ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE FISICHE E NATURALI

Corso di Laurea Triennale in Informatica

BASI DI DATI BIOLOGICI AD ACCESSO PUBBLICO

Tesi di Laurea di:
Mauro Spalletti

Relatore:
Chiar.mo Prof. Danilo Montesi

Correlatori:
Prof.ssa Rita Casadio
Dott. Gianluca Tasco

Sessione II

Anno Accademico 2003-2004

INDICE

1 INTRODUZIONE

La ricerca nel campo della microbiologia volge oggi l'attenzione ad un'importante famiglia di proteine, quelle derivate da organismi termofili. Tali proteine, a differenza di quelle dei "normali" organismi, espletano la loro funzione a temperature elevate, quasi sempre superiori a 50°C. Tale prerogativa viene sfruttata, ad esempio, nella produzione industriale riducendo al minimo l'impiego di costosi dispositivi di raffreddamento o negli stessi processi come catalizzatori. Si ipotizza inoltre che i termofili siano stati i progenitori di tutte le altre forme di vita che popolano il pianeta.

1.1 Organismi Termofili

Gli sforzi vengono concentrati al momento sull'individuazione delle differenze tra proteine termofile e proteine mesofile strutturalmente simili. Fermo restando che le somiglianze strutturali spesso denotano affinità comportamentali, è facile capire perché questo tipo di ricerca sia estremamente interessante ai fini non solo scientifici ma anche industriali.

1.2 Basi di dati biologici

La gestione di una banca dati di proteine è alla base di un'efficiente ricerca scientifica. Tuttavia il principale organo di riferimento in tal senso, l'*RCSB*, che conta oltre 27.000 strutture, proprio a causa di questa mole di

dati non sempre è in grado di garantire la qualità dell'informazione; inoltre i dati presenti in questo database non sono correlati tra di loro in alcun modo particolare.

Nasce così l'esigenza di una base di dati contenente informazioni aggiornate e controllate, arricchite dei riferimenti interproteici necessari alla ricerca e disponibile all'intera comunità scientifica.

Prende vita in quest'ottica il progetto di ampliamento di *TOPDB*, una base di dati di proteine non condivisa nata in seno a precedenti elaborati di tesi. TOPDB oltre a contenere una replica controllata delle informazioni sulle proteine termofile utilizza gli accoppiamenti strutturali forniti dall'*FSSP* per determinare quali proteine mesofile sono simili a quelle in archivio, facilitando il compito di analisi e comparazione.

TOPDB è realizzato con *Microsoft® Active Server Page* (ASP), un ambiente di scripting lato server utilizzato per creare applicazioni web dinamiche ed interattive [Mic04]. Il linguaggio di scripting utilizzato è *Microsoft® Visual Basic® Scripting Edition* (VBScript) [Mic04b].

Il progetto si sviluppa in più fasi. Durante la prima fase viene effettuata un'attenta revisione delle procedure ed una sostanziosa riscrittura del codice, in modo da apportare ottimizzazioni ed aumentare la compatibilità delle pagine HTML generate. Nella seconda fase vengono affinati i meccanismi per l'autenticazione e create procedure per l'iscrizione, quindi, nella terza fase, vengono implementate le procedure per la gestione degli account che comprende la moderazione delle iscrizioni e l'attribuzione di privilegi.

Nell'ultima fase vengono presi in considerazione gli aggiornamenti. Si pone particolare attenzione alla gestione degli errori in fase di importazione dei dati ed alla salvaguardia dell'integrità delle informazioni, spesso minata dalla non sempre coerente codifica dei dati nell'RCSB.

Il progetto si conclude con l'implementazione di alcune utilità non originariamente previste, come le statistiche sul contenuto del database ed un modulo per la configurazione dell'applicazione.

2 PROTEINE

2.1 Aminoacidi

Le proteine sono eteropolimeri in grado di auto organizzarsi in contatto con il solvente polare. Queste biomolecole complesse sono importanti perché prendono parte a numerosi processi e funzioni biologicamente rilevanti: costituiscono una componente fondamentale di ogni cellula vivente. Inoltre sono la parte integrante e funzionale delle membrane biologiche e possono essere enzimi ed ormoni che rivestono un ruolo decisivo nei sistemi biologici complessi. Le proteine sono eteropolimeri risultanti dalla combinazione di unità monomeriche fondamentali, dette aminoacidi, in catene di lunghezza e complessità diverse. Tale caratteristica conferisce alle singole proteine proprietà chimico-fisiche diverse le une dalle altre e proprio ciò sta alla base della loro grande versatilità biologica. Esistono vari tipi di aminoacidi ma la quasi totalità delle proteine sono costituite a partire da soli 20 aminoacidi detti standard. Aminoacidi più rari e che ricoprono un ruolo marginale a livello proteico sono detti generalmente aminoacidi non standard.

Gli aminoacidi sono molecole organiche caratterizzate dalla presenza di un atomo di carbonio principale detto C_α, cui sono legati contemporaneamente un gruppo carbossilico (-COOH), un gruppo amminico (-NH2) ed un ulteriore catena laterale diversa da un aminoacido all'altro. Proprio quest'ultima differenzia gli aminoacidi fra loro. I 20 aminoacidi standard sono rappresentati in Figura 2.1.

Gli aminoacidi hanno caratteristiche fisiche paragonabili a quelle dei composti ionici quali punto di fusione elevato e scarsa solubilità in

solventi apolari. Tale carattere è legato alla possibilità di questi composti di esistere nella forma ionica dipolare detta anche zwitterone. Come si è già accennato, gli aminoacidi si differenziano per la catena laterale che è legata all'atomo di carbonio centrale. La natura chimica di tale catena influenza le principali proprietà degli aminoacidi. Ad esempio, aminoacidi che possiedono catene laterali di tipo alifatico e aromatico presentano uno spiccato carattere apolare, mentre la presenza nelle catene laterali di atomi come zolfo, ossigeno e azoto conferiscono all'aminoacido un carattere polare (che può essere di tipo acido basico o neutro). Quattro aminoacidi hanno catene laterali cariche. Gli aminoacidi basici hanno cariche positive ad un pH fisiologico sono lisina ed arginina. L'acido aspartico e l'acido glutammico sono invece carichi negativamente a valori di pH superiori a 3.

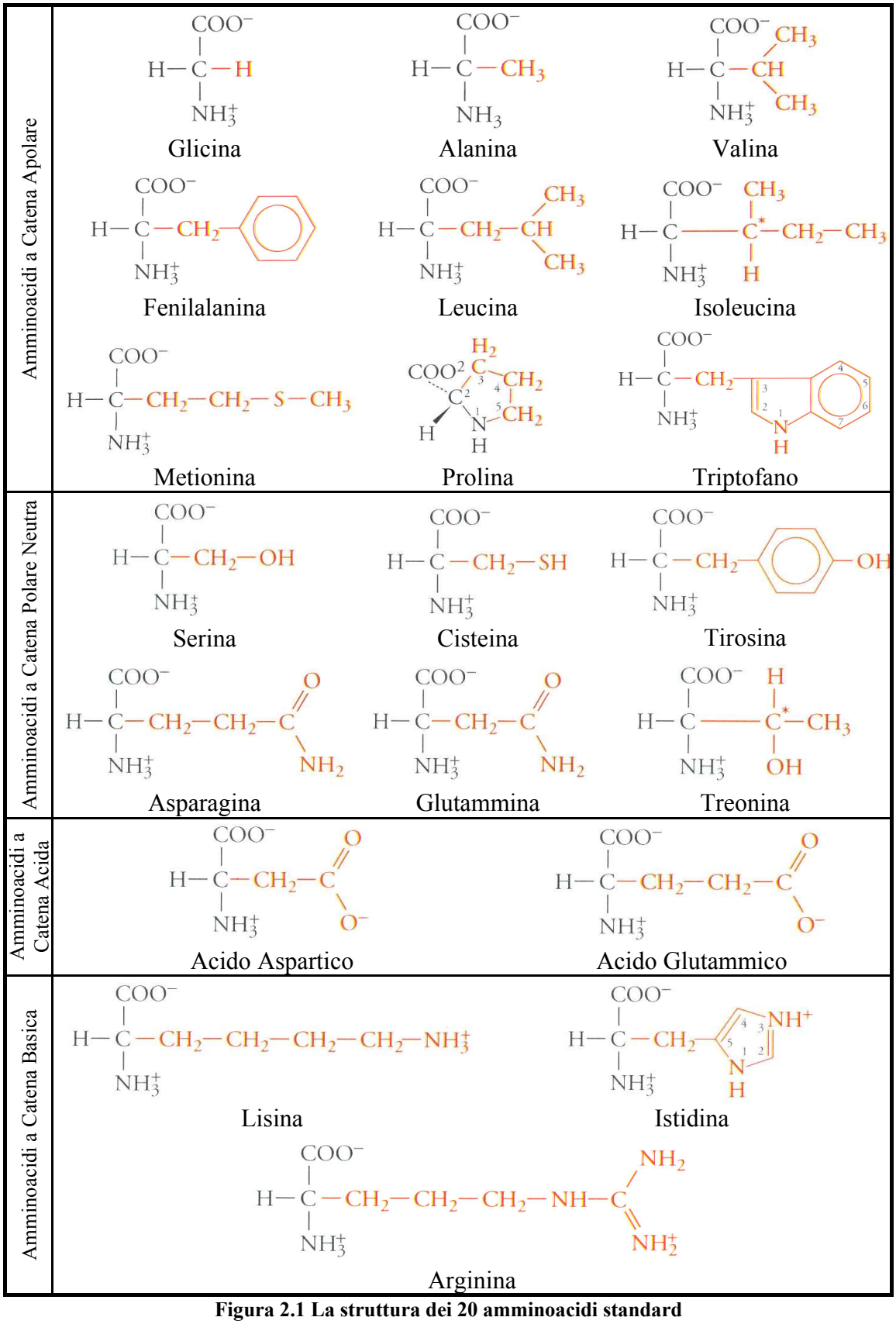


Figura 2.1 La struttura dei 20 amminoacidi standard

2.2 Legame peptidico e struttura delle proteine

Due aminoacidi possono legarsi mediante la formazione di un legame covalente tra il gruppo carbossilico di un aminoacido e il gruppo amminico dell'altro; il legame CO-NH che si origina è detto peptidico ed è il prodotto di una reazione di condensazione (eliminazione di una molecola d'acqua). Il risultato di tale reazione viene detto genericamente peptide, mentre le singole unità amminoacidiche vengono chiamati residui amminoacidici.

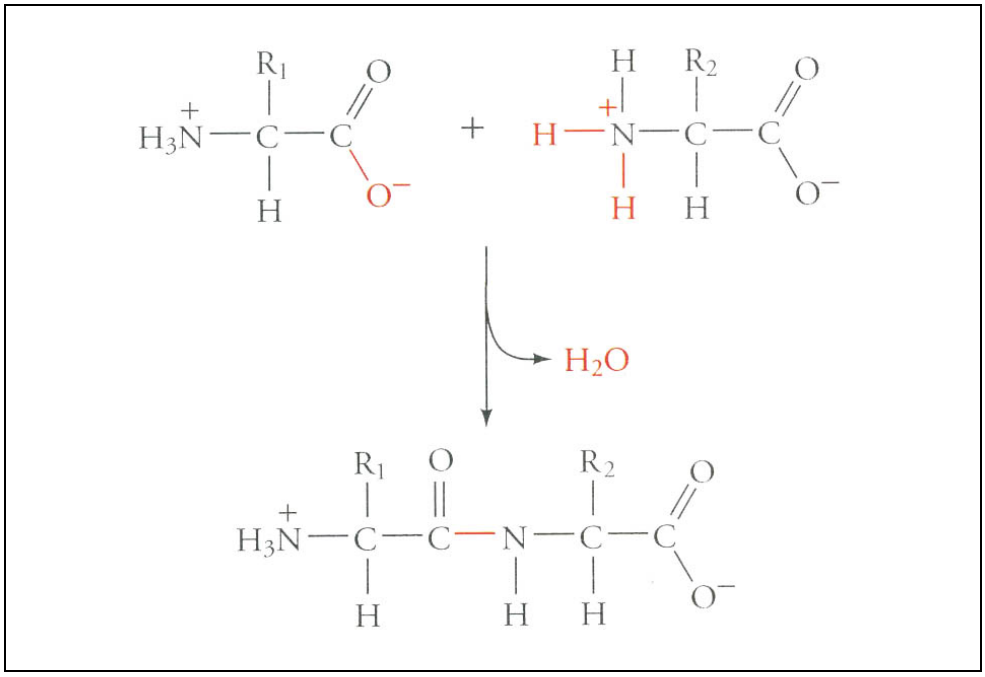


Figura 2.2 Schema della reazione di condensazione che porta alla formazione di un legame peptidico

Tale processo può coinvolgere più amminoacidi e portare alla formazione di una lunga catena peptidica, detta anche polipeptide. Un polipeptide é una struttura costituita da uno scheletro formato dalla successione dei legami C – C – N – C di più amminoacidi. Gli amminoacidi che si trovano alle estremità della catena sono chiamati residuo N-terminale (quello con il gruppo amminico libero che per convenzione viene posto a

sinistra nelle rappresentazioni) e residuo C-terminale (quello col gruppo carbossilico). Le proprietà chimico-fisiche di un polipeptide, soprattutto per quanto riguarda le sue relazioni con l'ambiente in cui si trova, sono determinate solo parzialmente dal suo scheletro. Infatti sono le catene laterali degli amminoacidi di partenza e la loro disposizione spaziale a determinare le principali caratteristiche di un polipeptide. Le proteine sono spesso costituite dall'interazione di più catene polipeptidiche non sempre uguali tra loro. La sequenza ordinata dei singoli residui della catena o delle catene costituenti una proteina è detta struttura primaria della proteina. La struttura primaria, come vedremo, riveste un ruolo fondamentale perché in essa, secondo l'ipotesi termodinamica di Anfinsen [Anf73], sono contenute tutte le informazioni necessarie a determinare la struttura tridimensionale della proteina in ambiente fisiologico.

