
              "</void>"+
              "<void property=\"criterion\">"+
              "<object class=\"org.obo.filters.IDSearchCriterion\"/>"+
              "</void>"+
              "<void property=\"value\">"+
              "<string>OBO_REL:is_a</string>"+
              "</void>"+
              "</void>"+
              "</object>"+
              "</void>"+
              "<void property=\"value\">"+
              "<string>" + id +"</string>"+
              "</void>"+
              "</object>"+
              "</void>");

          }else if(id.startsWith("SI")){
            out.println("<void method=\"add\">" +
```

```
                "<object class=\"org.obo.filters.ObjectFilterImpl\">" +
                "<void property=\"comparison\">" +
                "<object id=\"EqualsComparison0\" class=\"org.obo.filters.EqualsComparison\"/>" +
                "</void>" +
                "<void property=\"criterion\">" +
                "<object class=\"org.obo.filters.NameSynonymSearchCriterion\"/>" +
                "</void>" +
                  "<void property=\"reasoner\">" +
                  "<object id=\"RuleBasedReasoner0\" class=\"org.obo.reasoner.rbr.RuleBasedReasoner\">"+
                  "<void property=\"properties\">" +
                  "<object class=\"java.util.HashSet\"/>" +
                  "</void>" +
                  "</object>"+
                  "</void>"+
                "<void property=\"value\">" +
                "<string>"+pat_name+"</string>" +
                "</void>" +
                "</object>" +
                "</void>" +
                //parte per richiedere padri raggiungibili con is_a
                "<void method=\"add\">"+
                "<object class=\"org.obo.filters.ObjectFilterImpl\">"+
                "<void property=\"aspect\"> " +
                "<object id=\"DescendantSearchAspect0\" class=\"org.obo.filters.DescendantSearchAspect\"/>"+
                "</void>"+
                "<void property=\"comparison\">"+
                "<object id=\"EqualsComparison0\" class=\"org.obo.filters.EqualsComparison\"/>"+
                "</void>"+
                "<void property=\"criterion\">"+
                "<object id=\"NameSynonymSearchCriterion0\" class=\"org.obo.filters.NameSynonymSearchCriterion\"/>"+
                "</void>"+
                "<void property=\"reasoner\">"+
                "<object id=\"RuleBasedReasoner0\" class=\"org.obo.reasoner.rbr.RuleBasedReasoner\">"+
                "<void property=\"properties\">"+
                "<object class=\"java.util.HashSet\"/>"+
                "</void>"+
                "</object>"+
                "</void>"+
                "<void property=\"traversalFilter\">"+
                "<object class=\"org.obo.filters.LinkFilterImpl\">"+
                "<void property=\"filter\">"+
                "<void property=\"comparison\">"+
                "<object class=\"org.obo.filters.EqualsComparison\"/>"+
                "</void>"+
                "<void property=\"criterion\">"+
                "<object class=\"org.obo.filters.IDSearchCriterion\"/>"+
                "</void>"+
```

```
                "<void property=\"value\">"+
                "<string>OBO_REL:is_a</string>"+
                "</void>"+
                "</void>"+
                "</object>"+
                "</void>"+
                "<void property=\"value\">"+
                "<string>" + pat_name +"</string>"+
                "</void>"+
                "</object>"+
                "</void>");
            }
          }

        //fine
        out.println("</void></object></java>");
    }

}
```

C.10 PathologyObj.java

```
package albero;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintStream;
import java.util.ArrayList;
import java.util.StringTokenizer;

public class PathologyObj {
  public String id="";
  public String name="";
  public String def="";
  public String subset="";
  public String comment="";
  public ArrayList<String> synonym = new ArrayList<String>();
  public ArrayList<String> xref = new ArrayList<String>();
  public String is_a="";
  public int pre = 0;
  public int post = 0;
  public int level = 0;

  public PathologyObj(String id) throws IOException{
      BufferedReader br = new BufferedReader( new FileReader("/home/
          HumanDO.txt"));
      String strLine;
      while( (strLine = br.readLine()) != null ){
        //System.out.println(strLine);
        if(strLine.startsWith("id: "+id)){
```

```
31            //System.out.println(strLine);
32            this.id = strLine;
33            while( !(strLine = br.readLine()).startsWith("[Term]")){
34            if(strLine.startsWith("name:")) this.name = strLine;
35            else if(strLine.startsWith("def:")) this.def = strLine;
36            else if(strLine.startsWith("subset:")) this.subset = strLine;
37            else if(strLine.startsWith("synonym:")) this.synonym.add(strLine);
38            else if(strLine.startsWith("xref:")) this.xref.add(strLine);
39            else if(strLine.startsWith("is_a:")) {
40              StringTokenizer st;
41              st = new StringTokenizer(strLine, "!");
42              String tmp = st.nextToken();
43              is_a = tmp.substring(0,tmp.length()-1);
44            }
45            }
46            break;
47          }
48        }
49    }
50
51    public void tostring(){
52      System.out.println(id);
53      System.out.println(name);
54      System.out.println(def);
55      System.out.println(subset);
56      System.out.println(comment);
57      for(int i=0; i<synonym.size();i++)
58        System.out.println(synonym.get(i));
59      for(int i=0; i<xref.size();i++)
60        System.out.println(xref.get(i));
61      System.out.println(is_a);
62    }
63
64    public String toString(){
65      return id+" "+level;
66    }
67
68  }
```

C.11 NTree.java

```
1  package albero;
2
3  import java.util.ArrayList;
4
5  public class NTree //classe pubblica albero
6  {
7          protected NTreeNode root; //radice dell'albero
8          public ArrayList<PathologyObj> array = new ArrayList<PathologyObj
               >(); //per dopo.. lista
9          public NTree() //costruttore
10         {
11                 root=null;
```

```
12          }
13
14          public void setRoot(NTreeNode root){
15            this.root = root;
16          }
17
18          public boolean insert(String parent_id, NTreeNode son) //metodo per
                l'inserimento
19          {
20                 if(this.search(root,parent_id)&&son!=null)
21                 {
22                     //System.out.println("trovato!");
23                         NTreeNode temp=this.searchNode(root,parent_id);
24                         NTreeNode temp2 = temp.firstSon;
25                         temp.firstSon=son;
26                         son.brother = temp2;
27                         //settiamo level prima di inserire
28                         son.element.level = temp.element.level+1;
29                         return true;
30                 }
31                 else
32                         System.out.println("Errore, nodo inesistente.
                               Impossibile inserire i nodi dei figli");
33                 //stampa errore
34                 return false;
35          }
36          public boolean search(NTreeNode StartPoint, String node_id) //
                metodo per la ricerca di un nodo
37          {
38              NTreeNode p = StartPoint;
39                 if(p!=null){
40                   boolean ris;
41                     //System.out.println(p.element.id + " confronto "+
                           node_id);
42                         if(p.element.id.equals(node_id)) return true;
43                         else
44                         {
45                                 NTreeNode t=p.firstSon;
46                                 while(t!=null)
47                                 {
48                                         ris = search(t, node_id);
49                                         if(ris) return true;
50                                         t=t.brother;
51                                 }
52                         }
53                 }
54                 return false;
55          }
56
57          public NTreeNode searchNode(NTreeNode StartPoint, String node_id) //
                metodo per la ricerca di un nodo
58          {
59              NTreeNode p = StartPoint;
60              NTreeNode ris;
```

```
            if(p!=null){
                if(p.element.id.equals(node_id)) return p;
                else
                {
                    NTreeNode t=p.firstSon;
                    while(t!=null)
                    {
                        ris = searchNode(t, node_id);
                        if(ris != null) return ris;
                        t=t.brother;
                    }
                }
            }
            return null;
        }

        public void preorder(){
            preorder(root);
        }
        public void preorder(NTreeNode p){
            if(p!=null){
                System.out.println(p.toString());
                NTreeNode t=p.firstSon;
                while(t!=null){
                    preorder(t);
                    t=t.brother;
                }
            }
        }

        public void set_preorder(){
            set_preorder(root,1);
        }
        public int set_preorder(NTreeNode p, int num){
            if(p!=null){
                p.element.pre = num;
                System.out.println(p.toString()+" "+ p.element.pre);
                NTreeNode t=p.firstSon;
                while(t!=null){
                    num = set_preorder(t, ++num);
                    t=t.brother;
                }
            }
            return num;
        }

        public void postorder(){
            postorder(root);
        }
        public void postorder(NTreeNode p){
            if(p!=null){
                NTreeNode t=p.firstSon;
                while(t!=null){
                    postorder(t);
```

```
                            t=t.brother;
                        }
                System.out.println(p.toString());
                }
        }

        public void set_postorder(){
            set_postorder(root,1);
        }
        public int set_postorder(NTreeNode p, int num){
            if(p!=null){
                    NTreeNode t=p.firstSon;
                    while(t!=null){
                            num = set_postorder(t, num);
                            num++;
                            t=t.brother;
                    }
            p.element.post = num;
            System.out.println(p.toString()+" "+ p.element.post);
            }
      return num;
      }

        public void toArraylist(){
          toArraylist(root);
        }
        public void toArraylist(NTreeNode p){
            if(p!=null){
                array.add(p.element);
                NTreeNode t=p.firstSon;
                while(t!=null){
                        toArraylist(t);
                        t=t.brother;
                }
        }
        }
}
```

C.12 NTreeNode.java

```
package albero;
public class NTreeNode //implementazione nodo albero con tipo generico E
{
        protected NTreeNode firstSon; //primo figlio
        protected NTreeNode brother; //fratello
        protected PathologyObj element; //elemento del nodo
        public NTreeNode() //costruttore
        {
                this(null,null, null);
        }
        public NTreeNode(PathologyObj element, NTreeNode firstSon)
        {
                this(element, firstSon,null);
```

```
        }
        public NTreeNode(PathologyObj element, NTreeNode firstSon, NTreeNode
              brother)
        {
                this.element=element;
                this.firstSon=firstSon;
                this.brother=brother;
        }
        public String toString(){
                return element.toString();
        }
}
```

C.13 TreeMaker.java

```
package albero;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.StringTokenizer;

public class TreeMaker {
  NTree tree;

  public TreeMaker() throws IOException {
    BufferedReader br = new BufferedReader( new FileReader("/home/
        allnodesfromobo.txt"));
    ArrayList<PathologyObj> list = new ArrayList<PathologyObj>();
    StringTokenizer st = null;
    String strLine,id;

    //while per la creazione della lista di oggetti che ci interessano
    while((strLine = br.readLine()) != null ){
      st = new StringTokenizer(strLine, " ");
      id = st.nextToken();
      //System.out.println(id);
      list.add(new PathologyObj(id));
    }

    //creiamo l'albero e colleghiamo i nodi
    tree = new NTree();
    for(int i=0; i<list.size(); i++)
      if(list.get(i).id.equals("id: DOID:4")) {
        tree.setRoot(new NTreeNode(list.get(i),null,null));
        list.remove(i);
      }

    while(!list.isEmpty()){
      for(int i=0; i<list.size(); i++){
```

```
37          if(i == 0) { System.out.println("Stampo albero:"); tree.preorder();
                System.out.println();}
38          //attenzione agli obsoleti! sono 3!
39          if(list.get(i).is_a.equals("")){ System.out.println("obsolete: "+
                list.get(i).id); list.remove(i); }
40          String parent_id = "id: "+ list.get(i).is_a.substring(6);
41          System.out.println("io sono: "+ list.get(i).id + " cerco: " +
                parent_id);
42          if(tree.insert(parent_id, new NTreeNode(list.get(i),null,null))) {
43            list.remove(i);
44            System.out.println("nodo esistente!!!");
45          }
46
47        }
48      }
49
50      //collego a mano gli ultimi con due is_a
51
52      //optical nerve disease
53      PathologyObj tmp = new PathologyObj("DOID:1891");
54      tmp.id = tmp.id+"bis";
55      tmp.is_a = "id: DOID:1393";
56      tree.insert(tmp.is_a, new NTreeNode(tmp,null,null));
57
58      //retinal cancer
59      tmp = new PathologyObj("DOID:4645");
60      tmp.id = tmp.id+"bis";
61      tmp.is_a = "id: DOID:2174";
62      tree.insert(tmp.is_a, new NTreeNode(tmp,null,null));
63
64      //retinal cell cancer
65      tmp = new PathologyObj("DOID:771");
66      tmp.id = tmp.id+"bis";
67      tmp.is_a = "id: DOID:4645bis";
68      tree.insert(tmp.is_a, new NTreeNode(tmp,null,null));
69
70      //retinoblastoma
71      tmp = new PathologyObj("DOID:768");
72      tmp.id = tmp.id+"bis";
73      tmp.is_a = "id: DOID:771bis";
74      tree.insert(tmp.is_a, new NTreeNode(tmp,null,null));
75
76      tree.set_preorder();
77      System.out.println();
78      tree.set_postorder();
79    }
80
81    public ArrayList<PathologyObj> array(){
82      tree.toArraylist();
83      return tree.array;
84    }
85
86  }
```

C.14 RiempiPatTable.java