Esistono tuttavia ulteriori livelli di organizzazione strutturale della sequenza amminoacidica. In particolare si possono distinguere:

- una *struttura secondaria* che descrive la disposizione spaziale locale degli atomi dello scheletro polipeptidico;
- una *struttura terziaria* che descrive la disposizione tridimensionale dell'intero polipeptide;
- una *struttura quaternaria* che descrive l'organizzazione spaziale delle catene polipeptidiche che costituiscono una proteina.

2.3 Interazioni tra residui

2.3.1 Interazioni elettrostatiche

Queste interazioni possono riguardare sia i gruppi ionici propri sia i gruppi caratterizzati da un'asimmetria nella distribuzione della densità elettronica, dovute ai legami covalenti della catena polipeptidica o dalla vicinanza di ioni carichi, che conferiscono loro un carattere dipolare. Tali interazioni, pur essendo abbastanza forti, hanno un'influenza limitata sul-

la stabilità di una proteina nativa in quanto l'energia libera legata alle coppie ioniche non compensa l'energia di solvatazione e l'entropia perse per la loro formazione. Questo spiega perché queste formazioni siano difficilmente conservate a livello evolutivo di proteine anche omologhe.

2.3.2 Forze di Van der Waals

Le forze di Van der Waals sono forze che entrano in gioco solo quando gli atomi si trovano ad una distanza inferiore ai 3-4Å e quindi agiscono principalmente nelle regioni ripiegate di una proteina nello stato nativo, mentre sono irrilevanti nelle catene "srotolate" delle proteine denaturate. Tali forze possono essere a carattere attrattivo o repulsivo a seconda della distanza tra gli atomi interagenti.

La natura di questo tipo di interazioni fa sì che il potenziale di Van der Waals possa essere schematizzato come un caso particolare di potenziale di Lennard-Jones del tipo riportato in Equazione 2.1.

$$E = -\frac{A}{R^6} + \frac{B}{R^{12}}$$

Equazione 2.1

dove A vale 10-77 J·m6 è il coefficiente legato al termine attrattivo legato alle forze di dispersione e B vale 10-136 J·m12 è il termine legato alla repulsione elettronica, mentre R è la distanza tra gli atomi.

Questa interazione fornisce energie abbastanza basse (circa 1 Kcal/mol) se riferite ad un sistema che coinvolge un numero limitato di atomi. Per un sistema complesso come una catena proteica il contributo totale all'energia dovuto a questo tipo di interazione diviene rilevante.

2.3.3 Legami a idrogeno

Se un idrogeno si trova legato covalentemente ad un atomo molto elettronegativo, l'orbitale elettronico di legame viene ad essere apprezza-

bilmente spostato verso quest'ultimo. Il contatto tra l'H ed un atomo che possegga una carica parziale negativa, a piccola distanza origina un legame di tipo essenzialmente elettrostatico. Molta importanza assumono i vincoli geometrici imposti dal legame ad idrogeno: l'idrogeno deve essere collineare all'atomo a cui è legato covalentemente e all'atomo col quale è in contatto.

Questo legame ha un'energia che a seconda degli atomi tra cui si instaura varia tra 3 e 6 Kcal/mol che è piuttosto alta se paragonata a quella degli altri legami non covalenti.

Tuttavia, se l'entità di tale interazione è fondamentale per la struttura proteica, essa è molto meno importante in ordine alla sua stabilità, questo perché in ambiente acquoso, dove la proteina tende a svolgersi, i gruppi che generano legami a idrogeno nella proteina tendono a fare altrettanto con le molecole di acqua dando luogo a legami ad idrogeno altrettanto energetici.

2.3.4 Effetto idrofobico

I residui non polari, in soluzione acquosa portano alla formazione intorno ad essi di strutture, dette clatrati, in cui l'acqua si organizza ordinatamente. Questo comporta una diminuzione dell'entropia del solvente. L'effetto idrofobico è pertanto una forza di tipo entropico che porta ai residui non polari ad avvicinarsi sottraendosi il più possibile al contatto col solvente.

2.3.5 Ponti disolfuro

Due residui di cisteina possono formare, in ambiente ossidato, un legame covalente che coinvolge i loro atomi di zolfo. Data la forza di questo legame, è l'unico tra tutti i legami descritti in questo capitolo ad essere considerato di tipo covalente.

2.3.6 Ponti salini

Due residui carichi di carica opposta possono dar luogo ad una interazione di tipo elettrostatico, detta ponte salino. La formazione di tali attrazioni ioniche non è particolarmente favorita in soluzione in quanto comporta una variazione di energia libera praticamente identica a quella di solvatazione. Questa interazione riguarda circa il 75% dei residui carichi presenti nelle proteine. Di norma tali formazioni si trovano localizzate sulla superficie della proteina. Le interazioni elettrostatiche agiscono a distanze di 3-4Å e in accordo con le leggi delle interazioni coulombiane tra cariche, tra dipoli e tra multipoli. Il contributo energetico di questo tipo di interazioni è di qualche Kcal/mol.

3 TERMOSTABILITÀ E SISTEMI BIOLOGICI

3.1 Temperatura come fattore ambientale

La temperatura è uno dei più importanti fattori ambientali in grado di influenzare la crescita e la sopravvivenza delle cellule a partire dalle prime forme di vita che si pensa siano comparse sulla terra 5 miliardi di anni fa.

All'aumentare della temperatura le reazioni chimiche ed enzimatiche della cellula procedono ad una velocità maggiore e la crescita diventa più rapida. Tuttavia al di sopra di certe temperature le proteine gli acidi nucleici e gli altri componenti cellulari possono andare incontro a denaturazione irreversibile. Esiste quindi un intervallo di temperatura in cui le funzioni metaboliche e la crescita aumentano e un punto in cui iniziano le reazioni di inattivazione, al di sopra del quale, le funzioni cellulari decadono bruscamente a zero.

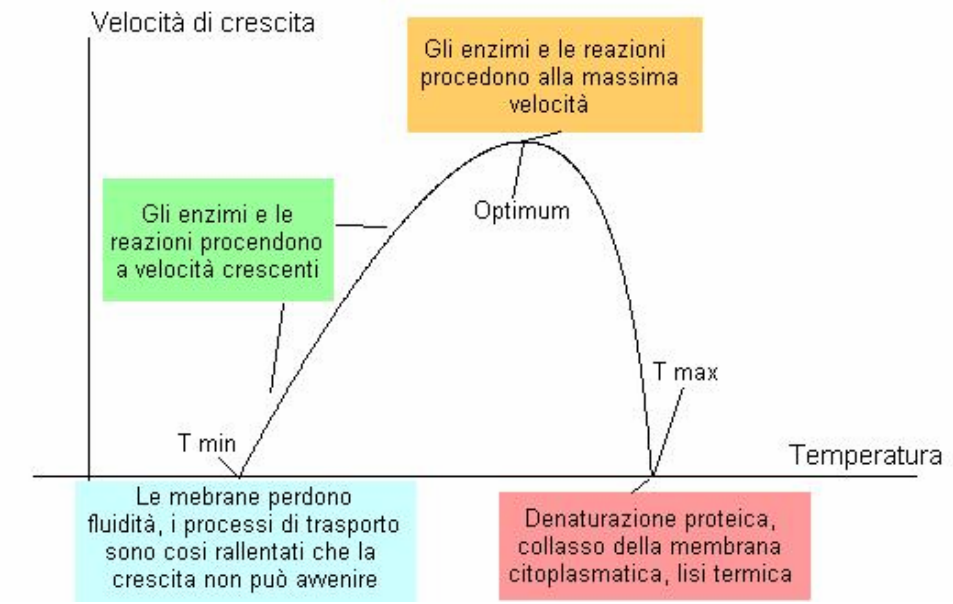


Figura 3.1 Grafico delle temperature di crescita

Per ogni organismo è possibile definire tre temperature di riferimento: T_{min} al disotto della quale non si osserva più la vita; T_{opt} (ottimale) temperatura alla quale la velocità di crescita raggiunge il suo massimo; infine T_{max} al di sopra della quale non solo non c'è più crescita ma si può osservare la morte cellulare. Mentre sembra che la temperatura massima di crescita di un dato organismo dipenda dal punto in cui la cellula comincia ad essere inattivata, i fattori che definiscono la temperatura minima sono meno chiari. É possibile che il minimo di temperatura sia definito dal punto in cui la membrana si irrigidisce e non riesce più a svolgere le sue funzioni di trasporto. La temperatura ottimale è sempre più vicina alla massima che alla minima (Figura 3.1). Studi trasversali alle specie [Ros93] mostrano che questi tre parametri sono altamente correlati, quindi la sola T_{opt} può essere una buona sintesi dell'effetto della temperatura su una specie.

Pur non esistendo una definizione degli ambienti di crescita cellulare, c'è un generale consenso circa alcuni dei suoi più importanti fattori fisici o chimici: "normale" è un habitat in cui la temperatura è compresa tra i 4° e i 40°C, il valore del pH è tra 5 e 8,5 la salinità è compresa tra quella

dell'acqua fresca (< 0,2 M cloruro di sodio) e quella del mare (circa 0,5 M cloruro di sodio).

I procarioti hanno un'organizzazione cellulare fondamentalmente diversa da quella degli eucarioti. Mentre le cellule eucariote hanno un nucleo racchiuso da una membrana e numerosi altri organuli dotati di membrana, le cellule procariote sono prive di queste caratteristiche strutturali; il loro DNA non si trova all'interno del nucleo. La struttura cellulare relativamente semplice dei procarioti è il principale motivo per cui i biologi hanno classificato questi organismi nel regno delle monere.

3.1.1 Habitat degli "estremofili"

É possibile considerare diversi tipi di habitat estremi, a seconda delle caratteristiche di temperature, pH o salinità. Per quanto riguarda il clima, gli habitat caratterizzati da temperature elevate sono generalmente associati a zone con attività sismica.

Habitat terrestri con temperature fino a 100°C sono le solfatare (caratterizzate anche da un ambiente acido per la presenza di solfati) mentre nella profondità dei mari esistono luoghi dove la temperatura può raggiungere i 400° C: le fumarole. Riguardo la salinità ci sono acque come quelle del Mar Morto ed del Gran Lago salato in cui la concentrazione di cloruro di sodio (circa 5,2 M) è cosi elevata da arrivare a saturazione, mentre habitat estremi per il pH sono, oltre le già citate solfatare, i suoli di alcuni deserti e le acque di numerosi laghi, che presentano un altro livello di alcalinità.

I batteri che popolano questi ambienti, a seconda del tipo di habitat, vengono classificati come:

- Termofili (resistenti alle temperature elevate)
- Psicrofili (resistenti alle basse temperature)
- Acidofili (resistenti al pH acido)
- Alcalofili (resistenti a un valore di pH superiore a 10)
- Alofili (resistenti alle elevate concentrazioni saline).

Era il 1972, e proprio quando venne implementato il primo software di posta elettronica, venne scoperto il primo battere ipertermofilo in grado di vivere al di sopra di 80°C; veniva così esteso il limite di temperatura preventivamente fissato per la vita sulla terra [BBBW72]. L'interesse che la scoperta suscitò dal punto di vista scientifico fu notevole. Contemporaneamente si aprivano interessanti prospettive anche nel settore economico: ciò che incuriosiva i ricercatori era una possibile fonte di guadagno per tutte quelle industrie che all'epoca si erano indirizzate verso la produzione di enzimi e che vedevano negli organismi termofili una miniera di catalizzatori molto più stabili e potenzialmente attivi anche alle alte temperature. L interesse scientifico era inoltre duplice, oltre ad essere un interessante oggetto di studio, gli organismi ipertermofili potevano servire come fonte di enzimi stabili per la neoemergente biologia molecolare. L'industria incontra quindi la ricerca: la scoperta di un battere ipertermofilo (Thermus Aquaticus) nel parco naturale di Yellowstone portò alla nascita di una delle più importanti e diffuse tecniche di amplificazione utilizzate in Biologia Molecolare, la PCR, con un giro d'affari iniziale di oltre 300 milioni di dollari americani. Tutto questo avvenne per la scoperta di un solo organismo e di un solo enzima: una DNA polimerasi. È chiaro quindi quanto possa essere vivo ancora oggi l'interesse per questi microrganismi e per le loro proteine. Direttamente è possibile infatti ricavarne nuovi enzimi, indirettamente è possibile studiarne le caratteristiche in confronto alla controparte mesofila per ricavare regole generali sulla termostabilità: trovare la chiave che la vita utilizza per esistere in ambienti così ostili ci consentirebbe di accedere a nuove strategie di termostabilizzazione. Se si conoscessero le regole che la natura utilizza per consentire la vita in condizioni così estreme, si potrebbero termostabilizzare praticamente tutte le proteine mesofile esistenti. Se si considera poi, che il limite attualmente stimato per la stabilità di un enzima non immobilizzato si attesta intorno ai 140°C [Bar91], nel contesto delle capacità catalitiche di queste proteine si possono immaginare moltissimi potenziali usi sia in campo industriale che scientifico. Recenti studi hanno poi dimostrato

completa indipendenza tra capacità catalitica alle basse temperature e stabilità alle alte, aprendo strade di impiego anche in quei processi che non necessariamente richiedano condizioni di lavoro estreme, ma semplicemente necessitano di maggiore longevità dei supporti utilizzati. Gli enzimi isolati dai batteri termofili, a differenza di quelli degli organismi mesofili, non si inattivano a elevate temperature e sono resistenti a solventi e altri agenti denaturanti; enzimi mesofili sono stati soppiantati da quelli termofili. Enzimi termofili sono usati in campo alimentare per la produzione di sciroppi ad alto contenuto di zuccheri o per migliorare la digeribilità di alcuni cibi. Sempre di natura termofila sono particolari proteine (chaperones) che hanno una forte valenza in terapia.