```
package albero;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.util.ArrayList;

public class RiempiPatTable {
  //classe per creare script per riempire tabelle: PATHOLOGY, SYNONYM, XREF
  //con quelle che conosco dalla ontologia
  public static void main(String[] args) throws IOException {

    FileOutputStream sql_file = new FileOutputStream("/home/Path-syn-xref.sql");
       PrintStream out = new PrintStream(sql_file);

    TreeMaker op = new TreeMaker();
    ArrayList<PathologyObj> list = op.array();

    //riempio quelle nell'array

    //aggiungo obsoleti
    PathologyObj tmp = new PathologyObj("DOID:8600");
    tmp.level= -1;
    System.out.println("try:"+tmp.level);
    list.add(tmp);

    tmp = new PathologyObj("DOID:4625");
    tmp.level= -1;
    list.add(tmp);

    tmp = new PathologyObj("DOID:410");
    tmp.level= -1;
    list.add(tmp);
    System.out.println("try:"+tmp.level);

    for(int i=0; i<list.size(); i++){
      //System.out.println(list.size());
      tmp = list.get(i);
      System.out.println("try:"+tmp.level);
      String def=",null";
      if(!tmp.def.equals("")) def = ",'"+tmp.def.substring(5).replace("'", "\"")+"'";
      String subset=",null";
      if(!tmp.subset.equals("")) subset = ",'"+tmp.subset.substring(8)+"'";
      String comment=",null";
      if(!tmp.comment.equals("")) comment = ",'"+tmp.comment.substring(9)+"'";
      String pre=",null",post=",null";
      if(tmp.pre != 0)
        pre = ","+Integer.toString(tmp.pre);
```

```
      if(tmp.post != 0)
        post = ","+Integer.toString(tmp.post);

      String sql = "INSERT INTO PATHOLOGY VALUES('"+tmp.id.substring(4)+"','"
          +"'"+tmp.name.substring(6).replace("'", "\"")+
      "'"+comment+def+subset+pre+post+","+tmp.level+");";
      out.println(sql);

      for(int j=0; j<tmp.synonym.size(); j++){
        sql = "INSERT INTO SYNONYM VALUES ('"+tmp.id.substring(4)+"','"+tmp.
            synonym.get(j).substring(9).replace("'", "\"")+"');";
        out.println(sql);
      }

      for(int j=0; j<tmp.xref.size(); j++){
        sql = "INSERT INTO XREF VALUES ('"+tmp.id.substring(4)+"','"+tmp.
            xref.get(j).substring(6).replace("'", "\"")+"');";
        out.println(sql);
      }

    }

  }
}
```

C.15 Index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>UCbase</title>
  <link rel="icon" href="http://localhost/appswebsitetemplate/favicon.ico" /
      >
  <link rel="stylesheet" href="css/style.css" type="text/css">
</head>
<body>
  <div class="page">
    <div class="sidebar">
      <div id="logo">
        <a href="index.html"><img src="images/logo.png" alt="logo"></a>
      </div>
      <ul class="navigation">
        <li class="selected">
          <a href="index.html">Home</a>
        </li>
        <li>
          <a href="uc_data_mining.php">UC Data Mining</a>
        </li>
        <li>
          <a href="related_works.html">Related Works</a>
        </li>
        <li>
```

```
26          <a href="about_us.html">About Us</a>
27        </li>
28      </ul>
29      <p id="suggest"> Type an ID, a GENE NAME <br> or what you need and GO
            !</p>
30      <form action="">
31        <input type="text" value="Rapid Search" onblur="this.value=!this.
              value?'Rapid Search':this.value;" onfocus="this.select()"
              onclick="this.value='';">
32        <input type="submit" value="" onclick="alert('Not implemented yet
              !');">
33      </form>
34
35      <p id="mail">Comments, questions? <br>
36        Email <a href="mailto:vinxlemons@gmail.com">vinxlemons@gmail.com</a
              ></p>
37    </div>
38    <div class="content">
39      <div class="article">
40        <h2>Hi and welcome to the UCbase, visitor!</h2>
41        <p>UCBase is the nerve center for the study of ultraconservate
              sequences: collects the latest news, research and information
              about it, as well as offering new features and approaches to
              research in the field of organic. Specifically provides the
              ability to query an open database of the UC sequences of the
              human genome as you could see in the "UC Data Mining" section.<
              /p>
42      </div>
43      <div class="blog">
44        <div class="date">
45          <span>08-04</span>
46          <span>2013</span>
47        </div>
48        <div>
49          <h2>Things to Know</h2>
50          <div>
51            border
52          </div>
53        </div>
54        <ul>
55          <li>
56            <div>
57              <div>
58                <a href="#"><img src="images/dna.jpg" alt=""></a>
59                <a onclick="alert('Not present!');">.pdf</a>
60              </div>
61              <div>
62                <h3>Why UCbase? </h3>
63                <p>UCbase arises from the need to create a hub about
                      ultraconservate sequences and an innovative way to query
                       the data in the biologic context. The database is built
                       online recovering genetic information and building a
                      real hierarchy of diseases to make possible correlations
                      .</p>
```

```
              <a onclick="alert('Not implemented yet!');">Read more>></a>
            </div>
          </div>
        </li>
        <li>
          <div>
            <div>
              <a href="#"><img src="images/dna.jpg" alt=""></a>
              <a onclick="alert('Not present!');">.pdf</a>
            </div>
            <div>
              <h3>What are the UC sequences?</h3>
              <p>In biology, conserved sequences are similar or identical
                  sequences that occur within nucleic acid sequences (such
                   as RNA and DNA sequences), protein sequences, protein
                  structures or polymeric carbohydrates across species (
                  orthologous sequences) or within different molecules
                  produced by the same organism (paralogous sequences).</p
                  >
              <a onclick="alert('Not implemented yet!');">Read more>></a>
            </div>
          </div>
        </li>
        <li>
          <div>
            <div>
              <a href="#"><img src="images/dna.jpg" alt=""></a>
              <a onclick="alert('Not present!');">.pdf</a>
            </div>
            <div>
              <h3>Database and technical information</h3>
              <p>Cascading Style Sheets (CSS) were used for UCbase web
                  development to style web pages written in HTML and PHP.
                  The CSS specifications are maintained by the World Wide
                  Web Consortium (W3C).
                UCbase was created using MySQL database under Debian Etch
                     Linux OS installed on a Quad Core Processor machine
                     with 32 GB RAM.</p>
              <a onclick="alert('Not implemented yet!');">Read more>></a>
            </div>
          </div>
        </li>
      </ul>
      <div class="section">
        <a href="#"></a>
      </div>
    </div>
  </div>
 </div>
<img id="beta" src="images/Beta.png" style="position:absolute;top:160px;
    right:150px;">
</body>
</html>
```

C.16 Uc_data_mining.php

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>UCbase - UC Data Mining</title>
  <link rel="icon" href="http://localhost/appswebsitetemplate/favicon.ico" />
  <link rel="stylesheet" href="css/style.css" type="text/css">
</head>
<body>
  <div class="page">
    <div class="sidebar">
      <div id="logo">
        <a href="index.html"><img src="images/logo.png" alt="logo"></a>
      </div>
      <ul class="navigation">
        <li>
          <a href="index.html">Home</a>
        </li>
        <li class="selected">
          <a href="uc_data_mining.php">UC Data Mining</a>
        </li>
        <li>
          <a href="related_works.html">Related Works</a>
        </li>
        <li>
          <a href="about_us.html">About Us</a>
        </li>
      </ul>
      <p id="suggest"> Type an ID, a GENE NAME <br> or what you need and GO !</p>
      <form action="">
        <input type="text" value="Rapid Search" onblur="this.value=!this.value?'Rapid Search':this.value;" onfocus="this.select()" onclick="this.value='';">
        <input type="submit" value="" onclick="alert('Not yet implemented!');">
      </form>

      <p id="mail">Comments, questions? <br>
        Email <a href="mailto:vinxlemons@gmail.com">vinxlemons@gmail.com</a></p>
    </div>
    <div class="content">
      <div>
        <h2>UC Data Mining</h2>
        <p>In this section you can browse and query the database as you like.
You will be able to do semi-structured query or directly query the database through an SQL query.
```

```
This is one of the strengths of UCbase, pointing to a direct experience of "Data Mining" for the purposes of research.
</p><p>Below you will find the structure of the database expressed in the E/R standard language for databases and the query section.</p>
     <a href="images/Diagramma1.png" target="_blank"><img src="images/Diagramma1.png" width="650" height="350"></img></a>
     </div>
     <div class="data_mining">
       <div>
         <h2>Pre-formed Query</h2>
       </div>
       <ul>
         <span>All info about </span>
         <form action="result.php" method="GET">

           <select name="uc.id" >
           <?php
             $con = mysql_connect("localhost","client","is1fs");
             if (!$con)
             {
               die('Could not connect: ' . mysql_error());
             }

             mysql_select_db("UCbase", $con);
             $query = "SELECT UC_NAME FROM UC ";

             $result = mysql_query($query);

             while($row = mysql_fetch_array($result, MYSQL_NUM))
             {
               if($row[0] =="uc.0")
                 echo '<option value="'.$row[0].'" selected="selected">'.$row[0].'</option>';
               else
                 echo '<option value="'.$row[0].'" >'.$row[0].'</option>';

             }
           ?>
           </select>
           <input class ="submit"type="submit" value="">
         </form>
         <span> ?</span>
       </ul>
       <ul>
         <span>All info about </span>
         <form action="result_gene.php" method="GET">

           <select name="gene_name" >
           <?php

             $query = "SELECT WIKIGENE_NAME FROM GENE ";

             $result = mysql_query($query);

```

```
94            while($row = mysql_fetch_array($result, MYSQL_NUM))
95            {
96              if($row[0]!=""){
97                if($row[0] =="PEX14")
98                  echo '<option value="'.$row[0].'" selected="selected">'.
                        $row[0].'</option>';
99                else
100                 echo '<option value="'.$row[0].'" >'.$row[0].'</option>'
                        ;
101             }
102           }
103         ?>
104         </select>
105         <input class="submit" type="submit" value="">
106       </form>
107       <span>?</span>
108     </ul>
109     <ul>
110       <span id="third">All UC correled to</span>
111       <form action="result_search_pat.php">
112
113         <select id="tre" name="pat" >
114         <?php
115
116           $query = "SELECT NAME FROM PATHOLOGY ";
117
118           $result = mysql_query($query);
119
120           while($row = mysql_fetch_array($result, MYSQL_NUM))
121           {
122             if($row[0]!=""){
123               if($row[0] =="disease")
124                 echo '<option value="'.$row[0].'" selected="selected">'.
                        $row[0].'</option>';
125               else
126                 echo '<option value="'.$row[0].'" >'.$row[0].'</option>'
                        ;
127             }
128           }
129         ?>
130         </select>
131         <input class="submit" type="submit" value="">
132       </form>
133       <span style="width: 150px;">or its subtypes?</span>
134     </ul>
135     <ul>
136       <span id="third">All UC in </span>
137       <form action="result_search_uc.php" method="GET">
138
139         <select name="chr_id" >
140         <?php
141
142           $query = "SELECT DISTINCT CHR FROM UC ";
143
```

```
144             $result = mysql_query($query);
145
146             while($row = mysql_fetch_array($result, MYSQL_NUM))
147             {
148               if($row[0]!=""){
149                 if($row[0] =="1")
150                   echo '<option value="'.$row[0].'" selected="selected">'.
                        $row[0].'</option>';
151                 else
152                   echo '<option value="'.$row[0].'" >'.$row[0].'</option>'
                        ;
153               }
154             }
155           ?>
156           </select>
157           <span> starting between </span>
158
159           <input type="text" style="width: 70px;" value="num_bp" name="
                num_bp1" onblur="this.value=!this.value?'num_bp':this.value;
                " onfocus="this.select()" onclick="this.value='';">
160
161           <span> and</span>
162
163           <input type="text" style="width: 70px;" value="num_bp" name="
                num_bp2" onblur="this.value=!this.value?'num_bp':this.value;
                " onfocus="this.select()" onclick="this.value='';">
164           <input class="submit" type="submit" value="">
165         </form>
166         <span> ?</span>
167       </ul>
168       <ul>
169         <span>Blast sequence:</span>
170         <form action="blastn.php" method="GET">
171
172           <input type="text" title="insert a nucleotide sequence! ex:
                ACGTACAGTACG" style='width:360px;'value="ACGTACAGTACG" name=
                "sequence" onblur="this.value=!this.value?'ACGTACAGTACG':
                this.value;" onfocus="this.select()" onclick="this.value='';
                ">
173           <input class ="submit"type="submit" value="">
174         </form>
175       </ul>
176       <ul>
177         <span>Blast sequence:</span>
178         <form action="blastpat.php" method="GET">
179
180           <input type="text" title="insert a nucleotide sequence! ex:
                ACGTACAGTACG" value="ACGTACAGTACG" name="sequence" onblur="
                this.value=!this.value?'ACGTACAGTACG':this.value;" onfocus="
                this.select()" onclick="this.value='';">
181           <span> with pathology:</span>
182           <select name="pathology" >
183           <?php
184
```