La risposta alla temperatura della funzione di crescita dei microrganismi ha un tipico andamento, comune a tutti, riportato in Figura 3.2. La curva è caratterizzata, durante la fase di crescita esponenziale, dalla specifica velocità di crescita μ della biomassa x:

$$\frac{dx}{dt} = \mu x$$

Equazione 3.1

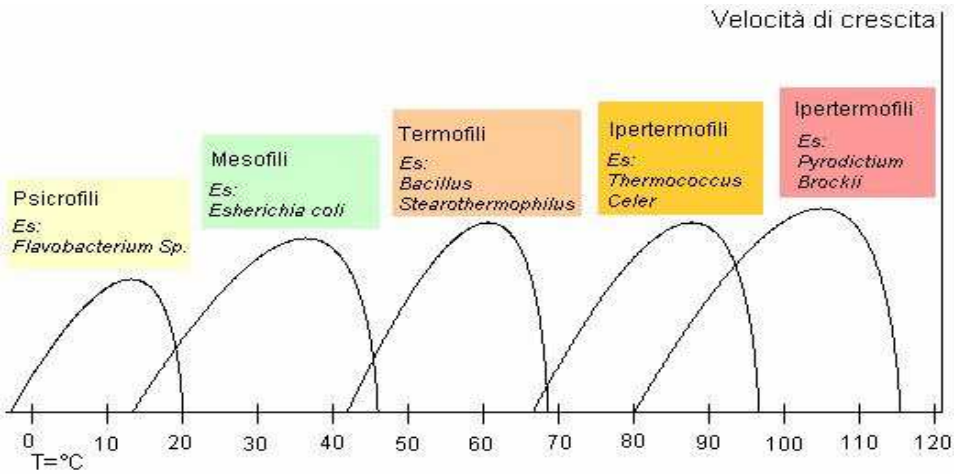


Figura 3.2 Influenza della temperatura nella crescita delle proteine

La specifica velocità di crescita, anche se generalmente caratteristica di ogni microrganismo, non può essere definita in modo assoluto in quanto è funzione di molti altri fattori ambientali: pH, attività dell'acqua, salinità, concentrazione del nutriente.

Tra i microrganismi, in particolare gli archeobatteri sono in grado di sopravvivere e proliferare in condizioni ambientali "estreme". Gli eucarioti invece sono incapaci di adattarsi alle alte temperature, essendo il loro limite fissato a 60-62°C. Il limite per piante e animali è ancora inferiore: meno di 50°C. Sopra i 60-62°C gli unici organismi presenti sono batteri (o procarioti). Alle inusuali temperature superiori ai 100°C troviamo solo particolari organismi adattati al calore del dominio archaea ipertermofili.

I procarioti, nel loro complesso, sono in grado di crescere in tutto l'intervallo di temperature compatibili con la vita; ciononostante nessuno di loro, come nessuno degli eucarioti, è in grado di coprire da solo un intervallo di temperature maggiore di 30°- 40°C. Sebbene l'intervallo da <0° ad oltre 100° gradi C sia coperto senza soluzione di continuità dalle temperature di crescita, è comunque utile suddividere gli organismi in quattro gruppi in base alla loro T_{opt}: psicrofili, mesofili, termofili e ipertermofili. Le tre temperature principali per ciascun gruppo sono riportate in Tabella 3.1.

Organismo	$\mathbf{T_{min}}$	$\mathbf{T_{opt}}$	$\mathbf{T_{max}}$
psicrofili	0°C o inferiore	15°C	sotto i 20°C
mesofili	sotto i 20°C	da 25°C a 40°C	attorno a 45°C
termofili	circa 45°C	da 45°C a 60°C	attorno a 65°C
ipertermofili	da 65°C a 90°C	75°C a 106°C	da 98°C a 113°C

Tabella 3.1 Procarioti e loro temperatura di crescita

Le condizioni che permettono la vita dei microrganismi termofili e ipertermofili sono proprie solo di ambienti limitati come ad esempio am-

bienti associati a fenomeni vulcanici. Le sorgenti termali hanno temperature prossime ai 100°C. Le sorgenti geotermali dei fondali oceanici possono raggiungere temperature superiori ai 350°C. A 2600 metri di profondità l'acqua non bolle fino alla temperatura di 450°C. Man mano che l'acqua si allontana dal punto di affioramento si raffredda gradualmente. Lungo questo gradiente termico possono crescere diverse specie di microrganismi e ogni specie occupa la zona che ha la temperatura adatta alla propria crescita.

Naturalmente una domanda ricorrente e correlata ai problemi evolutivi è quali siano gli adattamenti che una cellula deve subire per resistere in ambienti estremi. Se consideriamo un mesofilo, sono molte le componenti che potrebbero venire danneggiate dall'esposizione ad alte temperature. Fra questi quelle fondamentali: acidi nucleici, DNA e RNA, e proteine. É quindi nell'instabilità di queste biomolecole che vanno cercate le ragioni del limite superiore di temperatura alla vita.

Va comunque considerato che le informazioni strutturali e funzionali note indicano che organismi mesofili e termofili hanno le stesse funzioni e quindi proteine adatte a svolgere funzioni analoghe ma a temperature molto diverse. Allora dove risiede l'origine della termostabilià? Nei prossimi paragrafi verranno descritte alcune delle caratteristiche delle biomolecole fondamentali.

3.2 Stabilità delle proteine: il problema del folding

Le proteine si trovano in ambiente fisiologico nello stato cosiddetto nativo, stabile e funzionale.

Una proteina nello stato nativo ha una stabilità termodinamica relativamente bassa: per denaturarla sono richiesti appena 0,4 kJ*mole-1 circa per residuo. Lo stato nativo è il risultato dell'equilibrio di interazioni co-

valenti e non covalenti e di interazioni solvente-soluto che concorrono, agendo con spinte opposte, a stabilizzarne la struttura.

Da questo punto di vista, il complesso del sistema della proteina e delle sue interazioni può essere ricondotto a quello di un sistema termodinamico dove la funzione potenziale è il risultato delle interazioni stesse e dove lo stato di minima energia individua lo stato nativo.

Il complesso dei processi auto organizzativi che portano la proteina allo stato nativo viene detto folding. La comprensione dei meccanismi secondo cui questi processi avvengono costituisce il problema del folding [FraWol94]. La teoria del folding trova il suo fondamento negli studi di Anfinsen [Anf61] [Anf73], da cui risulta che la struttura tridimensionale di una proteina dipende unicamente dalla sua struttura primaria la quale ne caratterizza le interazioni tra i suoi residui e tra questi e il solvente.

Il processo di folding implica una serie di interazioni a livello atomico e molecolare che stabilizzano la stessa struttura tridimensionale della proteina. Queste interazioni possono essere di tipo non covalente o di tipo covalente, fra quelle riportate nel paragrafo precedente, a cui si deve aggiungere il contributo apportato dal cosiddetto effetto idrofobico che coinvolge l'ambiente biologico in cui si trova la proteina tramite interazioni residui-solvente.

3.2.1 Termostabilità delle proteine

Il termine termostabilità proteica si riferisce alla preservazione di una struttura chimica e spaziale unica per una catena polipeptidica sotto condizioni di temperature estreme: in altre parole indica la conservazione di un'unica e sola conformazione. La comparazione delle sequenze e delle strutture delle proteine degli estremofili con la corrispondente controparte derivante da organismi mesofili dovrebbe portarci alla comprensione del meccanismo che la natura ha impiegato per accrescere la stabilità delle proteine.

4 IL PDB

PDB è l'acronimo di *Protein Data Bank* ovvero la banca dati delle proteine risolte. Contiene informazioni sulle strutture di proteine, acidi nucleici e complessi (proteina/proteina, proteina/acido nucleico). Tutti i dati sono ricavati sperimentalmente per mezzo di tecniche quali la Cristallografia a raggi X e la Risonanza Magnetica Nucleare. Attualmente sono presenti circa 27.000 entry, in continuo aggiornamento.

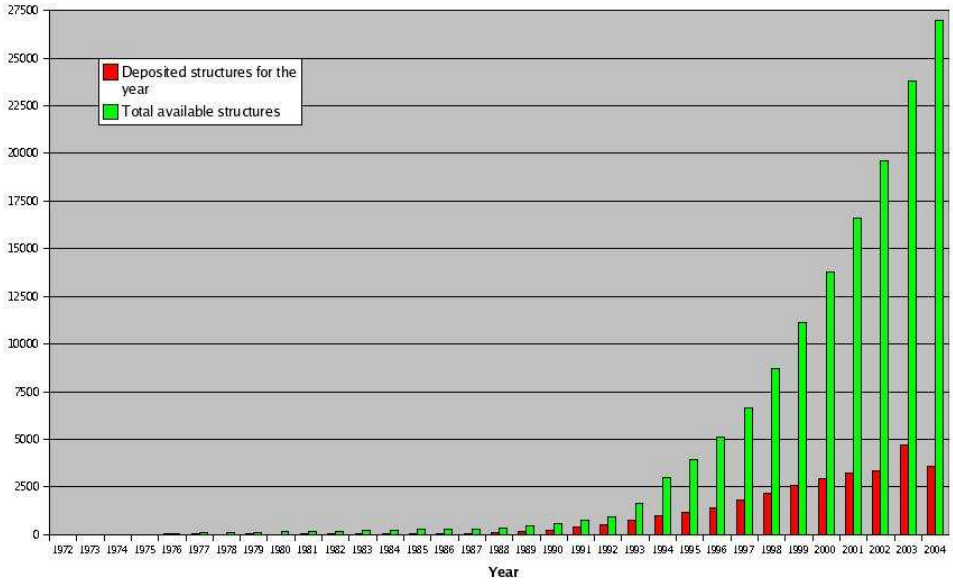


Figura 4.1 Grafico della crescita del PDB dal 70 ad oggi [RCSB04]

Le informazioni riguardanti ogni singola molecola sono raccolte in file (*Atomic Coordinate Entry Format Descriptior*, comunemente chiamato file PDB, ad indicarne il formato). Ogni file contiene le coordinate cartesiane che contraddistinguono in modo univoco la posizione di ogni atomo della proteina. L'informazione è completata dalle citazioni bibliografiche, struttura primaria e secondaria, funzione, organismo da cui deri-

va la proteina, la tecnica utilizzata per caratterizzare la struttura terziaria ed eventualmente la risoluzione e le date di rilascio e revisione del file.

Ai fini del progetto sarà tuttavia sufficiente analizzare la sezione contenente le informazioni descrittive, la *Title Section* e quella contenente le informazioni sulla struttura primaria, la *Primary Structure Section*.

4.1 Formato

Ogni file PDB è composto da una serie di campi distribuiti su una o più righe, ciascuna delle quali è di lunghezza fissa di 80 caratteri (eventualmente spazi vuoti) ed interrotta da un carattere di fine riga (*Line feed*, `&h0a`).

Caratteri	Lunghezza	Descrizione
1 - 6	6	Nome del campo
7 - 8	2	Riservati
9 - 10	2	Numero di riga, per campi estesi su più righe
11 - 70	60	Contenuto del campo. Può contenere a sua volta dei sottocampi
71 - 80	10	Eventualmente contiene informazioni sulla revisione
81	1	Fisso, carattere ASCII numero 10 (line feed)

Tabella 4.1 Struttura dei file PDB

I primi sei caratteri di ogni riga indicano il nome del campo, allineato a sinistra, ed i dati relativi al campo si estendono fino alla colonna 70, lasciando di fatto 10 caratteri vuoti eventualmente utilizzati per indicare successive modifiche al documento. Nel caso le informazioni di un campo siano distribuite su più righe, queste vengono identificare da un intero

progressivamente crescente. In ogni caso, le informazioni relative ad un campo terminano sempre con un carattere di punto e virgola (`&h3b`).

4.1.1 Title Section

In questa sezione sono presenti tutte le informazioni necessarie a descrivere la proteina e tutte le macromolecole che la compongono. In Tabella 4.2 vengono indicati i contenuti dei campi presi in considerazione dall'applicazione.

Campo	**Contenuti**
HEADER	Definisce univocamente la proteina tramite l'idCode. Contiene informazioni sulla classificazione e sulla data di deposito della proteina
TITLE	Contiene il nome dell'esperimento o dell'analisi che rappresenta la proteina
COMPND	Informazioni sul contenuto macromolecolare della proteina
SOURCE	Sorgente biologica o chimica di ogni molecola biologica definita in COMPND
KEYWDS	Insieme di termini correlati all'esperimento, utili per categorizzare la proteina e per facilitarne l'individuazione
EXPDATA	Tecnica sperimentale utilizzata
AUTHOR	Insieme di ricercatori responsabili dei contenuti nel PDB
REVDAT	Storia delle modifiche dell'entità
JRNL	Informazioni sulle citazioni originali a proposito della proteina
REMARK	Informazioni accessorie. Utilizzato per determinare la risoluzione in Angstrom

Tabella 4.2 Campi e contenuti della Title Section

Alcuni campi non rispettano il formato generico descritto al paragrafo 4.1. Per quanto riguarda HEADER, a differenza degli altri campi sappiamo che esso è composto sempre e solo da una sola riga, la prima del file, e le informazioni sono localizzate in posizioni precise.

COMPND e SOURCE, correlati tra loro, contengono degli insiemi di coppie *chiave: valore* che riportano informazioni sulle molecole che compongono la proteina.

REVDAT può essere espresso su più righe, ognuna delle quali contiene informazioni su una differente revisione del file e in REMARK possono essere presenti i dati più disparati (dettagli sull'esperimento, commenti, annotazioni e più in generale tutto quello che non è contenuto in altri campi) alcuni dei quali vengono identificati da parole chiave; in ogni caso possono essere estratti solo a seguito di una ricerca all'interno delle varie righe che compongono il campo.

4.1.2 Primary Structure Section

La sezione riporta tutti i dati sulla sequenza di residui che compongono ogni catena di residui della molecola. Racchiuse in queste informazioni possiamo trovare gli identificativi delle catene ed i numeri di sequenza usati da altri campi come riferimento, come ad esempio in COMPND. Di questa sezione viene preso in considerazione il campo SEQRES, dal quale possiamo ricavare il numero di residui di tutte le catene identificate nel campo COMPND.

4.1.3 Altri campi

Infine vengono considerati altri due campi per ottenere informazioni accessorie: FORMUL per gli atomi di acqua e SSBOND per i legami disulfidici. Nel primo caso è necessario individuare le righe contenenti un asterisco alla colonna 19 e leggere il numero immediatamente seguente, men-

tre per i legami disulfidici è sufficiente trovare almeno una riga che ne descriva: è infatti di interesse la sola presenza e non il valore.

4.2 FSSP

Il Protein Data Bank ha un ritmo di crescita vertiginoso, il numero di strutture raddoppia approssimativamente in un tempo di 18 mesi.

Questa crescita, con andamento quasi esponenziale, porta alla necessità di implementare un metodo automatico al fine di organizzare in modo adeguato l'elevata mole di dati. Questo è il motivo principale per cui nel 1992 è stato istituito il database FSSP (*Fold classification based on Structure-Structure alignment of Proteins*) al quale è stato affiancato nel 1998 il *Dali Domain Dictionary* [Hol01], entrambi volti a processare le nuove strutture 3D continuamente rilasciate dal PDB. Il server FSSP/DALI (Figura 4.2) svolge una classificazione strutturale in modo completamente automatizzato, senza cioè l'apporto di ulteriori risorse umane, come avviene invece, ad esempio, per il database delle strutture SCOP di Murzin [Mur03]. Un'altra motivazione dello sviluppo di FSSP/DALI è quello di evitare le ridondanze; il PDB è infatti una banca dati ridondante in termini di sequenze e similarità strutturale.

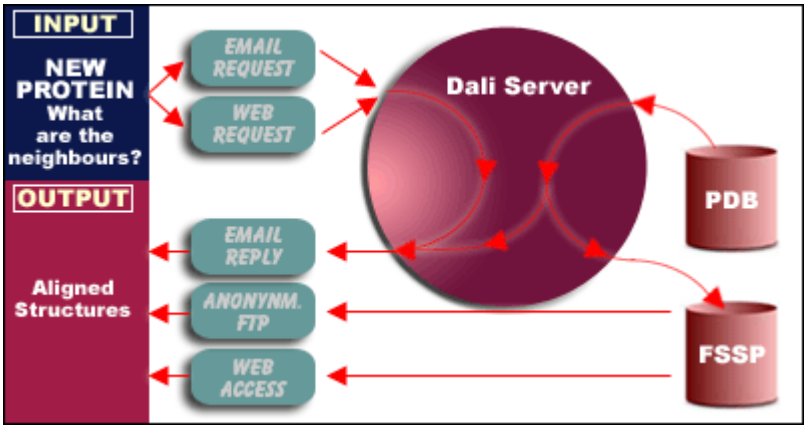


Figura 4.2 Struttura del Dali Server [Hol95]

Lo scopo che si prefissa questo server è quello di ottenere una descrizione completa e al contempo poco onerosa in termini di tempo e denaro dei dati strutturali attualmente disponibili. L'utente ha la possibilità di richiedere (tramite e-mail o direttamente via web) se esistono e quali sono le proteine vicine alla proteina di suo interesse. DALI interroga il PDB, l'FSSP ed infine ritorna (ancora una volta utilizzando l'e-mail o il web a seconda della scelta dell'utente) le eventuali proteine simili come allineamento multiplo. Non è una regola ma la comparazione di strutture tridimensionali rispetto alla comparazione delle relative sequenze può rivelare similarità di un certo interesse biologico, nonostante le due sequenze non presentino una identità di sequenza ragionevolmente elevata.

5 TOPDB

TOPDB (*Data Base of Proteins from Thermophilic Organisms*) è la banca dati delle proteine tratte da organismi termofili derivanti dal Protein Data Bank. I vantaggi di TOPDB rispetto al PDB sono molteplici. Anzitutto esiste una precisa classificazione tra proteine termofile e mesofile, alla base degli studi condotti dal *Biocomputing Group of the University of Bologna*, inoltre grazie a TOPDB è possibile effettuare una ricerca mirata tramite il database FSSP ed ottenere i risultati distinti in omologhi termofili e mesofili ad una certa percentuale di identità di sequenza. In realtà TOPDB non si collega all'FSSP/DALI perché le omologie sono precalcolate. Ciò avviene durante l'aggiornamento, quando viene trovato un nuovo cristallo proveniente da organismo termofilo che non sia già compreso in TOPDB. In questo caso infatti si ricerca nuovamente sul server FSSP/DALI e si salvano i risultati sul database locale.

È stato scelto di rendere indipendente il TOPDB dagli altri database (aggiornamenti esclusi) per evitare possibili problemi legati a cadute dei server o lavori di mantenimento, che impedirebbero il collegamento e quindi la possibilità di reperire le informazioni ricercate, con conseguente perdita di tempo.

La prima versione di TOPDB è il frutto di un precedente lavoro di tesi di laurea triennale, realizzato con *Microsoft® Active Server Page* (ASP), un ambiente di scripting lato server utilizzato per creare applicazioni web dinamiche ed interattive [Mic04]. Il linguaggio di scripting utilizzato è *Microsoft® Visual Basic® Scripting Edition* (VBScript) [Mic04b].

Questo progetto prevede l'analisi ed il consolidamento delle funzionalità già presenti nel programma e l'aggiunta di quelle necessarie ad aumentarne l'usabilità ed estenderne l'accesso ad un vasto numero di utenti tramite la pubblicazione in internet.

Gli obiettivi sono stati raggiunti sviluppando i punti seguenti:

- Reingegnerizzazione
  - Aggregazione delle funzioni e procedure di pubblica utilità in moduli di inclusione esterni;
  - Gerarchizzazione dei file in sottocartelle in base al tipo;
  - Revisione delle procedure;
  - Revisione dell'output in conformità allo standard XHTML;
  - Impiego dei CSS per la definizione dell'aspetto.
- Accesso pubblico
  - Revisione del sistema di accesso;
  - Sistema di iscrizione automatico e moderato;
  - Gestione dell'account personale.
- Gestione degli account
  - Moderazione degli account;
  - Gestione dei privilegi.
- Gestione degli aggiornamenti
  - Analisi della procedura esistente;
  - Analisi del sistema ciclico;
  - Aggiornamento delle proteine;
  - Analisi dei file PDB;
  - Modello OO delle proteine;
  - Aggiornamento degli FSSP;
  - Gestione della lista di organismi termofili.
- Altre modifiche
  - Aggiornamento singola proteina;
  - Statistiche database;
  - Configurazione sistema.

5.1 Reingegnerizzazione

Il primo passo, necessario per agevolare i successivi, è stato la revisione e, in buona parte, la ricodifica del programma in modo da rendere il codice più omogeneo, modulare e riutilizzabile, aggregando ad esempio le funzioni utilizzate in più pagine `asp` in un'unica pagina inclusa di volta in volta nelle altre. Questa fase non apporta modifiche all'aspetto e non aggiunge funzionalità al programma; tuttavia le numerose ottimizzazioni semplificheranno le fasi successive, evitando ad esempio la scrittura di funzioni già esistenti, come la connessione al database, oppure eliminando il problema della resa grafica, definita una volta per tutte nei fogli di stile. In definitiva il tempo speso in questa fase è stato un proficuo investimento per quelle successive.

5.1.1 Modularizzazione

Uno dei concetti base dell'ingegneria del software è il riutilizzo del codice sorgente, esigenza favorita dagli stessi linguaggi di programmazione, grazie tra l'altro ad apposite sintassi che permettono l'inclusione di moduli esterni.

Nonostante le pagine restituite da TOPDB possano avere anche sostanziali differenze, esistono alcune procedure da eseguire in tutti i casi, come ad esempio la connessione alla fonte di dati, il caricamento di costanti globali, ecc., senza considerare l'intestazione ed il piede pagina, praticamente identici in tutte le pagine.

Oltre a questo, esistono molte funzioni e procedure di utilità generica, come la gestione delle stringhe, la validazione dei dati, ecc. che anche se attualmente vengono utilizzate solo da TOPDB potrebbero essere riutilizzate da altre applicazioni, con conseguente risparmio di risorse.

A seguito della suddivisione in moduli è possibile definire una generica struttura delle pagine dell'applicazione come quella riportata in Figura 5.1.

```
Inclusione modulo di configurazione

Eventuale inclusione moduli con funzioni e
    procedure necessarie

Inclusione intestazione pagina

Codice specifico pagina
    ...
    ...
    ...
Fine codice specifico pagina

Inclusione piede pagina
```

Figura 5.1 Struttura generica di una pagina dell'applicazione

In Tabella 5.1 è presente la lista e la descrizione di tutti i moduli di inclusione definiti per questo progetto. I moduli `function.asp` e `mail.asp` definiscono funzioni e procedure la cui utilità non è limitata agli scopi di questo programma, e pertanto possono essere riutilizzati in altre applicazioni.

`config.asp`	Caricamento della configurazione, definizione delle costanti, apertura della connessione alla fonte dati
`function.asp`	Generiche funzioni non standard per la validazione e la conversione dei dati
`mail.asp`	Gestione invio e-mail
`header.asp`	Intestazione generica
`listheader.asp`	Intestazione per le pagine di tipo lista (risultati interrogazioni)
`plist.asp`	Funzioni per la generazione della lista di risultati
`listfooter.asp`	Piede pagina per le pagine di tipo lista
`footer.asp`	Piede di pagina generico
`update.asp`	Funzioni per la gestione degli aggiornamenti del database

Tabella 5.1 Moduli di inclusione

Grazie alla suddivisione in moduli di inclusione è stato inoltre possibile creare un'astrazione della base di dati. La connessione e la comunicazione con la fonte di dati vengono gestite infatti dal modulo di configurazione, e cambiando i parametri in tale modulo è possibile utilizzare altri tipi di database, purché compatibili con lo standard SQL. A testimonianza di questo è il fatto che, nonostante lo sviluppo sia avvenuto principalmente con database di tipo *Access*, il prodotto finito funzioni correttamente anche con database di tipo *MSSQL*.

5.1.2 Gerarchizzazione dei file

Per migliorare la chiarezza del progetto sono state apportate modifiche anche dal punto di vista dell'organizzazione dei vari file.

Da un'originale disposizione lineare, ove tutti i file risiedevano nella *root* (radice), essi ora sono stati suddivisi in base al loro tipo ed alla loro funzione, come mostrato in Figura 5.2.

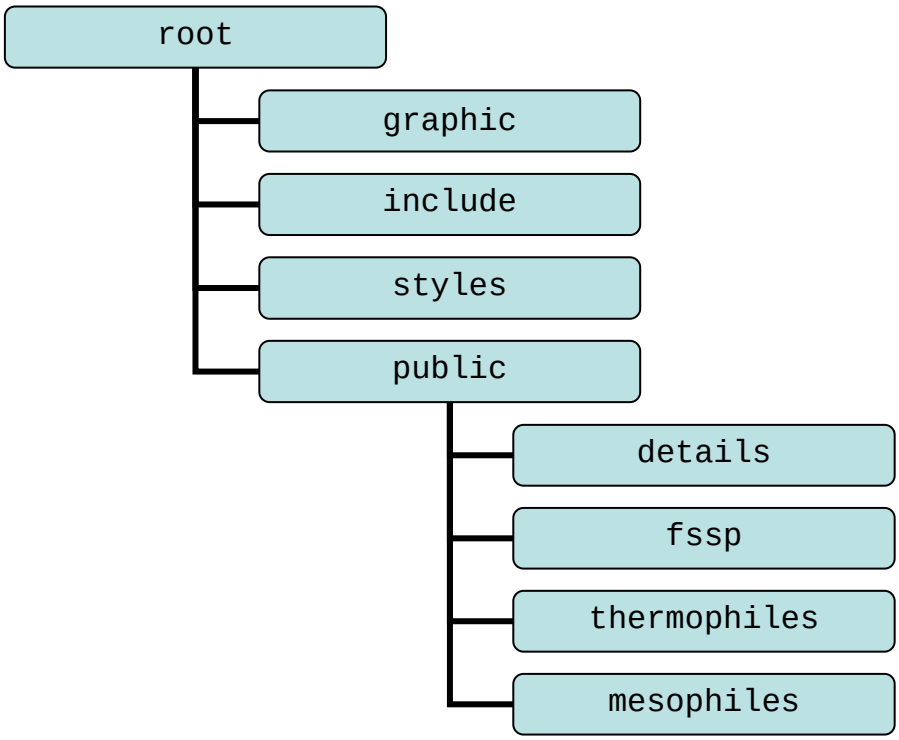


Figura 5.2 Albero delle directory

Il percorso di accesso ad ognuna delle sottocartelle (*path*) è stato parametrizzato sotto forma di costanti globali, in modo da poter facilmente rinominare o riorganizzare la struttura presentata in situazioni di test o in vista di nuove installazioni.

In Tabella 5.2 vengono descritti i tipi di documenti contenuti in ognuna delle sottocartelle.

`root`	Pagine navigabili del progetto, sia statiche (*html*) che dinamiche (*asp*)
`graphic`	Immagini, icone e figure utilizzate
`include`	Contiene i moduli di inclusione, descritti nel paragrafo 5.1.1
`styles`	Fogli di stile per la resa grafica delle pagine. Attualmente è presente solo lo stile definito per il media `screen`
`public`	Cartella aperta in scrittura. Qui e nelle sottocartelle vengono salvati file generati dall'applicazione
`details`	Dettagli delle catene polimeriche scaricate
`fssp`	FSSP delle proteine termofile
`thermophiles`	PDB delle proteine termofile scaricate
`mesophiles`	PDB delle proteine mesofile scaricate

Tabella 5.2 Funzioni delle directory

5.1.3 Revisione delle procedure

Durante la fase di riorganizzazione e distribuzione delle procedure generiche in moduli di inclusione è stata apportata una revisione ed ottimizzazione di alcune funzioni.