```
            $query = "SELECT NAME FROM PATHOLOGY ";

            $result = mysql_query($query);

            while($row = mysql_fetch_array($result, MYSQL_NUM))
            {
              if($row[0]!=""){
                if($row[0] =="disease")
                  echo '<option value="'.$row[0].'" selected="selected">'.
                      $row[0].'</option>';
                else
                  echo '<option value="'.$row[0].'" >'.$row[0].'</option>'
                      ;
              }
            }
          ?>
          </select>
          <input class ="submit"type="submit" value="">
        </form>
      </ul>
      <div>
        <h2 id="second">Type your own Query!</h2>
        <ul>

          <form action="table.php">

            <input id="second"type="text" name="query" value="SELECT *
                FROM UC" onblur="this.value=!this.value?'SELECT * FROM UC
                ':this.value;" onfocus="this.select()" onclick="this.value
                ='';">
            <input class="submit"type="submit" value="">
          </form>
        </ul>

      </div>
      <div class="section">
        <a id='end' href="#"></a>
      </div>
    </div>
  </div>
</body>
</html>
```

C.17 Table.php

```

<html>
<body>

<!DOCTYPE html>
<html>
<head>
```

```
  <meta charset="UTF-8">
  <title>UCbase - UC Data Mining</title>
  <link rel="icon" href="http://localhost/appswebsitetemplate/favicon.ico" />
  <link rel="stylesheet" href="css/style.css" type="text/css">
</head>
<body>
  <div class="page">
    <div class="sidebar">
      <div id="logo">
        <a href="index.html"><img src="images/logo.png" alt="logo"></a>
      </div>
      <ul class="navigation">
        <li>
          <a href="index.html">Home</a>
        </li>
        <li class="selected">
          <a href="uc_data_mining.php">UC Data Mining</a>
        </li>
        <li>
          <a href="related_works.html">Related Works</a>
        </li>
        <li>
          <a href="about_us.html">About Us</a>
        </li>
      </ul>
      <p id="suggest"> Type an ID, a GENE NAME <br> or what you need and GO !</p>
      <form action="">
        <input type="text" value="Rapid Search" onblur="this.value=!this.value?'Rapid Search':this.value;" onfocus="this.select()" onclick="this.value='';">
        <input type="submit" value="" onclick="alert('Not yet implemented!');">
      </form>

      <p id="mail">Comments, questions? <br>
        Email <a href="mailto:vinxlemons@gmail.com">vinxlemons@gmail.com</a></p>
    </div>
    <div class="content">
      <div>
        <h2>Search Result</h2>
      </div>
      <div class="data_mining">
        <div class="result">
          <h2 style='font-size:15px;'>
            <?php
              $query = $_GET["query"];
              $query = str_replace("<","try",$query);
              $query = filter_var($query, FILTER_SANITIZE_STRING);
              $query = str_replace("try","<",$query);
              echo $query;
            ?>
```

```
56          </h2>
57        </div>
58        <div class="section" id="result">
59 <?php
60 $con = mysql_connect("localhost","client","is1fs");
61 if (!$con)
62   {
63   die('Could not connect: ' . mysql_error());
64   }
65
66 mysql_select_db("UCbase", $con);
67 $query = $_GET["query"];
68 $query = str_replace("<","try",$query);
69 $query = filter_var($query, FILTER_SANITIZE_STRING);
70 $query = str_replace("try","<",$query);
71
72 $query = html_entity_decode($query, ENT_QUOTES);
73 $result = mysql_query($query);
74
75 if($result != false){
76   echo "<table id='big'>";
77   $i=0;
78   $nohits=1;
79   while($row = mysql_fetch_array($result, MYSQL_ASSOC))
80   {
81     $nohits =0;
82     if($i%2 == 0) echo "<tr class ='alt'>";
83     else echo "<tr>";
84
85     if($i==0){
86       foreach($row as $key => $value)
87       {
88       echo "<th>";
89       echo $key;
90       echo "</th>";
91       }
92       echo "</tr>";
93     }
94
95     if($i==0) echo "<tr>";
96     foreach($row as $x=>$x_value)
97     {
98       echo "<td>";
99       if($x == "UC_NAME"){
100              $uc_name = $x_value;
101              echo "<a href='result.php?uc_id=".$uc_name."'>".$x_value."</a>
                     ";
102      }
103      else if($x == "UC_NAME"){
104              $uc_name = $x_value;
105              echo $x_value;
106      }
107      else if($x == "DEFINITION" && $x_value !=""){
```

```
            $x_value = preg_replace('!(http|ftp|scp)(s)?:\/\/[a-zA-Z0-9.?&_=/]+!', "<a target='_blank' href=\"\\0\">\\0</a>", $x_value);
            echo $x_value;
      }
      else if($x == "SEQUENCE") {
        echo strtolower(substr($x_value, 0, 13)."...");
        echo "<a class='button' href='sequence.php?uc_id=".$uc_name."'> Get Sequence!</a>"; }
      else
        echo $x_value;
      echo "</td>";

    }
    echo "</tr>";
    $i++;
  }
  if($nohits)
    echo "<p style='text-align:center; color:black; font-size:24px;'>NO HITS FOUND.</p>";
  echo "</table>";
}
else{
  echo "<p style='margin-left:5px; font-size:24px; color:black;'>Query error.</p>";
   $message = 'Invalid query: ' . mysql_error() . "\n";
    $message .= 'Whole query: ' . $query;
    echo $message;
}

mysql_close($con);
?>
        </div>
      </div>
    </div>
  </div>
</body>
</html>

</body>
</html>
```

C.18 Sequence.php

```

<?php $con = mysql_connect("localhost","client","is1fs");
  if (!$con){
      die('Could not connect: ' . mysql_error());
    }

  $uc = filter_input(INPUT_GET,"uc_id", FILTER_SANITIZE_STRING);
  mysql_select_db("UCbase", $con);
  $result = mysql_query("SELECT * FROM UC WHERE UC_NAME='".$uc."'");
```

```
  $row = mysql_fetch_array($result);
  echo "<html> <body> <p style='font-size:12px;'> double click on it to select! <br><br>".$row["SEQUENCE"]."</p></body></html>";
  mysql_close($con);
?>
```

C.19 Result_search_uc.php

```

<?php

  $chr = filter_input(INPUT_GET,"chr_id", FILTER_SANITIZE_STRING);
  $num_bp1 = filter_input(INPUT_GET,"num_bp1", FILTER_SANITIZE_NUMBER_INT);
  $num_bp2 = filter_input(INPUT_GET,"num_bp2", FILTER_SANITIZE_NUMBER_INT);

  $query = "SELECT * FROM UC WHERE UC.CHR= '".$chr."' AND UC.START BETWEEN ".$num_bp1." AND ".$num_bp2."";
  header("location: http://localhost/UCbase/table.php?query=".$query);
?>
```

C.20 Result_search_pat.php

```

<?php

  $pat = filter_input(INPUT_GET,"pat", FILTER_SANITIZE_STRING);
  if($pat == "retinal cancer")
    $query = "SELECT UC_NAME,NAME,MIN(LEVEL)AS LEVEL FROM UC INNER JOIN (
        SELECT ENSAMBL_GENE_ID,NAME,LEVEL FROM RELATED_TO INNER JOIN (
        SELECT ID,NAME,LEVEL FROM PATHOLOGY WHERE PREORDER >= (SELECT
        PREORDER FROM PATHOLOGY WHERE ID='DOID:4645') AND POSTORDER <=(
        SELECT POSTORDER FROM PATHOLOGY WHERE ID='DOID:4645'))AS B WHERE
        RELATED_TO.ID = B.ID)AS C WHERE UC.ENSAMBL_GENE_ID = C.
        ENSAMBL_GENE_ID group by uc_name, name ORDER BY LEVEL";
  else if($pat == "optic nerve disease")
    $query = "SELECT UC_NAME,NAME,MIN(LEVEL)AS LEVEL FROM UC INNER JOIN (
        SELECT ENSAMBL_GENE_ID,NAME,LEVEL FROM RELATED_TO INNER JOIN (
        SELECT ID,NAME,LEVEL FROM PATHOLOGY WHERE PREORDER >= (SELECT
        PREORDER FROM PATHOLOGY WHERE ID='DOID:1891') AND POSTORDER <=(
        SELECT POSTORDER FROM PATHOLOGY WHERE ID='DOID:1891'))AS B WHERE
        RELATED_TO.ID = B.ID)AS C WHERE UC.ENSAMBL_GENE_ID = C.
        ENSAMBL_GENE_ID group by uc_name, name ORDER BY LEVEL";
  else if($pat == "retinal cell cancer")
    $query = "SELECT UC_NAME,NAME,MIN(LEVEL) AS LEVEL FROM UC INNER JOIN (
        SELECT ENSAMBL_GENE_ID,NAME,LEVEL FROM RELATED_TO INNER JOIN (
        SELECT ID,NAME,LEVEL FROM PATHOLOGY WHERE PREORDER >= (SELECT
        PREORDER FROM PATHOLOGY WHERE ID='DOID:771') AND POSTORDER <=(
        SELECT POSTORDER FROM PATHOLOGY WHERE ID='DOID:771') )AS B WHERE
        RELATED_TO.ID = B.ID)AS C WHERE UC.ENSAMBL_GENE_ID = C.
        ENSAMBL_GENE_ID group by uc_name, name ORDER BY LEVEL";
  else if($pat == "retinoblastoma")
    $query = "SELECT UC_NAME,NAME, MIN(LEVEL) AS LEVEL FROM UC INNER JOIN (
        SELECT ENSAMBL_GENE_ID,NAME,LEVEL FROM RELATED_TO INNER JOIN (
```

```
          SELECT ID,NAME,LEVEL FROM PATHOLOGY WHERE PREORDER >= (SELECT
          PREORDER FROM PATHOLOGY WHERE ID='DOID:768') AND POSTORDER <=(
          SELECT POSTORDER FROM PATHOLOGY WHERE ID='DOID:768') )AS B WHERE
          RELATED_TO.ID = B.ID)AS C WHERE UC.ENSAMBL_GENE_ID = C.
          ENSAMBL_GENE_ID group by uc_name, name ORDER BY LEVEL";
  else $query = "SELECT UC_NAME,NAME,MIN(LEVEL)AS LEVEL FROM UC INNER JOIN
       (SELECT ENSAMBL_GENE_ID,NAME,LEVEL FROM RELATED_TO INNER JOIN (SELECT
        ID,NAME,LEVEL FROM PATHOLOGY WHERE PREORDER >= (SELECT PREORDER FROM
        PATHOLOGY WHERE NAME='".$pat."') AND POSTORDER <=(SELECT POSTORDER
       FROM PATHOLOGY WHERE NAME='".$pat."') ORDER BY LEVEL)AS B WHERE
       RELATED_TO.ID = B.ID)AS C WHERE UC.ENSAMBL_GENE_ID = C.
       ENSAMBL_GENE_ID group by uc_name, name ORDER BY LEVEL";

  header("location: http://localhost/UCbase/table.php?query=".$query);
?>
```

C.21 Result_gene.php

```

<html>
<body>

<!DOCTYPE html>
<!-- Website template by freewebsitetemplates.com -->
<html>
<head>
  <meta charset="UTF-8">
  <title>UCbase - UC Data Mining</title>
  <link rel="icon" href="http://localhost/appswebsitetemplate/favicon.ico"
      />
  <link rel="stylesheet" href="css/style.css" type="text/css">
</head>
<body>
  <div class="page">
    <div class="sidebar">
      <div id="logo">
        <a href="index.html"><img src="images/logo.png" alt="logo"></a>
      </div>
      <ul class="navigation">
        <li>
          <a href="index.html">Home</a>
        </li>
        <li class="selected">
          <a href="uc_data_mining.php">UC Data Mining</a>
        </li>
        <li>
          <a href="related_works.html">Related Works</a>
        </li>
        <li>
          <a href="about_us.html">About Us</a>
        </li>
      </ul>
```