Le più importanti riguardano la ricerca (*Search*) e la ricerca avanzata (*Advanced Search*). Le due procedure, viste le notevoli somiglianze, sono state fuse in un unico modulo che opera in modalità semplice o avanzata in base al tipo di input ricevuto.

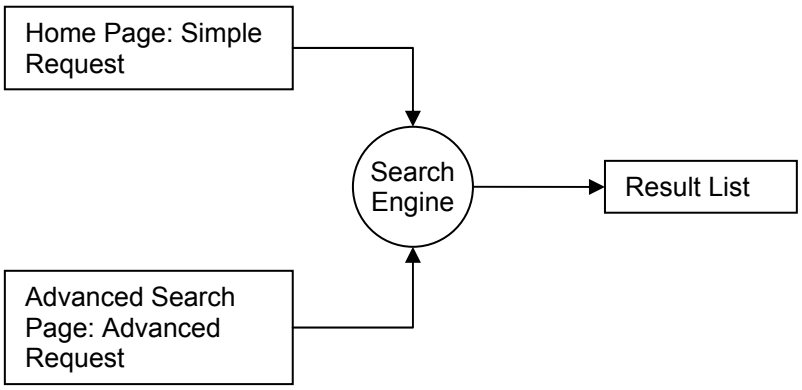


Figura 5.3 Ricerca e Ricerca Avanzata

La richiesta di ricerca può essere formulata dall'home page (`default.asp`) in caso di ricerca semplice, oppure dalla pagina della ricerca avanzata (`advsearch.asp`). In entrambe i casi essa, che contenente tutti i parametri di ricerca, viene inoltrata alla pagina predisposta alla restituzione di risultati (`resultlist.asp`), che la elabora e genera la lista delle proteine che la soddisfano.

Concentrando l'elaborazione di due tipi di richiesta in un unico modulo è stato possibile eliminare la ridondanza della gestione degli errori e della generazione della lista dei risultati, sostanzialmente simile nei due casi, al prezzo di un parametro extra nella richiesta di ricerca (`SearchType`) che ne specifica il tipo. Questa scelta garantisce inoltre la coerenza nella presentazione dei risultati della ricerca, giacché generati dallo stesso codice.

Sempre per questioni di coerenza, anche la generazione della parte di HTML che rappresenta il *Compound* (composizione) della proteina è stata revisionata e spostata nel modulo `plist.asp` (vedi Tabella 5.1) per poter essere utilizzata sia nella generazione della lista di risultati di una ricerca (semplice o avanzata), che nella visualizzazione dei dettagli della proteina (`visualizzapdb.asp`).

Infine, l'ultima revisione di rilevante entità riguarda la pagina degli *FSSP*, ossia la lista delle proteine simili da un punto di vista strutturale ad

una proteina data. In questo caso è stato completamente ricodificato il motore di ricerca preposto all'individuazione di proteine simili.

Nella tabella degli FSSP risiedono i record che descrivono le similitudini tra due proteine e, dato che le proteine prese in considerazione dal programma sono quelle termofile, le uniche coppie presenti sono quelle che legano proteine termofile a proteine mesofile o termofile ad altre termofile. Ne consegue che, per ogni proteina termofila, nella tabella degli FSSP siano presenti da zero ad *n* record con descrizioni sulle similitudini tra le strutture.

Tuttavia, nella vecchia versione, nel caso di due termofile simili esistevano due record con le stesse informazioni: il primo che legava la prima proteina alla seconda, ed il secondo che legava la seconda alla prima, dal momento che le similitudini tra due proteine prese in considerazione dall'FSSP godono della proprietà riflessiva. Ciò dava origine ad un caso di ridondanza dell'informazione, come si può notare dalla Tabella 5.3 dove, ad esempio, le righe 1 e 3 riportano gli stessi dettagli.

Riga	Proteina	Prot. Simile	Dettagli
1	A (thermo)	B (thermo)	Dettagli similitudini A-B
2	A (thermo)	C (meso)	Dettagli similitudini A-C
3	B (thermo)	A (thermo)	Dettagli similitudini A-B
4	B (thermo)	D (meso)	Dettagli similitudini B-D

Tabella 5.3 Esempio di tabella degli FSSP nella vecchia versione

Le ridondanze non sono sempre inutili [ACPT99] e difatti l'approccio della vecchia versione andava a vantaggio dell'algoritmo di ricerca: era sufficiente individuare tutti i record in cui il campo `Proteina` coincidesse con il codice della proteina nota.

É stato tuttavia possibile ottenere gli stessi risultati della vecchia versione eliminando tutte le ridondanze e senza alterare la struttura del database al prezzo di un minimo aumento dei costi computazionali. Semplicemente, il nuovo algoritmo cerca il codice della proteina data sia nel campo `Proteina 1` (`Proteina`) che nel campo `Proteina 2` (`Prot. Simile`). In Tabella 5.4 è possibile avere un esempio dei nuovi

contenuti della tabella FSSP in grado di definire gli stessi dati presenti in Tabella 5.3.

Riga	Proteina 1	Proteina 2	Dettagli
1	A (thermo)	B (thermo)	Dettagli similitudini A-B
2	A (thermo)	C (meso)	Dettagli similitudini A-C
3	B (thermo)	D (meso)	Dettagli similitudini B-D

Tabella 5.4 Esempio di tabella degli FSSP nella nuova versione

Intuitivamente il risparmio di record duplicati è direttamente proporzionale al numero di proteine termofile presenti nel database. Idealmente, se il database contenesse solo proteine di questo tipo, il risparmio sarebbe del 50%, ed i valori reali non dovrebbero discostarsi grandemente da questo indice in quanto le proteine contenute nel database sono termofile nella maggioranza dei casi; un semplice comando SQL (Figura 5.4) ci permette di ottenere il numero esatto di record duplicati e quindi di determinare informazioni più dettagliate sull'entità del risparmio: su circa 2850 record nella tabella degli FSSP presenti nel vecchio formato circa il 44% rappresentavano informazioni ridondanti, che grazie al nuovo algoritmo di ricerca sono divenute inutili e quindi sono state eliminate.

```
SELECT Count(*) as Redoundance
FROM Fssp INNER JOIN Fssp AS ReFssp ON
    (ReFssp.repre = Fssp.pdb_id) AND
    (Fssp.pdb_chain = ReFssp.repre_chain) AND
    (Fssp.repre_chain = ReFssp.pdb_chain) AND
    (Fssp.repre = ReFssp.pdb_id)
WHERE (Fssp.pdb_id <> Fssp.repre) OR
    (Fssp.pdb_chain <> Fssp.repre_chain);
```

Figura 5.4 Query per la determinazione del numero di ridondanze

5.1.4 Formato delle pagine generate

Il sistema migliore e più largamente impiegato per rendere disponibile un'applicazione al pubblico è renderla fruibile via *World Wide Web*. Normalmente ciò avviene grazie all'impiego di particolari linguaggi ed architetture in grado di formulare risposte sotto forma di pagine *HTML*

inviate agli utenti che ne facciano richiesta. Tali pagine vengono elaborate e visualizzate in locale, genericamente con *browser web*. Nonostante questi programmi siano ampiamente diffusi, la loro realizzazione viene effettuata da diverse software house e per le più disparate ragioni possono presentare numerose incompatibilità per quanto riguarda la resa grafica.

A questo scopo il *World Wide Web Consortium* (W3C) definisce standard universalmente riconosciuti e rispettati. Uno di questi è l'*XHTML* [Pem02], che pone alcune rigide restrizioni all'HTML in modo da renderlo a tutti gli effetti un documento *XML,* estendendone la compatibilità. I documenti XML possono infatti essere utilizzati come input (direttamente o dopo essere state filtrate da un XSLT che ne possono alterare la struttura) di altre applicazioni (non solo da browser, quindi) e facilmente elaborate per estrarne informazioni o per alterarne la resa grafica.

Si è quindi optato per cambiare il formato delle pagine generate dal programma in modo da renderlo conforme a questo standard.

Esistono tre differenti versioni di XHTML:

- *Strict*, la più rigida;
- *Transitional*, la più compatibile al semplice HTML;
- *Frameset*, per definire pagine divise in finestre.

Sebbene le restrizioni rispettate siano per la maggior parte fedeli all'XHTML Strict, per mantenere un'assoluta coerenza con la vecchia versione è stato usato l'XHTML Transitional a causa dei numerosi attributi `target="_blank"` presenti nei link dell'applicazione, attributi deprecati in XHTML Strict.

Tutte le pagine hanno superato un controllo effettuato tramite il *Markup Validation Service* (http://validator.w3.org/), un servizio del W3C che certifica la qualità dei documenti.

5.1.5 Scissione tra contenuti e presentazione

Secondo le direttive del W3C, è di fondamentale importanza attuare una netta scissione tra i contenuti di un documento ed il suo aspetto

[Lil03], tanto che il nuovo formato adottato, l'XHTML, non prevede l'utilizzo degli attributi HTML necessari a definire la resa grafica di una pagina. Tuttavia, grazie ai *Cascading Style Sheet* (CSS) [HåkBer99], è possibile definire l'aspetto di una pagina senza intervenire su di essa.

I CSS, o fogli di stile, permettono infatti di definire l'aspetto di ogni elemento semantico di un documento (titolo, paragrafo, figura, ecc.) "dall'esterno", a differenza del vecchio approccio dove ogni elemento era tenuto a definire il proprio aspetto per via di attributi e per lo stesso tipo di elemento era necessario riformulare tali definizioni.

I fogli di stile consistono in uno o più file contenenti le definizioni citate. Il file viene incluso una volta per tutte dal documento XHTML, ed i browser in grado di supportare i fogli di stile automaticamente formattano il documento in base alle direttive del CSS.

Anche in questo caso i vantaggi sono numerosi, e ne verranno citati alcuni. Anzitutto la coerenza: lo stesso foglio di stile può essere utilizzato da più documenti, i quali ovviamente verranno formattati in maniera simile. Inoltre, in caso risulti necessario modificare l'aspetto generale dei documenti, sarà sufficiente modificare il solo foglio di stile, e non tutte le pagine XHTML. É altresì possibile generare fogli di stile in grado di rendere i documenti compatibili con altri dispositivi, come le stampanti o dispositivi di sintesi vocale per garantire l'accessibilità ai non vedenti.

Ai fini dell'applicazione e per mantenere coerenza con la vecchia versione, è stato creato un solo foglio di stile che ne riprende l'aspetto.

5.2 Accesso pubblico

Per consentire l'accesso all'applicazione da utenti generici è stato necessario implementare un sistema di iscrizione ed una rigida gestione dei privilegi, in modo da monitorare gli accessi e ridurre il rischio di errore da parte degli utenti inesperti.

La gestione dei privilegi è riservata agli amministratori, e verrà descritta nel paragrafo 5.3; qui saranno analizzate le scelte effettuate e le procedure implementate per la gestione delle iscrizioni e per la modifica dei dati personali.

5.2.1 Revisione del sistema di login

Data la criticità dell'operazione, la procedura di accesso è stata perfezionata aggiungendo ulteriori controlli sui dati inviati nella richiesta di autenticazione. Inoltre, per una corretta gestione delle iscrizioni, è stato necessario arricchire la struttura dati della tabella contenente le informazioni sugli account con i campi descritti in Tabella 5.5.

`IsActive`	Booleano che definisce lo stato dell'account (attivo/sospeso)
`EMail`	Stringa contenente l'indirizzo di posta elettronica dell'iscritto
`Name`, `Occupation`, `Organization`	Dati informativi supplementari

Tabella 5.5 Nuovi campi della tabella Utenti

```
ALTER TABLE Utenti ADD COLUMN IsActive BYTE
    DEFAULT 1 Not Null;
ALTER TABLE Utenti ADD COLUMN EMail TEXT (255)
    UNIQUE;
ALTER TABLE Utenti ADD COLUMN Name TEXT (255);
ALTER TABLE Utenti ADD COLUMN Occupation TEXT
    (255);
ALTER TABLE Utenti ADD COLUMN Organization TEXT
    (255);
```

Figura 5.5 Comandi SQL per l'aggiunta dei nuovi campi nella tabella Utenti

5.2.2 Sistema di iscrizione

É possibile ottenere un nome utente ed una password di accesso solo su richiesta di iscrizione. Essa viene gestita dal programma e può essere di due tipi, automatica o moderata. Il tipo di procedura utilizzato dipende dalla configurazione (paragrafo 5.5.3).

L'iscrizione automatica garantisce l'accesso al programma in modo immediato. Un'e-mail contenente la password di accesso (generata casualmente) viene inviata all'indirizzo specificato in fase di registrazione e, assieme al nome utente scelto, permette di accedere all'applicazione.

L'iscrizione moderata si effettua invece attraverso due fasi. Nella prima fase l'utente compila una richiesta di iscrizione a seguito della quale un'e-mail viene inviata all'utente a scopo informativo ed un'altra viene inviata agli amministratori. Una volta presa in considerazione la richiesta, gli amministratori hanno la facoltà di confermarla (come descritto nel paragrafo 5.3.1), nel qual caso l'utente viene avvisato, sempre tramite e-mail, ed ottiene l'accesso all'applicazione.

5.2.3 Gestione dell'account personale

Una volta attivato (in modo automatico o dagli amministratori), l'account può essere usato dal proprietario per utilizzare l'applicazione. É inoltre possibile accedere ad un'area che permette di modificare gli estremi dell'account (ad eccezione del nome utente). Tuttavia, se la modalità dell'iscrizione è moderata, ogni modifica disattiva l'account ed opportune mail vengono inviate sia all'utente che all'amministratore, allo stesso modo di quanto accade nel caso di nuove iscrizioni. Questa misura preventiva permette agli amministratori un controllo efficiente degli account in quanto sollecitati dal programma a controllare tutte le modifiche.

5.3 Gestione degli account

Una volta fornita agli utenti la possibilità di effettuare un'iscrizione automatizzata, sorge la necessità da parte degli amministratori e dei moderatori di poter monitorare gli account creati. Le opzioni disponibili nell'area di gestione degli account permettono di controllare, modificare o annullare tutte le iscrizioni e quindi di individuare account errati o maliziosi.

É tuttavia necessario definire una gerarchia tra i vari livelli di utenza.

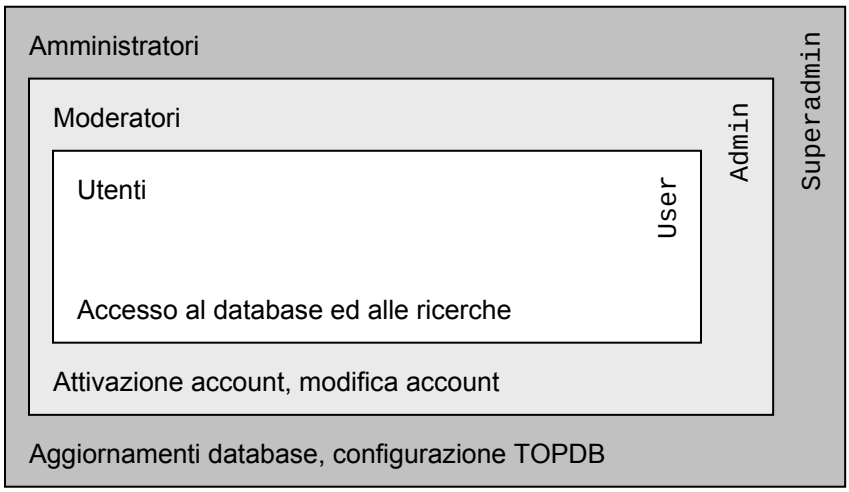


Figura 5.6 Livelli di utenza e relativi privilegi

Esistono tre livelli di utenze: gli utenti semplici, i moderatori e gli amministratori, in ordine gerarchico. I primi hanno i privilegi minori e possono solamente effettuare interrogazioni sul database per mezzo della ricerca e ricerca avanzata.

I moderatori, oltre ai privilegi degli utenti semplici, possono accedere all'area di gestione degli account e, oltre ad attivare o disattivare un account, possono apportare modifiche ad essi oppure cancellarli definitivamente.

Infine gli amministratori hanno accesso a tutte le aree dell'applicazione, compresa la gestione della configurazione e degli aggiornamenti al database.

5.3.1 Moderazione degli account

Accessibile solo da moderatori ed amministratori, quest'area permette di moderare rapidamente gli account.

Data una lista di tutti gli iscritti, è possibile individuare velocemente gli account attivi (con i dettagli in nero) da quelli inattivi (con i dettagli in grigio e in corsivo). Con un semplice comando è possibile disattivare un qualsiasi account attivo, o scegliere se attivare o cancellare un account inattivo.

Icona	Azione	Descrizione
	Dettagli	Mostra i dettagli dell'account
	Attiva	Attiva un account non attivo
	Disattiva	Disattiva un account attivo
	Elimina	Elimina un account non attivo

Tabella 5.6 Comandi per la moderazione degli account

Sono state definite due regole necessarie a garantire la funzionalità della suddivisione in gerarchie. La prima prevede che sia possibile agire solo su utenze pari o inferiori al proprio. Ne consegue un'importante invariante:

$$n_{adm} \geq 1$$

Invariante 5.1

dove n_{adm} è il numero di amministratori; senza amministratori non è possibile crearne altri e quindi non sarebbe più possibile accedere ad alcune funzioni dell'applicazione riservate ad essi.

La seconda regola impone che sia impossibile disattivare il proprio account. Queste due semplici regole garantirebbero il rispetto dell'Invariante 5.1, dato che gli amministratori possono essere disattivati solo da altri amministratori ed un amministratore non può disattivare sé stesso; tuttavia per l'effetto *race condition*, è possibile (seppur remotamente) che due amministratori da postazioni differenti per errore disattivino reciprocamente il loro account. Per evitare questo rischio è stata posta un'ulteriore definitiva condizione: non è possibile eliminare l'ultimo amministratore.

5.3.2 Gestione degli account e dei privilegi

Dall'area di moderazione degli account (vedi par. 5.3.1) è possibile richiedere il dettaglio di un account per modificarne gli estremi, sempre che il livello sia pari o inferiore a quello di chi ne fa richiesta.

La pagina è simile alla gestione dell'account personale descritta nel paragrafo 5.2.3, ma oltre a modificare i normali campi dell'account (inseriti in fase di registrazione) è possibile accedere anche al suo stato (attivo o inattivo) e livello (utente, moderatore, amministratore) e, sebbene inizialmente tutti gli account vengano generati al livello utente, è possibile eleggere nuovi moderatori o amministratori, o revocarne la carica.

Naturalmente anche in questo caso vigono le restrizioni citate nel paragrafo 5.3.1: è possibile assegnare o revocare privilegi pari o inferiori al proprio, e non è possibile revocare il privilegio di amministratore ad un account se questo è il solo ad averlo.

5.4 Gestione degli aggiornamenti

L'aggiornamento del database prevede la connessione al sito del *Research Collaboratory for Structural Bioinformatics* (RCSB) per la determinazione delle proteine di interesse e la conseguente lettura dei dati ad esse relativi.

L'analisi e l'implementazione di un'efficiente procedura di aggiornamento del database ha rappresentato la sfida più impegnativa al raggiungimento degli scopi del progetto. La procedura precedentemente realizzata presentava alcune limitazioni soprattutto dal punto di vista dell'usabilità in quanto, ad esempio, prevedeva la preparazione manuale di liste di proteine per le quali effettuare gli aggiornamenti. Un'attenzione particolare è stata rivolta alla gestione e recupero degli errori, assente nella precedente versione: l'operazione di aggiornamento può infatti impegnare la macchina anche per ore, e la possibilità di poter riprendere l'aggiornamento dopo un'interruzione viene a rappresentare una necessità irrinunciabile.

5.4.1 Analisi della procedura esistente

É stata effettuata una sorta di *reverse engineering* sulla vecchia procedura di aggiornamento per determinare il tipo di approccio utilizzato, e sono emerse alcune imperfezioni di non facile risoluzione, come il controllo sul tempo di generazione della risposta. Ogni pagina `asp`, infatti, deve essere generata entro un intervallo di tempo, solitamente di 15 minuti, dopo il quale l'esecuzione viene terminata forzatamente con un conseguente errore di timeout di esecuzione. Sebbene sia possibile aumentare questo intervallo di tempo, non è possibile determinare a priori la durata dell'aggiornamento, in quanto non è noto quali siano le proteine di interesse e quali debbano essere scartate.

Data l'assenza di meccanismi per la gestione di timeout di esecuzione, risulta impossibile controllare anche altri tipi di errore, come quelli

(molto probabili) causati da timeout dovuti alle connessioni al server remoto, e non esiste nessun meccanismo di ripristino dell'elaborazione a seguito di una qualsiasi interruzione.

Dal punto di vista del *parsing* delle informazioni scaricate non esiste una completa compatibilità con i file PDB, a causa degli enormi scostamenti tra il loro formato ideale (vedi par. 4.1) e le informazioni effettivamente presenti nei file scaricati, per cui è assente, ad esempio, la gestione delle catene con identificativo `NULL` e delle catene orfane.

Si è quindi ritenuto opportuno creare una versione totalmente nuova della procedura basata sull'esperienza accumulata dall'analisi di quella esistente nel tentativo di risolvere problemi ai quali non è stato possibile dare soluzione nella prima implementazione.

5.4.2 Approccio secondo il sistema ciclico

Il problema principale della procedura di aggiornamento sta nella necessità di dover eseguire un'operazione la cui durata è imprecisata tramite un'architettura che impone elaborazioni di una durata massima prestabilita.

Va considerato tuttavia un fattore: la durata dell'aggiornamento si protrae per un tempo più o meno lungo in base al numero t di proteine presenti nel database remoto, le quali devono essere analizzate una per una. L'idea adottata è stata perciò quella di implementare l'aggiornamento affinché proceda per insiemi discreti e predeterminati di n elementi, ciclicamente, fino al raggiungimento del totale t, dopo un numero $\lfloor (t-1)/n \rfloor + 1$ di cicli.

Il sistema utilizzato è molto semplice. Ad ogni iterazione vengono memorizzate, in variabili persistenti, informazioni sullo stato di avanzamento dell'aggiornamento (fase della procedura, ultimo elemento analizzato, totale elementi da analizzare), quindi viene generato l'output sotto forma di pagina XHTML che, in automatico e grazie ad un comando *javascript*, chiede di ricaricarsi invocando così l'iterazione successiva. Que-

sta soluzione, facile da implementare, ha apportato anche altri importanti vantaggi. Anzitutto, da non sottovalutare, la presenza di un feedback costante in grado di riportare lo stato di avanzamento o, nel caso di insorgere di errori, il tipo di problema riscontrato; ma è soprattutto la possibilità di ripristinare le interruzioni in modo molto efficiente che rende l'approccio ciclico estremamente vantaggioso: è sufficiente ricaricare la pagina perché l'operazione, il cui stato è definito in variabili persistenti, riprenda dal punto esatto in cui si è interrotta.

Variabile Persistente	Significato
`UpdateDateFrom`	Data iniziale del periodo di riferimento dell'aggiornamento
`UpdateDateTo`	Data finale del periodo di riferimento dell'aggiornamento
`UpdateType`	Tipo di aggiornamento (Proteine o FSSP)
`UpdatePhase`	Fase dell'aggiornamento
`UpdateUser`	Utente che ha invocato l'aggiornamento
`UpdateLastID`	Ultimo identificativo analizzato
`UpdateTotalPDB`	Totale degli elementi da analizzare
`UpdateCurrentPDB`	Totale degli elementi analizzati

Tabella 5.7 Variabili per la definizione dello stato dell'aggiornamento

5.4.3 Aggiornamento delle proteine

La procedura di aggiornamento, il cui accesso è riservato agli amministratori, avviene in modo automatico e prende in esame tutte le proteine rilasciate nel lasso di tempo scelto. Essa segue quattro fasi: la lettura delle nuove proteine, l'analisi delle proteine valide (termofile), il download dei dati, il parsing e la scrittura nel database.

Durante la prima fase vengono prelevati i soli identificativi delle nuove proteine. Questi identificativi (se non sono già presenti nel database) vengono scritti su una tabella di appoggio precedentemente azzerata, `UpdatingPDB`, e vengono marcati come "non validi".

Nella seconda fase vengono presi in esame tutti i record della tabella di appoggio e ne viene scaricata l'intestazione del file PDB, più leggera della versione integrale perché priva delle coordinate atomiche. Da questa è possibile determinare gli organismi dai quali deriva la proteina (*source*) ed in base a questi stabilire se è termofila o mesofila. L'informazione viene quindi registrata nella tabella di appoggio, e nel caso di termofile, l'identificativo viene marcato come "valido".

Durante la terza fase vengono considerati tutti gli identificativi validi presenti nella tabella di appoggio. Per ciascuno di essi viene scaricato il relativo file PDB, questa volta in forma integrale, che viene salvato in locale in una cartella predefinita (vedi par. 5.1.2). Vengono scaricate anche informazioni sui dettagli delle catene.

Nella quarta ed ultima fase i file PDB ed i dettagli sulle catene precedentemente scaricati vengono analizzati e le informazioni estratte vengono registrate nel database.

Per ottimizzare il processo di aggiornamento è necessario aumentare il numero di elementi analizzati ad ogni ciclo al fine di ridurre il numero di cicli; tuttavia è possibile analizzare più elementi solo ad un maggior costo computazionale, e quindi aumentando la durata dell'elaborazione. Inoltre la durata delle analisi svolte in fasi differenti possono differire anche di grandi entità. Si è deciso quindi di parametrizzare il numero di elementi analizzati in ogni fase, e variando sperimentalmente questi valori è stato possibile ridurre il numero di cicli senza tuttavia rendere il loro periodo di esecuzione così lungo da rischiare un timeout.

Fase	El.	Tipo di analisi	Note
I	25	Lettura del codice	Ad ogni ciclo avviene una sola connessione in remoto
II	5	Lettura dell'intestazione	Una connessione in remoto per ogni elemento analizzato
III	2	Lettura del file PDB	Una connessione in remoto per ogni elemento analizzato
IV	20	Analisi del file PDB	Nessuna connessione in remoto.

Tabella 5.8 Elementi analizzati per ogni ciclo

Ai fini della ciclicità della procedura aggiornamento è stato necessario aggiungere una tabella di appoggio ove registrare tutti gli identificativi delle nuove proteine, il loro tipo (mesofila o termofila) e la validità. Non è stato necessario mettere la nuova tabella in relazione con quelle esistenti, per cui non sono richieste analisi per la normalizzazione.

```
CREATE TABLE UpdatingPDB (
    PDBID TEXT (4) NOT NULL,
    Type BYTE NOT NULL,
    Valid BYTE NOT NULL,
    CONSTRAINT UPPrimaryKey PRIMARY KEY (PDBID,
    Type)
)
```

Figura 5.7 Comando SQL per la creazione della tabella UpdatingPDB

5.4.4 Analisi dei file PDB

Sono sorte numerose difficoltà durante l'implementazione della funzione di lettura dei file PDB. Infatti, nonostante le informazioni contenute in questi file rispettino idealmente una struttura ben definita (vedi par. 4.1), bisogna considerare che mentre le prime definizioni risalgono al 1972 [Ald72], il formato dei PDB viene ufficialmente definito solo nel 1996 [PDB96]. A causa dell'introduzione di nuovi campi, della ridefini-

zione di vecchi e, più banalmente, di errori di immissione i PDB letti dal sito dell'RCSB risultano ampiamente disomogenei.