```
      <p id="suggest"> Type an ID, a GENE NAME <br> or what you need and GO !</p>
      <form action="">
        <input type="text" value="Rapid Search" onblur="this.value=!this.value?'Rapid Search':this.value;" onfocus="this.select()" onclick="this.value='';">
        <input type="submit" value="" onclick="alert('Not yet implemented!');">
      </form>

      <p id="mail">Comments, questions? <br>
        Email <a href="mailto:vinxlemons@gmail.com">vinxlemons@gmail.com</a></p>
    </div>
    <div class="content">
      <div>
        <h2>Search Result</h2>
      </div>
      <div class="data_mining">
        <div class="result">
          <h2> <?php
          $gene_name = filter_input(INPUT_GET,"gene_name", FILTER_SANITIZE_STRING);
          echo $gene_name;
          echo "</h2>" ;
        echo "</div>";
          $con = mysql_connect("localhost","client","is1fs");
          if (!$con){
              die('Could not connect: ' . mysql_error());
            }
          $gene = "";
          mysql_select_db("UCbase", $con);
          $result = mysql_query("SELECT * FROM GENE WHERE WIKIGENE_NAME='" .$gene_name."'");

          $row = mysql_fetch_array($result); //solo una dovrebbe essere
          echo "<table id='result'>";
          if($row != null){
            foreach($row as $key => $value) //elimino indici numerici
            {
                if(is_int($key)){
                    unset($row[$key]);
                }
            }
            foreach($row as $x=>$x_value)
            {  $i=0;
              if($i%2 == 0) echo "<tr class ='alt'>";
              else echo "<tr>";
              echo "<td class = 'one'>";
              echo ucfirst(strtolower($x));
              echo "</td>";
              echo "<td>";
              if($x == "SEQUENCE") echo substr($x_value, 0, 20)."...";
              else {if($x_value != "") echo $x_value; else echo "none";}
```

```
81              if($x == "SEQUENCE")
82                echo "<a class='button' href='sequence.php?uc_id=".$uc."'>
                      Get Sequence!</a>";
83              echo "</td>";
84              echo "</tr>";
85              $i++;
86            }
87            echo "</table>";
88            $gene = $row["ENSAMBL_GENE_ID"];
89          }
90          else
91            echo "<p style='margin-left:22px;margin-top: 20px; font-size:24
                  px; color:black;'>Not found.</p>";
92
93  $result = mysql_query("SELECT SPLICING.* FROM SPLICING,HAS WHERE HAS.
        ENSAMBL_GENE_ID ='".$gene."' AND HAS.EVENT_TYPE_COD = SPLICING.
        EVENT_TYPE_COD ");
94    $i =0;
95    while(($row = mysql_fetch_array($result, MYSQL_ASSOC)) != null){
96      if($row != null && $i == 0) echo "<div class='result'><h2> Gene Splicing
            </h2></div>";
97        echo "<table id='result'>";
98        foreach($row as $x=>$x_value)
99          {  $i=0;
100           if($i%2 == 0) echo "<tr class ='alt'>";
101           else echo "<tr>";
102           echo "<td class = 'one'>";
103           echo ucfirst(strtolower($x));
104           echo "</td>";
105           echo "<td>";
106           if($x_value != "") echo $x_value; else echo "none";
107           echo "</td>";
108           echo "</tr>";
109           $i++;
110       }
111       echo "</table>";
112       $i++;
113   }
114
115 $result = mysql_query("SELECT UC.* FROM UC WHERE UC.ENSAMBL_GENE_ID ='".
        $gene."'");
116
117   $i =0;
118   while(($row = mysql_fetch_array($result, MYSQL_ASSOC)) != null){
119     if($row != null && $i == 0) echo "<div class='result'><h2> Gene correled
            UC</h2></div>";
120       echo "<table id='result'>";
121         $uc ="";
122           foreach($row as $x=>$x_value)
123           {  $i=0;
124             if($i%2 == 0) echo "<tr class ='alt'>";
125             else echo "<tr>";
126             echo "<td class = 'one'>";
127             echo ucfirst(strtolower($x));
```

```
128             echo "</td>";
129             echo "<td>";
130             if($x == "UC_NAME") {$uc = $x_value; $x_value ="<a href='result
                    .php?uc.id=".$uc."'>".$uc."</a>"; }
131             if($x == "SEQUENCE") echo substr($x_value, 0, 20)."...";
132             else {if($x_value != "") echo $x_value; else echo "none";}
133             if($x == "SEQUENCE")
134               echo "<a class='button' href='sequence.php?uc_id=".$uc."'>
                      Get Sequence!</a>";
135             echo "</td>";
136             echo "</tr>";
137             $i++;
138           }
139       echo "</table>";
140       $i++;
141   }
142
143 $result = mysql_query("SELECT PATHOLOGY.* FROM PATHOLOGY,RELATED_TO WHERE
        RELATED_TO.ENSAMBL_GENE_ID ='".$gene."' AND RELATED_TO.ID = PATHOLOGY.
       ID");
144
145   $i =0;
146   while(($row = mysql_fetch_array($result, MYSQL_ASSOC)) != null){
147     if($row != null && $i == 0) echo "<div class='result'><h2> Gene correled
            pathology</h2></div>";
148       echo "<table id='result'>";
149       foreach($row as $x=>$x_value)
150         {
151             if($i%2 == 0) echo "<tr class ='alt'>";
152             else echo "<tr>";
153             echo "<td class = 'one'>";
154             echo ucfirst(strtolower($x));
155             echo "</td>";
156             echo "<td>";
157             if($x == "ID"){ $pat = $x_value; }
158             if($x == "DEFINITION" && $x_value !=""){
159               $x_value = preg_replace('!(http|ftp|scp)(s)?:\/\/[a-zA-Z0-9.?&
                      _=/-]+!', "<a target='_blank' href=\"\\0\">\\0</a>",
                      $x_value);
160               echo $x_value;
161             }
162             else if($x_value != "") echo $x_value; else echo "none";
163             echo "</td>";
164             echo "</tr>";
165             $i++;
166       }
167       $i=0;
168       $result2 = mysql_query("SELECT SYNONYM.NAME FROM SYNONYM WHERE
            SYNONYM.ID ='".$pat."'");
169       while(($row2 = mysql_fetch_array($result2, MYSQL_ASSOC)) != null){
170         foreach($row2 as $x2=>$x_value2)
171         {
172             if($i%2 == 0) echo "<tr class ='alt'>";
173             else echo "<tr>";
```

```
            echo "<td class = 'one'>";
            echo "Synonym".$i;
            echo "</td>";
            echo "<td>";
            if($x_value2 != "") echo $x_value2; else echo "none";
            echo "</td>";
            echo "</tr>";
            $i++;
        }
      }
      $i=0;
      $result3 = mysql_query("SELECT XREF.REF FROM XREF WHERE XREF.ID ='".
          $pat."'");
      while(($row3 = mysql_fetch_array($result3, MYSQL_ASSOC)) != null){
        foreach($row3 as $x3=>$x_value3)
        {
            if($i%2 == 0) echo "<tr class ='alt'>";
            else echo "<tr>";
            echo "<td class = 'one'>";
            echo "Xref".$i;
            echo "</td>";
            echo "<td>";
            if($x_value3 != "") echo $x_value3; else echo "none";
            echo "</td>";
            echo "</tr>";
            $i++;
        }
      }
      echo "</table>";
      $i++;
  }
mysql_close($con);
        ?>
        <div class="section">
          <a href="#"></a>
        </div>
      </div>
    </div>
  </div>
</body>
</html>

</body>
</html>
```

C.22 Result.php

```

<html>
<body>

<!DOCTYPE html>
<html>
```

```
<head>
  <meta charset="UTF-8">
  <title>UCbase - UC Data Mining</title>
  <link rel="icon" href="http://localhost/appswebsitetemplate/favicon.ico" />
  <link rel="stylesheet" href="css/style.css" type="text/css">
</head>
<body>
  <div class="page">
    <div class="sidebar">
      <div id="logo">
        <a href="index.html"><img src="images/logo.png" alt="logo"></a>
      </div>
      <ul class="navigation">
        <li>
          <a href="index.html">Home</a>
        </li>
        <li class="selected">
          <a href="uc_data_mining.php">UC Data Mining</a>
        </li>
        <li>
          <a href="related_works.html">Related Works</a>
        </li>
        <li>
          <a href="about_us.html">About Us</a>
        </li>
      </ul>
      <p id="suggest"> Type an ID, a GENE NAME <br> or what you need and GO !</p>
      <form action="">
        <input type="text" value="Rapid Search" onblur="this.value=!this.value?'Rapid Search':this.value;" onfocus="this.select()" onclick="this.value='';">
        <input type="submit" value="" onclick="alert('Not yet implemented!');">
      </form>

      <p id="mail">Comments, questions? <br>
        Email <a href="mailto:vinxlemons@gmail.com">vinxlemons@gmail.com</a></p>
    </div>
    <div class="content">
      <div>
        <h2>Search Result</h2>
      </div>
      <div class="data_mining">
        <div class="result">
          <h2> <?php
          $uc = filter_input(INPUT_GET,"uc_id", FILTER_SANITIZE_STRING);
          echo $uc;
          echo "</h2>" ;
        echo "</div>";
          $con = mysql_connect("localhost","client","is1fs");
          if (!$con){
```

```
55              die('Could not connect: ' . mysql_error());
56            }
57
58          mysql_select_db("UCbase", $con);
59          $result = mysql_query("SELECT * FROM UC WHERE UC_NAME='".$uc."'"
                );
60
61          $row = mysql_fetch_array($result); //solo una dovrebbe essere
62          echo "<table id='result'>";
63          if($row != null){
64            foreach($row as $key => $value) //elimino indici numerici
65            {
66                if(is_int($key)){
67                    unset($row[$key]);
68                }
69            }
70            $i=0;
71            foreach($row as $x=>$x_value)
72            {
73              if($i%2 == 0) echo "<tr class ='alt'>";
74              else echo "<tr>";
75              echo "<td class = 'one'>";
76              echo ucfirst(strtolower($x));
77              echo "</td>";
78              echo "<td>";
79              if($x == "SEQUENCE") echo substr($x_value, 0, 20)."...";
80              else {if($x_value != "") echo $x_value; else echo "none";}
81              if($x == "SEQUENCE")
82                echo "<a class='button' href='sequence.php?uc_id=".$uc."'>
                      Get Sequence!</a>";
83              echo "</td>";
84              echo "</tr>";
85              $i++;
86            }
87            echo "</table>";
88          }
89          else
90            echo "<p style='margin-left:22px;margin-top: 20px; font-size:24
                  px; color:black;'>Not found.</p>";
91            $gene = ""; //definisco per dopo..
92  if($row["ENSAMBL_GENE_ID"] != ""){
93          echo "<div class='result'><h2>".$row["ENSAMBL_GENE_ID"]."</h2></
                div>";
94
95          $gene = $row["ENSAMBL_GENE_ID"];
96
97          mysql_select_db("UCbase", $con);
98          $result = mysql_query("SELECT * FROM GENE WHERE ENSAMBL_GENE_ID
                ='".$gene."'");
99
100         $row = mysql_fetch_array($result); //solo una dovrebbe essere
101         echo "<table id='result'>";
102         foreach($row as $key => $value) //elimino indici numerici
103         {
```

```
                if(is_int($key)){
                    unset($row[$key]);
                }
            }
            $i=0;
            foreach($row as $x=>$x_value)
              {
              if($i%2 == 0) echo "<tr class ='alt'>";
              else echo "<tr>";
              echo "<td class = 'one'>";
              echo ucfirst(strtolower($x));
              echo "</td>";
              echo "<td>";
              echo $x_value;
              echo "</td>";
              echo "</tr>";
              $i++;
            }
            echo "</table>";
}

$result = mysql_query("SELECT SPLICING.* FROM SPLICING,HAS WHERE HAS.
    ENSAMBL_GENE_ID ='".$gene."' AND HAS.EVENT_TYPE_COD = SPLICING.
    EVENT_TYPE_COD ");
  $i =0;
  while(($row = mysql_fetch_array($result, MYSQL_ASSOC)) != null){
    if($row != null && $i == 0) echo "<div class='result'><h2> Gene Splicing
        </h2></div>";
      echo "<table id='result'>";
      $i=0;
      foreach($row as $x=>$x_value)
        {
            if($i%2 == 0) echo "<tr class ='alt'>";
            else echo "<tr>";
            echo "<td class = 'one'>";
            echo ucfirst(strtolower($x));
            echo "</td>";
            echo "<td>";
            if($x_value != "") echo $x_value; else echo "none";
            echo "</td>";
            echo "</tr>";
            $i++;
      }
      echo "</table>";
      $i++;
  }
$result = mysql_query("SELECT SNP.* FROM SNP, UC WHERE SNP.CHR = UC.CHR
    AND SNP.START_UC = UC.START AND UC.UC_NAME = '".$uc."'");