É stato possibile stilare una casistica dei "scostamenti" dal formato ideale solo a seguito di una lunga analisi sperimentale, importando tutte le proteine rilasciate ed analizzando di volta in volta le discrepanze tra i dati ottenuti e quelli presenti in vecchie versioni del database.

Ad eccezione degli occasionali errori di "battitura", le maggiori differenze possono essere trovate nei file PDB inseriti prima del dicembre 1996; è infatti in questa data che il formato dei file attuale è stato definito. I file PDB antecedenti il dicembre 1996 seguono un "vecchio formato" dove molte delle informazioni sono impossibili da reperire con sistemi automatici o sono del tutto assenti. Questi problemi interessano principalmente il campo `TITLE`, i sottocampi in `COMPND` ed il campo `SOURCE`.

Per il campo `TITLE` la soluzione è stata semplice e soddisfacente: lo stesso valore è infatti presente nel sottocampo `TITL` di `JRNL`, quindi nel caso non venga trovato il primo campo si cercano le informazioni nel secondo.

Per quanto riguarda la composizione della proteina, invece, i problemi sono di più difficile risoluzione. Nelle vecchie versioni, infatti, le informazioni vengono riportate in maniera descrittiva, ed è impossibile catalogarle in modo automatico. La descrizione della composizione viene quindi salvata come "molecola fittizia", priva di tutti i dettagli comuni alle altre molecole e non conteggiata con esse. Queste molecole si contraddistinguono per il loro identificativo, sempre uguale a 9999.

Il campo `SOURCE` è quello che ha richiesto i maggiori sforzi. Bisogna infatti ricordare che solo in base al contenuto di questo campo è possibile decidere se la proteina è termofila o mesofila. Sebbene il problema non sia l'estrazione dell'informazione, il formato col quale viene presenta è di assai difficile interpretazione, specie in riferimento ai file PDB precedenti il 1996; è stato quindi necessario implementare complicate funzioni

di identificazione del genere e della specie dell'organismo "sorgente" in grado di filtrare tutti gli elementi indesiderati.

Proteina	SOURCE	Organismo
2YHX	BAKER'S YEAST (SACCHAROMYCES $CEREVISIAE)	Saccharomyces Cerevisiae
1TGL	(RHIZOMUCOR $MIEHEI)	Rhizomucor Miehei
2TEC	(THERMOACTINOMYCES $VULGARIS) AND LEECH (HIRUDO $MEDICINALIS)	Thermoactinomyces Vulgaris
1ZRP	(THERMUS THERMOPHILUS, STRAIN HB8)	Thermus Thermophilus

Tabella 5.9 Alcuni esempi di SOURCE non standard

5.4.5 Modello OO delle proteine

Come già indicato nel paragrafo 4.1, numerose informazioni all'interno di un file PDB sono correlate tra loro, in particolare per quanto riguarda `COMPND`, `SOURCE` e `SEQRES`. A causa di occasionali errori i riferimenti tra queste informazioni possono andare persi, quindi risulta necessario collezionare per intero i dati della proteina, organizzarli, controllarli ed eventualmente correggerne le ambiguità referenziali prima di procedere con la scrittura nel database. Sorge la necessità di un meccanismo in grado di gestire tutti i dati delle proteine in modo isolato per non aumentare eccessivamente la complessità delle procedure che ne fanno uso.

Si è optato per l'implementazione di una serie di classi in grado di rappresentare tutte le informazioni di una singola proteina, dotate di meccanismi per il controllo delle violazioni referenziali.

Possiamo infatti considerare una proteina come un'entità composta da molecole, le quali sono composte da catene di residui. Due catene con lo stesso identificativo non possono esistere all'interno di una stessa proteina, nemmeno se fanno parte di due molecole differenti.

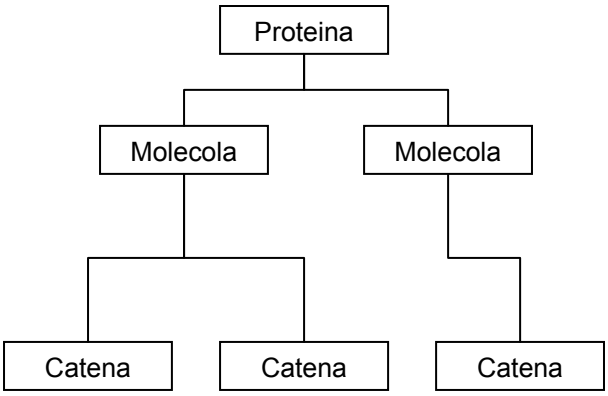


Figura 5.8 Composizione di una proteina

Sono state quindi definite tre classi: `clsChain`, `clsMolecule` e `clsProtein`. Ad ogni oggetto di tipo `clsProtein` possono essere assegnati oggetti di tipo `clsMolecule`, ed a questi possono essere assegnati oggetti di tipo `clsChain` (ottenendo una struttura come quella presente in Figura 5.8). Gli oggetti `clsProtein` possono essere interrogati sia a proposito dei propri `clsMolecule` che dei `clsChain` in essi contenuti cosicché, da un singolo oggetto `clsProtein`, è possibile accedere a tutti i dati di una proteina.

Membro	**Tipo**	**Operazione**
`sName`	Attributo	Nome univoco della catena. Può essere assegnato una sola volta
`dblLength`, `sType`	Attributo	Attributi passivi

Tabella 5.10 Interfaccia della classe clsChain

Membro	Tipo	Operazione
sId	Attributo	Identificativo della molecola. Può essere assegnato una sola volta
Fields(chainID)	Funzione	Restituisce la catena indicata
Count()	Funzione	Numero di catene associate alla molecola
NewChain(chainID)	Funzione	Crea una nuova catena e la associa alla molecola
sDescription, sMolecule, sFragment, sSynonim, sEc, sEnginereed, sMutation, sBiologicalUnit, sOtherDetails, sGenus, sSpecies	Attributo	Attributi passivi

Tabella 5.11 Interfaccia della classe clsMolecule

Membro	Tipo	Operazione
Fields(molID)	Funzione	Restituisce la molecola indicata
Chain(chainID)	Funzione	Restituisce la catena indicata
Count()	Funzione	Numero di molecole associate alla proteina
NewMol(molID)	Funzione	Crea una nuova molecola e la associa alla proteina
chainExists(chainID)	Funzione	Indica se la catena indicata esiste tra tutte le catene di tutte le molecole
sKeywords, sAtuhors, sPDBID, sTitle, dblResolution, sResolution, sExpData, bSSBond, sClassif, dRevDate, dblHOH, bPtype	Attributo	Attributi passivi

Tabella 5.12 Interfaccia della classe clsProtein

Un errore tipico presente nei file PDB è rappresentato dalla presenza di catene in SEQRES non precedentemente definite in COMPND (come accade normalmente). In assenza di meccanismi di controllo la catena verrebbe scartata, con conseguente perdita di informazioni. Grazie invece al nuovo approccio OO è facile "chiedere" alla nostra proteina se la catena esiste già (oProtein.chainExists(chainID)), ed in caso negativo creare al volo una molecola "fittizia" atta a contenerla.

```
If Not oProtein.chainExists(chainID) Then
    oProtein.NewMol = CStr(oProtein.Count + 1)
    oProtein(oProtein.Count - 1).sMolecule =
    "UNKNOWN ENTRY"
    oProtein(oProtein.Count - 1).NewChain =
    chainID
End If
```

Figura 5.9 Codice per il controllo delle catene "orfane"

Sebbene non sia comunque possibile risalire ai dati della molecola, i dati della catena evitano di andare persi e risultano correttamente referenziati.

5.4.6 Aggiornamento degli FSSP

Anche questa procedura avviene secondo l'approccio ciclico, sempre attraverso quattro fasi: il download degli FSSP, il download delle eventuali nuove proteine, il parsing e scrittura nel database dei dati delle proteine, il parsing e scrittura nel database degli FSSP.

A differenza di quanto avviene per l'aggiornamento delle proteine sappiamo immediatamente quanti e quali FSSP è necessario importare. Per determinarli è sufficiente confrontare la lista delle proteine nel database con la lista degli eventuali FSSP già presenti. Le informazioni sulle classificazioni strutturali vengono scaricate sotto forma di documenti HTML e salvati in una cartella locale (vedi par. 5.1.2). Nel caso le informazioni sugli FSSP siano assenti, viene comunque salvato un file vuoto, a testimonianza del fatto che, comunque, è già stato effettuato un controllo per quella proteina.

Dato che le classificazioni strutturali prendono sempre in considerazione due proteine (una delle quali nota), è necessario controllare la loro effettiva esistenza nel database. Qualora una delle due fosse assente, ne viene memorizzato il codice nella tabella di appoggio `UpdatingPDB` (precedentemente azzerata) marcato come "valido".

Nella seconda fase viene fatto il download dei file PDB delle proteine presenti nella tabella di appoggio. Viene effettuato un controllo sul

loro tipo (termofile o mesofile) e quindi vengono salvate in cartelle specifiche in locale (vedi par. 5.1.2). Nella terza fase gli stessi file vengono analizzati, e le informazioni estratte vengono inserite nel database.

Durante la quarta fase sono invece agli FSSP ad essere analizzati e registrati sul database, secondo il nuovo formato descritto al paragrafo 5.1.3.

Date le similitudini tra le fasi due e tre dell'aggiornamento degli FSSP e le fasi tre e quattro dell'aggiornamento delle proteine (vedi par. 5.4.3) si è deciso di apportare delle piccole modifiche a queste fasi in modo da utilizzare le stesse procedure in entrambe gli aggiornamenti. Tali modifiche riguardano principalmente il controllo del tipo della proteina, necessario solo in caso di aggiornamento degli FSSP.

5.4.7 Gestione della lista di organismi termofili

Le proteine vengono classificate in termofile o mesofile in base al tipo di organismo dal quale vengono estratte. Per distinguerne il tipo, infatti, da una proteina vengono letti i vari `SOURCE` e vengono confrontati con organismi termofili conosciuti. Se almeno un `SOURCE` corrisponde ad uno di questi organismi, la proteina viene considerata termofila.

La lista degli organismi termofili nella vecchia versione risiedeva su file; si è deciso tuttavia di incorporarla nel database in modo da rendere più veloci le ricerche.

É stata quindi aggiunta la tabella `ValidOrganism` che contiene tutti gli organismi termofili organizzati in generi e specie.

```
CREATE TABLE ValidOrganism (
    Genus TEXT (255),
    Species TEXT (255),
    CONSTRAINT VOPrimaryKey PRIMARY KEY (Genus,
    Species)
)
```

Figura 5.10 Comando SQL per la creazione della tabella ValidOrganism

L'accesso all'area di Gestione della lista di organismi termofili è consentito solo agli amministratori, e da qui è possibile aggiungere o rimuovere elementi dalla lista degli organismi termofili. Oltre ad inserire eventuali nuovi organismi riconosciuti come termofili è possibile introdurre, ad esempio, ridondanze di organismi il cui spelling appare errato all'interno di alcune proteine, consentendo il riconoscimento di queste ultime a prescindere dall'errore. Questo è quello che accade, ad esempio, per il *Sulfolobus Solfataricus* cambiato spesso in *Sulfolobus Solfactaricus* ed il *Thermus Aquaticus* cambiato in *Thermus Acuaticus*.

Genus	**Species**
Acidianus	Ambivalens
Acidothermus	Cellulolyticus
Alicyclobacillus	Acidocaldarius
Aquifex	Pyrophilus
Archaeoglobus	Fulgidus
Bacillus	Acidocaldarius
Bacillus	Caldevelox
Bacillus	Caldolyticus
Bacillus	Caldotenax
Bacillus	Caldovelox
Bacillus	Coagulans
Bacillus	Schlegelii
Bacillus	Stearothermophilus
Bacillus	Thermoproteolyticus
Clostridium	Thermocellum
Desulfurococcus	Mobilis
Desulfurococcus	Tok
Dictyoglomus	Thermophilum
Hydrogenobacter	Thermophilus
Methanobacterium	Thermoautotrophicum
Methanobacterium	Thermoformicicum
Methanocaldococcus	Janaschii
Methanococcus	Jannaschii
Methanococcus	Thermolithotrophicus
Methanopyrus	Kandleri
Methanosarcina	Barkeri
Methanosarcina	Thermophila
Methanothermobacter	Thermautotrophicus
Methanothermus	Fervidus
Moorella	Thermoacetica
Pyrobaculum	Aerophilum
Pyrococcus	Abyssi
Pyrococcus	Furiosus
Pyrococcus	Horikoshii
Pyrococcus	Kodakaraensis
Pyrococcus	Woesei

Rhizomucor	Miehei
Rhizomucor	Pusillus
Rhodothermus	Marinus
Solfolobus	Solfataricus
Staphylothermus	Marinus
Sulfolobus	Acidocaldarius
Sulfolobus	Solfactaricus
Sulfolobus	Solfataricus
Sulfolobus	Sp.
Thermoactinomyces	Vulgaris
Thermoanaerobacter	Brockii
Thermoanaerobium	Brockii
Thermoascus	Aurantiacus
Thermochromatium	Tepidum
Thermococcus	Celer
Thermococcus	Gorgonarius
Thermococcus	Kodakaraensis
Thermococcus	Litoralis
Thermococcus	Profundus
Thermococcus	Sp.
Thermomonospora	Fusca
Thermoplasma	Acidophilum
Thermosphaera	Aggregans
Thermotoga	Maritima
Thermotoga	Neapolitana
Thermus	Acuaticus
Thermus	Aquaticus
Thermus	Aquaticus Flavus
Thermus	Filiformis
Thermus	Flavus
Thermus	Sp.
Thermus	Thermophilus

Tabella 5.13 Lista degli organismi termofili (in grigio quelli erronei)

5.5 Altre modifiche

Una volta gestite iscrizioni ed aggiornamenti sono state implementate alcune utilità non espressamente richieste dalle specifiche. Sebbene di secondaria importanza ai fini della funzionalità dell'applicazione, queste utilità sono in grado di integrarle per soddisfare tutte le necessità di utenti ed amministratori.