  $i =0;
  while(($row = mysql_fetch_array($result, MYSQL_ASSOC)) != null){
    if($row != null && $i == 0) echo "<div class='result'><h2> UC correled
        SNP</h2></div>";
```

```
153         echo "<table id='result'>";
154         $i=0;
155         foreach($row as $x=>$x_value)
156           {
157               if($i%2 == 0) echo "<tr class ='alt'>";
158               else echo "<tr>";
159               echo "<td class = 'one'>";
160               echo ucfirst(strtolower($x));
161               echo "</td>";
162               echo "<td>";
163               if($x_value != "") echo $x_value; else echo "none";
164               echo "</td>";
165               echo "</tr>";
166               $i++;
167         }
168         echo "</table>";
169         $i++;
170     }
171
172   $result = mysql_query("SELECT PATHOLOGY.* FROM PATHOLOGY,RELATED_TO WHERE
          RELATED_TO.ENSAMBL_GENE_ID ='".$gene."' AND RELATED_TO.ID = PATHOLOGY.
         ID");
173
174     $i =0;
175     while(($row = mysql_fetch_array($result, MYSQL_ASSOC)) != null){
176       if($row != null && $i == 0) echo "<div class='result'><h2> Gene correled
              pathology</h2></div>";
177
178         if(strstr($row['ID'] , "bis" ))
179           continue;
180         echo "<table id='result'>";
181         foreach($row as $x=>$x_value)
182           {
183               if($i%2 == 0) echo "<tr class ='alt'>";
184               else echo "<tr>";
185               echo "<td class = 'one'>";
186               echo ucfirst(strtolower($x));
187               echo "</td>";
188               echo "<td>";
189               if($x == "ID"){ $pat = $x_value; }
190               if($x == "DEFINITION" && $x_value !=""){
191                 $x_value = preg_replace('!(http|ftp|scp)(s)?:\/\/[a-zA-Z0-9.?&
                      _=/-]+!', "<a target='_blank' href=\"\\0\">\\0</a>",
                      $x_value);
192                 echo $x_value;
193               }
194               else if($x_value != "") echo $x_value; else echo "none";
195               echo "</td>";
196               echo "</tr>";
197               $i++;
198         }
199         $i=0;
200         $result2 = mysql_query("SELECT SYNONYM.NAME FROM SYNONYM WHERE
              SYNONYM.ID ='".$pat."'");
```

```
      while(($row2 = mysql_fetch_array($result2, MYSQL_ASSOC)) != null){
        foreach($row2 as $x2=>$x_value2)
        {
            if($i%2 == 0) echo "<tr class ='alt'>";
            else echo "<tr>";
            echo "<td class = 'one'>";
            echo "Synonym".$i;
            echo "</td>";
            echo "<td>";
            if($x_value2 != "") echo $x_value2; else echo "none";
            echo "</td>";
            echo "</tr>";
            $i++;
        }
      }
      $i=0;
      $result3 = mysql_query("SELECT XREF.REF FROM XREF WHERE XREF.ID ='".
          $pat."'");
      while(($row3 = mysql_fetch_array($result3, MYSQL_ASSOC)) != null){
        foreach($row3 as $x3=>$x_value3)
        {
            if($i%2 == 0) echo "<tr class ='alt'>";
            else echo "<tr>";
            echo "<td class = 'one'>";
            echo "Xref".$i;
            echo "</td>";
            echo "<td>";
            if($x_value3 != "") echo $x_value3; else echo "none";
            echo "</td>";
            echo "</tr>";
            $i++;
        }
      }
      echo "</table>";
      $i++;
  }
mysql_close($con);
        ?>
        <div class="section">
          <a href="#"></a>
        </div>
      </div>
    </div>
  </div>
</body>
</html>

</body>
</html>
```

C.23 Related_works.html

```
1
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>UCbase - Related Works</title>
  <link rel="icon" href="http://localhost/appswebsitetemplate/favicon.ico" />
  <link rel="stylesheet" href="css/style.css" type="text/css">
</head>
<body>
  <div class="page">
    <div class="sidebar">
      <div id="logo">
        <a href="index.html"><img src="images/logo.png" alt="logo"></a>
      </div>
      <ul class="navigation">
        <li>
          <a href="index.html">Home</a>
        </li>
        <li>
          <a href="uc_data_mining.php">UC Data Mining</a>
        </li>
        <li class="selected">
          <a href="related_works.html">Related Works</a>
        </li>
        <li>
          <a href="about_us.html">About Us</a>
        </li>
      </ul>
      <p id="suggest"> Type an ID, a GENE NAME <br> or what you need and GO !</p>
      <form action="">
        <input type="text" value="Rapid Search" onblur="this.value=!this.value?'Rapid Search':this.value;" onfocus="this.select()" onclick="this.value='';">
        <input type="submit" value="" onclick="alert('Not implemented yet!');">
      </form>

      <p id="mail">Comments, questions? <br>
        Email <a href="mailto:vinxlemons@gmail.com">vinxlemons@gmail.com</a></p>
    </div>
    <div class="content">
      <div>
        <h2>Related works</h2>
        <p>In this section you will find all information relating to the UC sequences even outside of our research laboratory.
        Paper and news about it will be posted below.
        </p>

      </div>
<div class="blog">
        <div class="date">
```

```
49          <span>12-03</span>
50          <span>2013</span>
51        </div>
52        <div>
53          <h2>Articles and Papers</h2>
54          <div>
55            border
56          </div>
57        </div>
58        <ul>
59          <li>
60            <div>
61              <div>
62                <a href="#"><img src="images/dna.jpg" alt=""></a>
63                <a href="http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2686429
                      /">.pdf</a>
64              </div>
65              <div>
66                <h3>UCbase & miRfunc: a database of ultraconserved sequences
                       and microRNA function</h3>
67                <p>One hundred and eighty-one ultraconserved sequences (UCRs
                      ) longer than 200 bases were discovered in the genomes
                      of human, mouse and rat. These are DNA sequences showing
                       100% identity among the three species... </p>
68                <a onclick="alert('Not implemented yet!');">Read more>></a>
69              </div>
70            </div>
71          </li>
72          <li>
73            <div>
74              <div>
75                <a href="#"><img src="images/dna.jpg" alt=""></a>
76                <a href="http://database.oxfordjournals.org/content/2011/
                      bar049.full">.pdf</a>
77              </div>
78              <div>
79                <h3>The BioMart project</h3>
80                <p>BioMart is a unique open source data federation
                      technology that provides unified access to distributed
                      databases storing a wide range of data. This DATABASE
                      issue recognizes BioMart's outstanding contributions to
                      bioinformatics and documents the achievements of the
                      BioMart community, which has grown impressively over the
                       last ten years to become what it is today...</p>
81                <a onclick="alert('Not implemented yet!');">Read more>></a>
82              </div>
83            </div>
84          </li>
85          <li>
86            <div>
87              <div>
88                <a href="#"><img src="images/dna.jpg" alt=""></a>
89                <a href="http://www.mirbase.org/">.pdf</a>
90              </div>
```

```
            <div>
              <h3>miRBase: the microRNA database</h3>
              <p>The miRBase database is a searchable database of
                  published miRNA sequences and annotation. Each entry in
                  the miRBase Sequence database represents a predicted
                  hairpin portion of a miRNA transcript (termed mir in the
                   database), with information on the location and
                  sequence of the mature miRNA sequence (termed miR). Both
                   hairpin and mature sequences are available for
                  searching and browsing, and entries can also...</p>
              <a onclick="alert('Not implemented yet!');">Read more>></a>
            </div>
          </div>
        </li>

      </ul>
      <div class="section">
        <a href="#"></a>
      </div>
    </div>
  </div>
</body>
</html>
```

C.24 Blastpat.php

```

<html>
<body>

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>UCbase - UC Data Mining</title>
  <link rel="icon" href="http://localhost/appswebsitetemplate/favicon.ico"
      />
  <link rel="stylesheet" href="css/style.css" type="text/css">
</head>
<body>
  <div class="page">
    <div class="sidebar">
      <div id="logo">
        <a href="index.html"><img src="images/logo.png" alt="logo"></a>
      </div>
      <ul class="navigation">
        <li>
          <a href="index.html">Home</a>
        </li>
        <li class="selected">
          <a href="uc_data_mining.php">UC Data Mining</a>
        </li>
```

```
26          <li>
27            <a href="related_works.html">Related Works</a>
28          </li>
29          <li>
30            <a href="about_us.html">About Us</a>
31          </li>
32        </ul>
33        <p id="suggest"> Type an ID, a GENE NAME <br> or what you need and GO
              !</p>
34        <form action="">
35          <input type="text" value="Rapid Search" onblur="this.value=!this.
                value?'Rapid Search':this.value;" onfocus="this.select()"
                onclick="this.value='';">
36          <input type="submit" value="" onclick="alert('Not yet implemented!')
                ;">
37        </form>
38
39        <p id="mail">Comments, questions? <br>
40          Email <a href="mailto:vinxlemons@gmail.com">vinxlemons@gmail.com</a
                ></p>
41      </div>
42      <div class="content">
43        <div>
44          <h2>Search Result</h2>
45        </div>
46        <div class="data_mining">
47          <div class="result">
48            <h2> <?php
49            $seq = filter_input(INPUT_GET,"sequence", FILTER_SANITIZE_STRING);
50            $pat = filter_input(INPUT_GET,"pathology", FILTER_SANITIZE_STRING)
                ;
51            if(strlen($seq)<=15) echo "Blast: ".$seq; else echo "Blast: ".
                substr($seq,0,15)."...";
52            echo "</h2>";
53          echo "</div>";
54          echo "<div class='blast' >";
55             $id = fopen("/var/www/UCbase/ncbi-blast-2.2.27+/bin/query.fasta"
                , 'w') or die("can't open file") ;
56             fwrite($id, ">".$seq."\n".$seq);
57
58            exec("cd /var/www/UCbase/ncbi-blast-2.2.27+/bin/; ./blastn -db
                UCdb -query query.fasta -task blastn-short 2>&1", $output);
59
60            //mi connetto al database
61            $con = mysql_connect("localhost","client","is1fs");
62            if (!$con)
63            {
64              die('Could not connect: ' . mysql_error());
65            }
66
67            mysql_select_db("UCbase", $con);
68
69            echo "<p>";
70
```

```
71          $final; //preparo output finale
72          $j=0;
73          $num_uc = 0;
74          $non_uc_count = 0;
75          $swap = 0;
76          for($i=10; $i<count($output); $i++){
77                $yn=false;
78                if(strpos($output[$i], "Lambda") === 0) { $final[$j] =
                      $output[0] ;break;}
79
80                if($i ==10){ $output[$i] = $output[$i]." (".substr($output
                      [11],11).")";
81                  $final[$j] = $output[$i];
82                  $j++;
83                }
84                else if($i != 11 and $i!= 12 and $i!= 16){
85                  if($i == 18) {
86                    $output[$i] = " Score E Score_pat Pathology";
87                  }
88                  if($i == 19) {
89                    $output[$i] = "Sequences alignments: (Bits) Value (Level)
                          Name";
90                  }
91                  if((strpos($output[$i], "Query=") === 0)){
92                    $output[$i] = "<b>Click on the uc link to read specific
                          Info.<br><br>".$output[$i]."</b>";
93                  }
94
95                  if((strpos($output[$i], "lcl") === 0)) {
96                    $output[$i] = str_replace(" ", " ", $output[$i]);
97                    $output[$i] = str_replace(" ", ".", $output[$i]);
98
99                    preg_match("/uc\.[0-9]*/", $output[$i],$uc[$num_uc]);
100                   $uc[$num_uc] = $uc[$num_uc][0];
101                   preg_match("/[0-9]+\.[0-9]+/", $output[$i],$score[
                          $num_uc]);
102                   //echo "uc mum ".$num_uc." : ".$score[$num_uc][0];
103
104                   $query = "SELECT UC_NAME FROM
105                   (SELECT DISTINCT UC_NAME FROM UC INNER JOIN (SELECT
                          ENSAMBL_GENE_ID,NAME FROM RELATED_TO INNER JOIN (
                          SELECT ID,NAME FROM PATHOLOGY WHERE PREORDER >= (
                          SELECT PREORDER FROM PATHOLOGY WHERE NAME='".$pat."')
                           AND POSTORDER <=(SELECT POSTORDER FROM PATHOLOGY
                          WHERE NAME='".$pat."') ORDER BY LEVEL)AS B WHERE
                          RELATED_TO.ID = B.ID)AS C WHERE UC.ENSAMBL_GENE_ID =
                          C.ENSAMBL_GENE_ID) as X
106                   WHERE UC_NAME = '".$uc[$num_uc]."'";
107                   $result = mysql_query($query);
108                   if(!($row = mysql_fetch_array($result, MYSQL_NUM)))
109                     {$yn=true; $non_uc[$non_uc_count] = $uc[$num_uc] ;
                            $non_uc_count++;}
110
111                   //collego patologia
```

```
112                     $query = "SELECT NAME FROM
113                     (SELECT DISTINCT UC_NAME,NAME FROM UC INNER JOIN (SELECT
                            ENSAMBL_GENE_ID,NAME FROM RELATED_TO INNER JOIN (
                            SELECT ID,NAME FROM PATHOLOGY WHERE PREORDER >= (
                            SELECT PREORDER FROM PATHOLOGY WHERE NAME='".$pat."')
                             AND POSTORDER <=(SELECT POSTORDER FROM PATHOLOGY
                            WHERE NAME='".$pat."') ORDER BY LEVEL)AS B WHERE
                            RELATED_TO.ID = B.ID)AS C WHERE UC.ENSAMBL_GENE_ID =
                            C.ENSAMBL_GENE_ID) as X
114                     WHERE UC_NAME = '".$uc[$num_uc]."'";
115                     $result = mysql_query($query);
116                     if(($row = mysql_fetch_array($result, MYSQL_NUM)))
117                     {
118                       $output[$i] = $output[$i].".........<b>".$row[0]."</b>";
119                       $pathology[$num_uc] = $row[0];
120
121                     }
122                     else { $pathology[$num_uc] = "null";}
123                     if(($row = mysql_fetch_array($result, MYSQL_NUM)))
124                       $output[$i] = $output[$i]." and more..";
125
126                     $index_row[$num_uc] = $i;
127
128                     //echo " ".$uc[$num_uc].": ".$pathology[$num_uc]." ".
                            $index_row[$num_uc]." ";
129
130                     $query = "select level from PATHOLOGY where name = '".
                            $pathology[$num_uc]."'";
131                     $result = mysql_query($query);
132                     if($row = mysql_fetch_array($result, MYSQL_NUM))
133                     {
134                       $level[$num_uc] = $row[0];
135                       list($pre, $post) = explode('<b>', $output[$i]);
136                       $output[$i] = $pre.$level[$num_uc].".........<b>".$post;
137                     }
138                     else
139                       $level[$num_uc] = 100;
140                     //echo $level[$num_uc]."<br>";
141
142                     //ordino anche per patologia rispetto a tutte le
                            precedenti
143                     if(!isset($non_uc))
144                       $non_uc[] = "";
145                     for($h=$num_uc-1; $h>=0; $h--){
146
147                       if($score[$h] == $score[$num_uc] and $level[$h] > $level
                              [$num_uc] and !in_array($uc[$h], $non_uc)){
148                         //echo "sono ".$uc[$num_uc]." scambio con ".$uc[$h
                                ]."<br>";
149                         $swap = $index_row[$h];
150                         $h_final = $h;
151                       }
152                     }
153                     if($swap){
```

```
154                     list($uc[$h_final],$uc[$num_uc]) = array($uc[$num_uc],
                            $uc[$h_final]);
155                     list($level[$h_final],$level[$num_uc]) = array($level[
                            $num_uc],$level[$h_final]);
156                   }
157
158                   $num_uc++;
159                 }
160
161
162
163                 $output[$i] = str_replace(" ", " ", $output[$i]);
164                 if(!(strpos($output[$i], "lcl") === 0))
165                   $output[$i] = preg_replace('!uc\.[0-9]*!', "<a style='
                          color: blue;' target='_blank' href=\"result.php?uc.id
                          =\\0\">\\0</a>",$output[$i]);
166                 else
167                   $output[$i] = preg_replace('!uc\.[0-9]*!', "<a style='
                          color: blue;' href=\"#\\0\">\\0</a>",$output[$i]);
168                 if($i != 10) $output[$i]= $output[$i]."<br>";
169
170                 if($swap){
171                   list($final[$swap-$non_uc_count-13],$output[$i]) = array(
                          $output[$i],$final[$swap-$non_uc_count-13]);
172
173                   $swap = 0;
174                   }
175
176                 if(!$yn)
177                   { $final[$j] = $output[$i];
178                     $j++;
179                   }
180
181               }
182
183           }
184
185
186           //scrivo
187           echo $final[0];
188           echo "</p>";
189         echo "</div>";
190         echo "<div class='blast' id ='special'>";
191           echo "<p> <br>";
192           $comment = false;
193           for($i=3; $i<count($final); $i++){
194 /*
195             if((strpos($final[$i], "lcl") === 0)){
196                 echo "sono numero: ".$i;
197               }
198 */
199             if((strpos($final[$i], "Query=") === 0)) echo "<b>Click on the
                    uc link to read specific Info.<br><br></b>";
200             //grassetto le lettere
```

```
            $final[$i] = preg_replace('![AGCT]+[AGCT]+!', "<b>\\0</b>", $final[$i]);
            if((strpos($final[$i], ">lcl") === 0)){
              preg_match("/uc\.[0-9]+/", $final[$i],$prova);
              if(in_array($prova[0], $non_uc))
                {$i +=11; continue;}
              else{
                  echo "</p></div> ";
                  if($prova[0] == $uc[0])
                    echo "<h2 style ='margin :10px 0px 10px 20px; width:400 px; text-align: center;'> Unsorted Specific Info</h2> ";
                  echo "<div class='blast'> <p> <a name='".$prova[0]."'></a><b> Click on the link for more info about the uc.</b><br><br>";
                }
            }
            else if (strpos($final[$i], "BLAS") === 0){
              echo "</p></div> <div class='blast'> <p>";
            }
/*
            else if((strpos($final[$i], "<br>") === 0) and (strpos($final[$i+1], ">lcl")=== 0) ) {
              echo "cazzoooooooooooooo";
              if($comment)
                {echo "-->"; $comment=false;}
              else
                echo "</p> </div> <div class='blast'> <p>";
            }
*/
            if((strpos($final[$i], "<br>") === 0) and $i <> count($final)-1 and (strpos($final[$i+1], "<br>") === 0) )
              $i++;
            echo $final[$i];

          }


          echo "</p>";
        echo "</div>";
          mysql_close($con);
          ?>
        <div class="section">
          <a href="#"></a>
        </div>
      </div>
    </div>
  </div>
</body>
</html>

</body>
</html>
```

C.25 Blastn.php

```

<html>
<body>

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>UCbase - UC Data Mining</title>
  <link rel="icon" href="http://localhost/appswebsitetemplate/favicon.ico" />
  <link rel="stylesheet" href="css/style.css" type="text/css">
</head>
<body>
  <div class="page">
    <div class="sidebar">
      <div id="logo">
        <a href="index.html"><img src="images/logo.png" alt="logo"></a>
      </div>
      <ul class="navigation">
        <li>
          <a href="index.html">Home</a>
        </li>
        <li class="selected">
          <a href="uc_data_mining.php">UC Data Mining</a>
        </li>
        <li>
          <a href="related_works.html">Related Works</a>
        </li>
        <li>
          <a href="about_us.html">About Us</a>
        </li>
      </ul>
      <p id="suggest"> Type an ID, a GENE NAME <br> or what you need and GO !</p>
      <form action="">
        <input type="text" value="Rapid Search" onblur="this.value=!this.value?'Rapid Search':this.value;" onfocus="this.select()" onclick="this.value='';">
        <input type="submit" value="" onclick="alert('Not yet implemented!');">
      </form>

      <p id="mail">Comments, questions? <br>
        Email <a href="mailto:vinxlemons@gmail.com">vinxlemons@gmail.com</a></p>
    </div>
    <div class="content">
      <div>
        <h2>Search Result</h2>
      </div>
```

```
46        <div class="data_mining">
47          <div class="result">
48            <h2> <?php
49            $seq = filter_input(INPUT_GET,"sequence", FILTER_SANITIZE_STRING);
50            if(strlen($seq)<=15) echo "Blast: ".$seq; else echo "Blast: ".
                substr($seq,0,15)."...";
51            echo "</h2>";
52          echo "</div>";
53          echo "<div class='blast' >";
54             $id = fopen("/var/www/UCbase/ncbi-blast-2.2.27+/bin/query.fasta"
                , 'w') or die("can't open file") ;
55             fwrite($id, ">".$seq."\n".$seq);
56
57            exec("cd /var/www/UCbase/ncbi-blast-2.2.27+/bin/; ./blastn -db
                UCdb -query query.fasta -task blastn-short 2>&1", $output);
58            echo "<p>";
59
60            $final; //preparo output finale
61            $j=0;
62            $num_uc = 0;
63            for($i=10; $i<count($output); $i++){
64
65                  if(strpos($output[$i], "Lambda") === 0) { $final[$j] =
                      $output[0] ;break;}
66
67                  if($i ==10){ $output[$i] = $output[$i]." (".substr($output
                      [11],11).")";
68                    $final[$j] = $output[$i];
69                    $j++;
70                  }
71                  else if($i != 11 and $i!= 12 and $i!= 16){
72                    if((strpos($output[$i], "lcl") === 0)){
73                      preg_match("/uc\.[0-9]*/", $output[$i],$uc[$num_uc]);
74                      $uc[$num_uc] = $uc[$num_uc][0];
75                      $num_uc++;
76                      $output[$i] = str_replace(" ", ".", $output[$i]);
77                    }
78
79                    if((strpos($output[$i], "Query=") === 0)){
80                      $output[$i] = "<b>Click on the uc link to read specific
                          Info.<br><br>".$output[$i]."</b>";
81                    }
82                    if($i != 11)$output[$i] = str_replace(" ", " ", $output
                        [$i]);
83                    if(!(strpos($output[$i], "lcl") === 0))
84                      $output[$i] = preg_replace('!uc\.[0-9]*!', "<a style='
                          color: blue;' target='_blank' href=\"result.php?uc.id
                          =\\0\">\\0</a>",$output[$i]);
85                    else
86                      $output[$i] = preg_replace('!uc\.[0-9]*!', "<a style='
                          color: blue;' href=\"#\\0\">\\0</a>",$output[$i]);
87                    if($i != 10) $final[$j]= $output[$i]."<br>";
88                    else $final[$j] = $output[$i];
89                    $j++;
```

```
90                }
91
92          }
93
94          echo $final[0];
95          echo "</p>";
96        echo "</div>";
97        echo "<div class='blast'>";
98          echo "<p> <br>";
99
100         for($i=3; $i<count($final); $i++){
101 /*
102           if((strpos($final[$i], "lcl") === 0)){
103               echo "sono numero: ".$i;
104             }
105 */
106           $final[$i] = preg_replace('![AGCT]+[AGCT]+!', "<b>\\0</b>",
                 $final[$i]);
107
108           if((strpos($final[$i], ">lcl") === 0)){
109             preg_match("/uc\.[0-9]+/", $final[$i],$prova);
110             if($prova[0] == $uc[0]){
111               echo "<form style='width: 360px;'action='blastpat.php'
                     method='GET'> <input type='text' name='sequence' value='
                     ".$seq."' style='display:none;'>";
112               echo "<span>  &nbspFilter with pathology: </span> <
                     select name='pathology'>";
113               $con = mysql_connect("localhost","client","is1fs");
114               if (!$con)
115               {
116                 die('Could not connect: ' . mysql_error());
117               }
118
119               mysql_select_db("UCbase", $con);
120               $query = "SELECT DISTINCT NAME FROM PATHOLOGY ";
121
122               $result = mysql_query($query);
123
124               while($row = mysql_fetch_array($result, MYSQL_NUM))
125               {
126                 if($row[0]!=""){
127                   if($row[0] =="disease")
128                     echo '<option value="'.$row[0].'" selected="selected">'
                          .$row[0].'</option>';
129                   else
130                     echo '<option value="'.$row[0].'" >'.$row[0].'</option>
                          ';
131                 }
132               }
133               echo "<input class ='submit' type='submit' value=''></select
                     ></form>";
134             }
135             echo "</p></div> ";
136             if($prova[0] == $uc[0])
```

```
                echo "<h2 style ='margin :10px 0px 10px 20px; width:400px; text-align: center;'> Specific Info</h2>";
              echo "<div class='blast'> <p> <a name='".$prova[0]."'></a><b> Click on the link for more info about the uc.</b><br><br>";
            }
            else if (strpos($final[$i], "BLAS") === 0){
              echo "</p></div> <div class='blast'> <p>";
            }
/*
            else if((strpos($final[$i], "<br>") === 0) and (strpos($final[$i+1], ">lcl")=== 0) ) {
              echo "cazzoooooooooooooo";
              if($comment)
                {echo "-->"; $comment=false;}
              else
                echo "</p> </div> <div class='blast'> <p>";
            }
*/
            if((strpos($final[$i], "<br>") === 0) and $i <> count($final)-1 and (strpos($final[$i+1], "<br>") === 0) )
              $i++;
            echo $final[$i];

          }
          echo "</p>";
        echo "</div>";
          ?>
        <div class="section">
          <a href="#"></a>
        </div>
      </div>
    </div>
  </div>
</body>
</html>

</body>
</html>
```

C.26 About_us.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>UCbase - About Us</title>
  <link rel="icon" href="http://localhost/appswebsitetemplate/favicon.ico" />
  <link rel="stylesheet" href="css/style.css" type="text/css">
</head>
<body>
```

```
  <div class="page">
    <div class="sidebar">
      <div id="logo">
        <a href="index.html"><img src="images/logo.png" alt="logo"></a>
      </div>
      <ul class="navigation">
        <li>
          <a href="index.html">Home</a>
        </li>
        <li>
          <a href="uc_data_mining.php">UC Data Mining</a>
        </li>
        <li>
          <a href="related_works.html">Related Works</a>
        </li>
        <li class="selected">
          <a href="about_us.html">About Us</a>
        </li>
      </ul>
      <p id="suggest"> Type an ID, a GENE NAME <br> or what you need and GO !</p>
      <form action="">
        <input type="text" value="Rapid Search" onblur="this.value=!this.value?'Rapid Search':this.value;" onfocus="this.select()" onclick="this.value='';">
        <input type="submit" value="" onclick="alert('Not yet implemented!');">
      </form>

      <p id="mail">Comments, questions? <br>
        Email <a href="mailto:vinxlemons@gmail.com">vinxlemons@gmail.com</a></p>
    </div>
    <div class="content">
      <div>
        <h2>About Us</h2>
        <h3>Information Systems Group - University of Modena and Reggio Emilia</h3>
        <p>The work of the ISGroup, here at the Computer Engineering Department (DII) of the University of Modena and Reggio Emilia, mainly focuses on the design and development of new systems, algorithms and data structures for the access and management of Information.</p>
        <h3>Join Our work!</h3>
        <p>If you're experiencing issues and concerns about our work, please email us or help us to create a better work working together with our research group.</p>
        <p>Design version: beta.<br>Code version: beta.<br><br>
        <h3>Contact Us</h3>
        <span><b>Telephone Nos. :</b> -----, ------</span>
        <span><b>Email :</b> vinxelmons@gmail.com</span>
        <span><b>Street Address :</b> ----- Modena, Italy </span>
      </div>
      <div class="links">
```

```
53          <div class="date">
54            <span>13-02</span>
55            <span>2013</span>
56          </div>
57          <div>
58            <h2>Useful links</h2>
59            <div>
60              border
61            </div>
62          </div>
63          <ul>
64            <li>
65              <a href="#"><img src="images/dna.jpg" alt=""></a>
66            </li>
67            <li>
68              <a href="#"><img src="images/dna.jpg" alt=""></a>
69
70            </li>
71            <li>
72              <a href="#"><img src="images/dna.jpg" alt=""></a>
73
74            </li>
75            <li>
76              <a href="#"><img src="images/dna.jpg" alt=""></a>
77            </li>
78          </ul>
79        </div>
80      </div>
81    </div>
82  </body>
83  </html>
```

C.27 Style.css

```
1
2   /* Website template by freewebsitetemplates.com */
3   /*------------------------- Layout styles ------------------------*/
4   body{
5     margin: 0;
6     background: url(../images/bg-body.gif) repeat-x #205eb0;
7   }
8   .page{
9     width: 950px;
10    padding: 0 5px;
11    margin: 0 auto;
12    position: relative;
13  }
14
15  /*------------------------- Sidebar ------------------------*/
16  .sidebar {
17    float:left;
18    padding:80px 0 0;
19    width:248px;
```

```
20  }
21  .sidebar div#logo {
22    background:url(../images/interface.png) no-repeat 0 -142px;
23    height:200px;
24    left:-85px;
25    padding:60px 0 0;
26    position:absolute;
27    top:0;
28    width:383px;
29  }
30  .sidebar div#logo a {
31    display:block;
32    height:57px;
33    margin:0 0 0 90px;
34    outline:none;
35    width:230px;
36  }
37  .sidebar div#logo a img {
38    border:0;
39  }
40  .sidebar ul.navigation {
41    background:url(../images/border-dashed.gif) repeat-x left bottom;
42    list-style:none;
43    margin:80px 0 0 20px;
44    padding:0 0 10px;
45    text-align:center;
46    width:192px;
47    position:relative;
48  }
49  .sidebar ul.navigation li {
50    height:36px;
51    margin:0 0 10px;
52  }
53  .sidebar ul.navigation li.selected {
54    background:url(../images/interface.gif) no-repeat;
55  }
56  .sidebar ul.navigation li.selected a,.sidebar ul.navigation li a:hover {
57    color:#d3d4de;
58  }
59  .sidebar ul.navigation li a {
60    color:#ffffff;
61    font-family:Times New Roman;
62    font-size:20px;
63    font-weight:700;
64    line-height:36px;
65    outline:none;
66    text-decoration:none;
67    text-shadow:0 2px 0 #00007c;
68  }
69  .sidebar form {
70    background:url(../images/border-dashed.gif) repeat-x left bottom;
71    margin:15px 0 0 20px;
72    overflow:hidden;
73    padding:0 0 20px;
```

```
  width:192px;
}
.sidebar input:first-child {
  background:none;
  background-color:#dadaed;
  border:0;
  color:#1325f0;
  cursor:auto;
  float:left;
  font-family:helvetica;
  font-size:12px;
  font-style:italic;
  font-weight:700;
  height:auto;
  margin:0;
  padding:5px 9px;
  width:132px;
}
.sidebar input {
  background:url(../images/interface.gif) no-repeat 0 -56px;
  border:0;
  cursor:pointer;
  float:right;
  height:25px;
  width:30px;
}

.sidebar #suggest {
  color:#ffffff;
  font-family:Times New Roman;
  font-size:12px;
  margin-left:45px;
  width:160px;
}

.sidebar #mail {
  color:#ffffff;
  font-family:Times New Roman;
  font-size:12px;
  margin:135px 0 0 20px;
  text-align:center;
  width:200px;
}

/*------------------------- Content ------------------------*/
 .content{
  float:left;
  padding:55px 0 24px;
  width:702px;
}
.content .article{
  margin:0 0 0 24px;
}
.content .article h2{
```

```
/* color:#FEDD14;*/
  color:#FFFFFF;
  font-family:Times New Roman;
  font-size:35px;
  margin:0;
  text-shadow:0 2px 10px #00007c;
}
.content .article p{
  color:#fff;
  font-family:Times New Roman;
  font-size:16px;
  font-style:normal;
  line-height:21px;
  margin:10px 0 15px;
  text-align:justify;
}

.content .blog{
  background-color:#dce3e8;
  margin:24px 0 0 24px;
  padding:18px 0 0;
}
.content .blog .date,.content .links .date{
  background:url(../images/interface.png) no-repeat;
  float:left;
  height:54px;
  margin:0 0 0 -24px;
  width:90px;
}
.content .blog .date span:first-child,.content .links .date span:first-child{
  font-size:14px;
  margin:0 0 0 35px;
  padding:2px 0 0;
}
.content .blog .date span,.content .links .date span{
  color:#ffecc2;
  display:block;
  font-family:Times New Roman;
  font-size:12px;
  margin:0 0 0 45px;
}
.content .blog div,.content .links div{
  float:left;
  margin:0;
}
.content .blog div h2,.content .links div h2{
  color:#205eb0;
  float:left;
  font-family:Times New Roman;
  font-size:34px;
  margin:4px 0 0 18px;
  text-shadow:0 1px 0 #fff7e4;
}
.content .blog div div,.content .links div div{
```

```
182   background:url(../images/border-dashed2.gif) repeat-x center center;
183   margin:15px 0 0 10px;
184   text-indent:-99999px;
185   width:210px;
186 }
187 .content .blog ul{
188   background:url(../images/border-dashed2.gif) repeat-x center bottom;
189   clear:both;
190   list-style:none;
191   margin:0 0 0 85px;
192   overflow:hidden;
193   padding:15px 0 25px;
194   width:527px;
195 }
196 .content .blog ul li{
197   margin:50px 0 0;
198   overflow:hidden;
199 }
200 .content .blog ul li div{
201   background:none;
202   float:none;
203   margin:0;
204   overflow:hidden;
205   text-indent:0;
206 }
207 .content .blog ul li div div:first-child{
208   background:none;
209   margin:0;
210   width:84px;
211 }
212 /*.content .blog ul li div div:first-child a:first-child{
213   background:url(../images/shadow.jpg) no-repeat center bottom;
214   height:100px;
215   width:84px;
216 }*/
217 .content .blog ul li div div:first-child a{
218   background-color:#989eb3;
219   color:#000000;
220   display:block;
221   font-family:Times New Roman;
222   font-size:16px;
223   font-style:normal;
224   height:83px;
225   line-height:83px;
226   outline:none;
227   text-align:center;
228   text-decoration:none;
229   text-indent:0;
230   text-shadow:none;
231   width:84px;
232 }
233 .content .blog ul li div div{
234   float:left;
235   margin:0 0 0 18px;
```

```
236   width:425px;
237 }
238 .content .blog ul li div div h3{
239   color:#0a578a;
240   font-family:Times New Roman;
241   font-size:22px;
242   font-style:normal;
243   font-weight:500;
244   margin:0;
245   text-transform:Capitalize;
246   text-shadow: none;
247 }
248 .content .blog ul li div div p{
249   color:#000000;
250   font-family:Times New Roman;
251   font-size:16px;
252   font-style:normal;
253   margin:12px 0 35px;
254   text-align:justify;
255 }
256 .content .blog ul li div div a{
257   color:#000000;
258   float:right;
259   font-family:Times New Roman;
260   font-size:14px;
261   font-style:italic;
262   outline:none;
263   text-decoration:none;
264   text-shadow:0 1px 0 #f6f8f5;
265 }
266 .content .blog ul li div div a:hover{
267   color:#0a578a;
268   cursor: hand;
269   cursor: pointer;
270 }
271 .content .blog .section{
272   float:none;
273   height:60px;
274   margin:0 0 0 85px;
275   width:527px;
276 }
277 .content .blog .section a:first-child{
278   height:48px;
279   outline:none;
280   background:url(../images/interface.png) no-repeat 0 -74px;
281   display: block;
282   width:48px;
283 }
284 .content .blog .section a{
285   color:#e8cb93;
286   margin-left: 245px;
287   margin-top: 10px;
288   font-family:Times New Roman;
289   font-style:italic;
```

```
290   line-height:84px;
291   outline:none;
292   text-decoration:none;
293 }
294 .content .blog .section a:hover{
295   color:#F89F1F;
296 }
297 .content .blog .section .paging{
298   background:none;
299   margin:39px 0 0 225px;
300   width:144px;
301 }
302 .content .blog .section .paging a{
303   background-color:#FEC97B;
304   color:#FFFBF5;
305   font-style:normal;
306   font-weight:700;
307   height:18px;
308   line-height:17px;
309   margin:0 0 0 12px;
310   outline:none;
311   text-align:center;
312   text-indent:0;
313   width:66px;
314 }
315 .content .blog .section .paging a:hover{
316   background-color:#e6cc93;
317   color:#ffeabf;
318 }
319 .content div{
320   margin:0 0 0 24px;
321 }
322 .content div h2{
323   color:#ffffff;
324   font-family:'Times New Roman';
325   font-size:35px;
326   margin:0;
327   text-shadow:#00007C 0 2px 10px;
328 }
329 .content div h3{
330   color:#ffffff;
331   font-family:Times New Roman;
332   font-size:18px;
333   font-style:italic;
334   font-weight:600;
335   margin:10px 0 0;
336   text-transform:uppercase;
337   text-shadow:#00007C 0 2px 0;
338 }
339 .content div p{
340   color:#FFFFFF;
341   font-family:'Times New Roman';
342   font-size:16px;
343   font-style:normal;
```

```
  line-height:21px;
  margin:10px 0 15px;
  text-align:justify;
}
.content div p a:hover{
  color:#FEDD14;
}
.content div p a{
  color:#fff;
  outline:none;
}
.content .links ul{
  background:none;
  clear:both;
  list-style:none;
  margin:0;
  overflow:hidden;
  padding:0px 0 50px;
}
.content .links ul li{
  float:left;
  margin:0 0 0 63px;
  width:84px;
}

.content .links ul li a{
  color:#ffebc4;
  display:block;
  font-family:Times New Roman;
  font-size:16px;
  font-style:normal;
  height:24px;
  line-height:25px;
  outline:none;
  text-align:center;
  text-decoration:none;
  width:84px;
  margin-bottom: 20px;
}
.content .data_mining h2{
  color:#ffffff;
  float:left;
  font-family:Times New Roman;
  font-size:34px;
  margin:4px 0 0 84px;
  padding:5px 120px;
  background-color: #205EB0;
  text-shadow:#00007C 0 2px 10px;

}
.content .data_mining select#tre{
width: 250px;
}
.content .data_mining select{
```

```
height:28px;
}
.content .data_mining input,select{
  background:none;
  float: left;
  background-color:#ffffff;
  border: 1px solid #5b76e3;
  color:#205EB0;
  cursor:auto;
  font-family:helvetica;
  font-size:14px;
  font-style:italic;
  font-weight:700;
  height:20px;
  margin:6px;
  padding:2px 0px 5px 9px;
  width:132px;
}

.content .data_mining input#second{
  width:560px;
}

.content .data_mining input.submit{
  background:url(../images/interface.gif) no-repeat 0 -56px;
  border:0;
  cursor:pointer;
  float:right;
  margin-top: 8px;
  height:25px;
  width:30px;
}

.content .data_mining span{
  color:#000000;
  display: block;
  font-family:"Times New Roman",Georgia,Serif;
  font-size:17px;
# font-style:bold;
  font-weight: bold;
  margin:10px 0 10px;
  float: left;
}

.content .data_mining h2#second{
  margin:20px 0 10px 84px;
  padding:5px 90px;
}

.content .data_mining div{
  margin:0;
  overflow:hidden;

}
```

```
.content .data_mining div div{
  background:url(../images/border-dashed2.gif) repeat-x scroll center center
      transparent;
  float:left;
  margin:15px 0 0 10px;
  text-indent:-99999px;
  width:325px;
}
.content div.result{
  background-color: #205eb0;
  width: 640px;
  margin-left: 20px;
}
.content div.blast#special{
overflow:scroll;
}
.content div.blast{
  background-color: #a1c5df;
  width: 640px;
  margin-left: 20px;
  margin-top: 10px;

}

.content div.result h2{
  margin-left: 10px;
  padding: 0;
}
.content .data_mining ul{
  background-color: #c0c8ed;
  list-style:none;
  margin: 8px 20px 0px 20px;
  overflow:hidden;
  width:630px;
  height: 41px;
  padding-left: 10px;
}
.content .data_mining .section#result{
  background:none;
  float:none;
  margin:10px 0 0 20px;
  padding:0 0 15px;
  overflow-x:scroll;
  overflow-y:scroll;
  height: 500px;
}

.content .data_mining .section{
  background:none;
  float:none;
  margin:0 0 0 87px;
  padding:0 0 15px;

}
```

```
.content .data_mining .section a#end{
  height:48px;
  margin:15px 0 0 225px;
  background:url(../images/interface.png) no-repeat 0 -74px;
  display: block;
  width:48px;
}
.content .appspage{
  background-color:#FFECC1;
  margin:24px 0 324px 45px;
  padding:18px 0 33px;
}
.content .appspage div h2{
  color:#F27109;
  float:left;
  font-family:Times New Roman;
  font-size:34px;
  margin:4px 0 0 18px;
  text-shadow:0 1px 0 #FFF7E4;
}
.content .appspage div div{
  background:url(../images/border-dashed2.gif) repeat-x scroll center center
      transparent;
  float:left;
  margin:15px 0 0 10px;
  text-indent:-99999px;
  width:350px;
}
.content .appspage .previous{
  background:url(../images/interface.png) no-repeat;
  float:left;
  height:54px;
  margin:0 0 0 -24px;
  text-align:center;
  width:90px;
}
.content .appspage .previous a{
  color:#FFECC2;
  font-family:Times New Roman;
  font-size:14px;
  line-height:35px;
  outline:none;
  text-align:center;
  text-decoration:none;
}
.content .appspage .section{
  background:url(../images/border-dashed2.gif) repeat-x scroll center bottom
      transparent;
  margin:0 0 0 63px;
  overflow:hidden;
  padding:0 0 99px;
  width:535px;
}
.content .appspage .section div:first-child{
```

```
557     float:left;
558     width:84px;
559   }
560   /*.content .appspage .section div:first-child a:first-child{
561     background:url(../images/shadow.jpg) no-repeat scroll center bottom
          transparent;
562     height:100px;
563     margin:0;
564     width:84px;*/
565   }
566   .content .appspage .section div:first-child a{
567     background-color:#F89F1F;
568     color:#FFECC1;
569     display:block;
570     font-family:Times New Roman;
571     font-size:16px;
572     font-style:normal;
573     height:24px;
574     line-height:25px;
575     margin:5px 0 0;
576     outline:none;
577     text-align:center;
578     text-decoration:none;
579     text-indent:0;
580     text-shadow:none;
581     width:84px;
582   }
583   .content .appspage .section div{
584     background:none;
585     margin:0 0 0 20px;
586     text-indent:0;
587     width:410px;
588   }
589   .content .appspage .section div h3{
590     color:#D13D01;
591     font-family:Times New Roman;
592     font-size:24px;
593     font-style:normal;
594     font-weight:400;
595     margin:0;
596     text-transform:none;
597   }
598   .content .appspage .section div p{
599     color:#F69F1C;
600     font-family:Times New Roman;
601     font-size:12px;
602     font-style:normal;
603     line-height:21px;
604     margin:12px 0 15px;
605     text-align:justify;
606   }
607   .content .appspage .section div ul{
608     list-style:none;
609     margin:0;
```

```
  padding:0;
}
.content .appspage .section div ul li{
  float:left;
  margin:0 0 0 7px;
}
.content .appspage .section div ul li a{
  outline:none;
}
.content div span{
  color:#fff;
  display:block;
  font-family:Times New Roman;
  font-size:16px;
  font-style:italic;
  margin:0 0 10px;
}
.content div span b{
  color:#ffd215;
  font-family:Times New Roman;
  font-size:16px;
  font-style:italic;
}
.content .article a:hover,.content .blog ul li div div:first-child a:hover,.content .links ul li a:hover,.content .data_mining ul li a:hover,.content .appspage .previous a:hover{
  color:#fff;
}
.content .blog ul li:first-child,.content .blog .section .paging a:first-child{
  margin:0;
  background:none;
}
.content .data_mining ul li:first-child,.content .appspage .section div ul li:first-child{
  margin: 0;
}
.content .blog ul li div div:first-child a:first-child img,.content .blog .section a:first-child img,.content .links ul li a:first-child img,.content .data_mining ul li a:first-child img,.content .data_mining .section a img,.content .appspage .section div:first-child a img,.content .appspage .section div ul li a img{
  border:0;
}
.content .blog ul li div div:first-child a:first-child img:hover,.content .links ul li a:first-child img:hover,.content .data_mining ul li a:first-child img:hover,.content .appspage .section div:first-child a img:hover{
  opacity:0.8;
}
.content .links,.content .data_mining{
  background-color:#DCE3E8;
  margin:24px 0 0 24px;
  padding:18px 0 0;
}
```

```
654
655 table#result
656 {
657 margin-top: 20px;
658 font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
659 width:640px;
660 margin-left: 20px;
661 margin-bottom: 20px;
662 border-collapse:collapse;
663 table-layout:fixed;
664 }
665 #result td, #result th
666 {
667 font-size:1em;
668 border:1px solid #ffffff;
669 padding:3px 7px 2px 7px;
670 }
671
672 #result tr.alt{
673 background-color:#cbd5ff;
674 }
675
676 #result td.one
677 {
678 width: 170px;
679 font-size:18px;
680 font-weight:bold;
681 text-align:right;
682 padding-top:5px;
683 padding-bottom:4px;
684 background-color:#205EB0;
685 color:#ffffff;
686 }
687
688 a.button{
689 width: 115px;
690 float: right;
691 text-align: center;
692 background-color:#205EB0;
693 color: #ffffff;
694 text-decoration: none;
695 margin-top:1px;
696 }
697
698 a.button:hover{
699 background-color:#3768f9;
700 }
701
702 table#big
703 {
704 font-family:Lucidatypewriter, monospace;
705 width:100%;
706 border-collapse:collapse;
707 }
```

```
#big td, #big th
{
font-size:12px;
border:1px solid #205EB0;
padding:3px 7px 2px 7px;
}
#big th
{
font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
font-size:14px;
text-align:left;
padding-top:5px;
padding-bottom:4px;
background-color:#205EB0;
color:#ffffff;
}
#big tr.alt td
{
color:#000000;
background-color:#e7eaf9;
}

div.blast p
{
  color:black;
  font-family:monospace;
  font-size: 12px;
  margin-left:10px;
}
```

Bibliografia

[1] Isgroup. Proposed Thesis. URL: http://www.isgroup.unimo.it/thesis.asp.

[2] C. J. Date and Hugh Darwen. *A Guide to the SQL Standard*. Addison-Wesley, 4th edition, 2002.

[3] Philipp K. Janert. *Data Analysis with Open Source Tools*. O'Reilly Media.

[4] GNU's Not Unix. The R Project for Statistical Computing. URL: http://www.r-project.org/.

[5] Microsoft. Microsoft Excel. URL: http://office.microsoft.com/it-it/excel/.

[6] D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. *Progetto di basi di dati relazionali*. Pitagora Editrice s.r.l., 4th edition, 2007.

[7] McKusick-Nathans Institute. Online Mendelian Inheritance in Manl. URL: http://www.omim.org/.

[8] EBI and EMBL. Ensembl. URL: http://www.ensembl.org/.

[9] Bioconductor. BiomaRt. URL: http://www.bioconductor.org/packages/release/bioc/html/biomaRt.html.

[10] Arek Kasprzyk. BioMart: driving a paradigm change in biological data management. *Oxford University Press*, 2011.

[11] Apache Software Foundation. openoffice. URL: http://www.openoffice.org/.

[12] Oracle Corporation. MySQL. URL: http://www.mysql.it/.

[13] The PHP Group. PHP: Hypertext Preprocessor. URL: http://www.php.net/.

[14] IHTSDO. SNOMED CT. URL: http://www.ihtsdo.org/snomed-ct/.

[15] The Open Biological and Biomedical Ontologies. OBO Foundry. URL: http://www.obofoundry.org/.

[16] G. Cabri and F. Zambonelli. *Programmazione a oggetti in Java: dai fondamenti a Internet*. Pitagora Editrice s.r.l., 2003.

[17] James Gosling and Sun Microsystems. Java. URL: http://www.java.com/.

[18] Eclipse Foundation. Eclipse. URL: http://www.eclipse.org/.

[19] Nomi Harris, John Day-Richter, Chris Mungall, Amina Abdulla, Nomi Harris, and Jennifer Deegan. Obo-Edit. URL: http://oboedit.org/.

[20] Pavel Zezula, Federica Mandreoli, and Riccardo Martoglia. Tree Signatures and Unordered XML Pattern Matching. 2004.

[21] Taccioli C, Fabbri E, Visone R, Volinia S, Calin GA, Fong LY, Gambari R, Bottoni A, Acunzo M, Hagan J, Iorio MV, Piovan C, Romano G, and Croce CM. UCbase & miRfunc: a database of ultraconserved sequences and microRNA function. *Database issue journal*, 2008.

[22] Gill Bejerano, Michael Pheasant, Igor Makunin, Stuart Stephen, W. James Kent, John S. Mattick, and David Haussler. Ultraconserved elements in the human genome. *Science Magazine*, 2004.

[23] Wikimedia Foundation. Wikipedia. URL: http://www.wikipedia.org.

[24] Istituto dell'Enciclopedia Italiana. Treccani. URL: http://www.treccani.it/.

[25] Sam Griffiths-Jones. The microRNA Registry. *Database issue journal*, 2003.

[26] R01RR025342. Human disease ontology. URL: http://diseaseontology.sourceforge.net/.

[27] National Center for Biotechnology Information. NCBI. URL: http://www.ncbi.nlm.nih.gov/.

[28] National Center for Biotechnology Information. BLAST. URL: http://blast.ncbi.nlm.nih.gov/Blast.cgi.