5.5.1 Aggiornamento singola proteina

Nonostante le numerose precauzioni prese (vedi par. 5.4.4) può comunque accadere che una proteina termofila venga considerata mesofila e quindi non venga importata, o viceversa. Per ovviare questo ed altri problemi simili è stato aggiunto *l'Aggiornamento singola proteina*, uno strumento in grado di gestire il database delle proteine apportando lievi modifiche o cancellando un elemento.

É in grado di agire su una singola proteina, della quale deve essere indicato l'identificativo, e può effettuare quattro diverse operazioni. L'importazione semplice permette di scaricare ed esaminare il PDB esattamente come se si trattasse della procedura iterativa di importazione. L'importazione forzata differisce da quella semplice nel fatto che una proteina viene forzatamente considerata termofila o mesofila, in base alla scelta dell'utente. La terza operazione permette di cambiare il tipo di una proteina già presente nel database, eventualmente spostando tutti gli eventuali file ad essa correlati nelle giuste cartelle. Infine è possibile eliminare una proteina dal database, nel qual caso anche tutti i file ad essa associati vengono eliminati.

Nella loro semplicità, tramite queste quattro operazioni è possibile effettuare tutte le correzioni necessarie a sistemare gli eventuali errori di aggiornamento.

5.5.2 Statistiche database

Seppur al di fuori dei requisiti del progetto è stata implementata una tabella di riepilogo dei dati contenuti nel database. Vengono riportate informazioni a proposito del numero di proteine, molecole e catene, il numero di autori che hanno collaborato alle ricerche ed il totale delle coppie FSSP. Per ognuna di queste voci è riportato il totale relativo ai soli termofili, ai soli mesofili o ad entrambi.

Tutte le informazioni sono elaborate al momento stesso della richiesta, per cui i risultati rispecchiano sempre i contenuti del database al momento dell'interrogazione.

	Termofili	Mesofili	Totale
Proteine	1618	221	1839
Molecole	2397	334	2731
Catene	4614	583	5197
Autori	2252	467	2605
Coppie FSSP	-	-	1925

Tabella 5.14 Contenuti del database il 9 settembre 2004

5.5.3 Configurazione sistema

Alcuni parametri globali dell'applicazione possono essere soggetti a modifica anche dopo la sua installazione, in base alle preferenze degli amministratori. Per evitare nel limite del possibile interventi sul codice sorgente e massimizzarne la portabilità, si è deciso di aggiungere un modulo per la gestione della configurazione. Esso permette di modificare l'indirizzo e-mail di riferimento dell'applicazione, il sistema di gestione delle e-mail (che può essere disattivato), il tipo di iscrizione (immediato o moderato), la capitalizzazione delle stringhe e la data dell'ultimo aggiornamento effettuato.

```
CREATE TABLE Config (
    [Key] TEXT (32) [kKey] UNIQUE,
    [Value] LONGTEXT
)
```

Figura 5.11 Comando SQL per la creazione della tabella Config

Le modifiche effettuate vengono memorizzate nel database in una nuova tabella, `Config`, dalla struttura elementare dove ogni record rap-

presenta una coppia chiave-valore. I record di questa tabella vengono interpretati dall'applicazione alla prima esecuzione e dopo ogni modifica alla configurazione, limitando in tal modo gli accessi al database ai casi di effettiva necessità. I valori letti sono assegnati a variabili persistenti che vengono utilizzate alla stregua di costanti. Tali variabili sono disponibili globalmente all'interno dell'applicazione e ne definiscono il comportamento.

6 CONCLUSIONI

A differenza delle proteine derivanti da "normali" organismi, quelle derivanti da organismi termofili sono in grado di espletare la loro funzione a temperature elevate, solitamente superiori a 50°C. Questa caratteristica del Thermus Aquaticus già nel 1983 portò alla nascita di una delle più importanti e diffuse tecniche di amplificazione utilizzate in Biologia Molecolare, la PCR, con un giro d'affari iniziale di oltre 300 milioni di dollari americani. Oggi le caratteristiche delle proteine termofile vengono sfruttata in molti campi, come ad esempio nella produzione industriale, dove la loro resistenza ad alte temperature porta il beneficio di un taglio dei costi relativi ai dispositivi di raffreddamento finora indispensabili.

La ricerca è quindi orientata all'individuazione di proteine termofile strutturalmente simili a quelle mesofile in uso, in considerazione del fatto che una somiglianza strutturale spesso denota affinità di comportamento.

6.1 Il TOPDB

Il progetto di ampliamento del *Data Base of Proteins from Thermophilic Organisms* (TOPDB) punta a facilitare il compito dei ricercatori proponendo una versione controllata delle proteine termofile presenti nel *Protein Data Base* (PDB).

Le procedure di aggiornamento ed importazione dei dati permettono di eseguire facilmente e regolarmente l'integrazione dei dati delle nuove proteine rilasciati in modo da assicurare ricerche su tutte le proteine conosciute. Inoltre dal *Fold classification based on Structure-Structure ali-*

gnment of Proteins (FSSP) vengono prelevati i dati necessari a stabilire relazioni di somiglianza strutturale tra le proteine importate, relazioni alla base delle ricerche e finora disponibili solo in database separati da quelli delle proteine.

La funzione di iscrizione permette a qualsiasi ricercatore di accedere al TOPDB senza incorrere in lunghi tempi di attesa, e la gestione dei privilegi assicura l'accesso a funzioni potenzialmente pericolose solo al personale competente.

Le funzioni di ricerca sono state ottimizzate e perfezionate, e le pagine HTML prodotte dall'applicazione, oltre a presentare uno standard nell'aspetto, sono state rese compatibili con tutti i browser grazie alla conformità con i formati XHTML (*Extensible HyperText Markup Language*) e CSS (*Cascading Style Sheet*).

6.2 Sviluppi futuri

Durante lo svolgimento di questo progetto il sito dell'RCSB che ospita il PDB ha annunciato la prossima apertura di una nuova versione del portale, di cui ora è disponibile solo la versione beta. In vista di questa modifica e considerato che in fase di aggiornamento alcune informazioni vengono estratte dall'HTML del sito, può rendersi necessaria una revisione di parte del codice relativo a questa fase. Tuttavia bisogna aggiungere che tali modifiche sono già state effettuate in via sperimentale senza incorrere in particolari imprevisti, grazie alla reingegnerizzazione, ad esclusione di una piccola perdita di performance causata probabilmente dalla lentezza della versione beta del nuovo portale.

Di maggiore interesse è l'allacciamento ad una fonte diversa dall'FSSP per quanto riguarda i dati sulle somiglianze strutturali; è noto difatti che il database FSSP non viene aggiornato dai primi mesi del 2002. Una buona alternativa potrebbe essere rappresentata dal *Secondary Struc-*

ture Matching (SSM), in grado di effettuare una comparazione strutturale non solo a coppie ma anche tra più strutture.

Infine, mentre in questo progetto gli sforzi erano concentrati sulla corretta classificazione delle proteine tra termofile e mesofile, l'attenzione potrebbe essere rivolta allo sviluppo di nuovi strumenti per la manutenzione e correzione dei dati importati che, come già accennato al paragrafo 5.4.4, non sempre sono esenti da errori.

A CONSIDERAZIONI SUL CODICE PRODOTTO

Nello sviluppo di questa applicazione è stato utilizzato *Visual Source Safe*, un sistema per il controllo e la gestione del versionamento dei sorgenti, grazie al quale è stato possibile, ad esempio, effettuare delle modifiche alla procedura di aggiornamento senza tuttavia dover abbandonare la versione precedente del codice.

Grazie a questo strumento è inoltre possibile effettuare delle considerazioni sull'andamento del lavoro svolto grazie ai rapporti generati. Una prima considerazione riguarda le righe di codice dei file modificati. A seguito della suddivisione in moduli (vedi cap. 5.1.1), tutte le modifiche hanno apportato una riduzione della lunghezza; questo è particolarmente significativo per il file `resultlist.asp` in questo caso infatti oltre ad essere stati aggiunti meccanismi per la segnalazione degli errori (prima in `error.asp`) gestisce ora sia la ricerca avanzata che quella semplice (prima in `defreslist.asp`).

File Modificati	Prima	Dopo	Funzione
/advsearch.asp	244	136	Richiesta ricerca avanzata
/autentica.asp	59	74	Gestione autentificazione
/default.asp	224	120	Pagina iniziale
/fsspvisual.asp	229	222	Visualizzazione FSSP
/Info.htm	8	8	File di Guida
/Infooffields.htm	8	8	File di Guida
/resultlist.asp	588	420	Risultati ricerche
/visualizzapdb.asp	305	231	Dettagli proteina
Totale	**1665**	**1219**	

Tabella A.1 File modificati e relative dimensioni

File Aggiunti	Righe	Funzione
/configPage.asp	111	Configurazione
/GetFSSPList.asp	465	Aggiornamento FSSP
/GetPDB.asp	263	Aggiornamento singola proteina
/GetPDBList.asp	478	Aggiornamento PDB
/login.asp	65	Maschera richiesta login
/sourcelist.asp	242	Gestione organismi termofili
/statistic.asp	253	Statistiche e aggiornamenti
/userdata.asp	562	Gestione profilo
/userlist.asp	190	Gestione account
/include/config.asp	190	Configurazione globale
/include/footer.asp	37	Piede pagina
/include/function.asp	393	Funzioni
/include/header.asp	140	Intestazione pagina
/include/listfooter.asp	2	Piede pagina liste
/include/listheader.asp	28	Intestazione pagina liste
/include/mail.asp	84	Gestione e-mail
/include/plist.asp	218	Funzioni per le liste
/include/update.asp	1865	Funzioni per gli aggiornamenti
/styles/normal.css	802	Foglio di stile
Totale	**6388**	

Tabella A.2 File aggiunti e relative dimensioni

File Eliminati	Righe	Funzione
/defreslist.asp	285	Risultati ricerca semplice
/error.asp	103	Messaggi di errore
/login.htm	58	Maschera richiesta login
/Statistic.htm	7	Statistiche
/Update/Agg.asp	223	Inizializzazione aggiornamento
/Update/download_det.asp	86	Download dettagli catene
/Update/download_meso.asp	149	Download mesofili
/Update/download_termo.asp	117	Download termofili
/Update/Inserimento.asp	17	Importazione dati
/Update/query_generator.asp	1106	Decodifica dati e scrittura su DB
Totale	**2151**	

Tabella A.3 File eliminati e relative dimensioni

Il prodotto finito è composto da 7607 righe di codice, contro le 3816 del prodotto iniziale. Le righe di codice prodotte in base alle informazioni di VSS sono 5942, anche se non viene tenuto conto del codice riscritto.

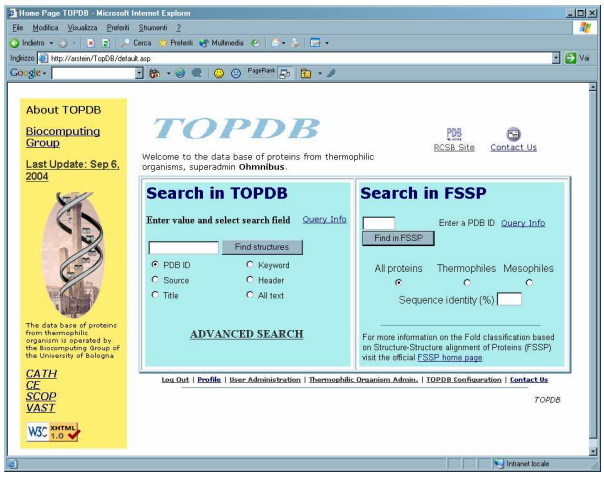
B ESEMPIO DI FILE PDB: 1KTQ

Segue un esempio di file PDB privo delle coordinate atomiche. Come esempio è stato preso il *DNA Polymerase* del *Thermus Aquaticus* che rese possibile la nascita del PCR.

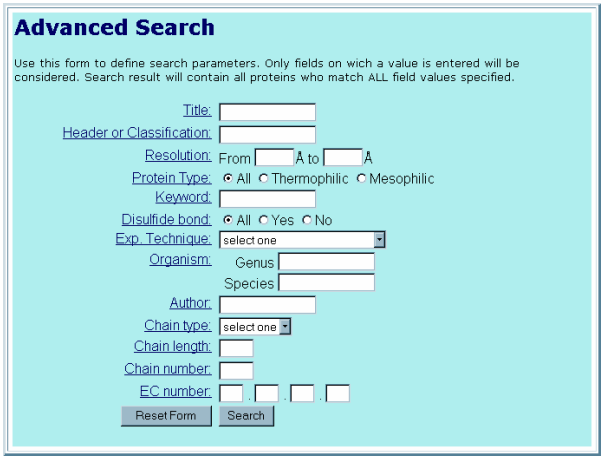
```
HEADER    NUCLEOTIDYLTRANSFERASE                  16-AUG-95   1KTQ
TITLE     DNA POLYMERASE
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: DNA POLYMERASE I;
COMPND   3 CHAIN: NULL;
COMPND   4 SYNONYM: KLENTAQ;
COMPND   5 EC: 2.7.7.7;
COMPND   6 ENGINEERED: YES
SOURCE    MOL_ID: 1;
SOURCE   2 ORGANISM_SCIENTIFIC: THERMUS AQUATICUS;
SOURCE   3 EXPRESSION_SYSTEM: ESCHERICHIA COLI
KEYWDS    NUCLEOTIDYLTRANSFERASE, DNA-REPLICATION
EXPDTA    X-RAY DIFFRACTION
AUTHOR    S.KOROLEV,G.WAKSMAN
REVDAT   1   08-NOV-96 1KTQ    0
JRNL        AUTH   S.KOROLEV,M.NAYAL,W.M.BARNES,E.DI CERA,G.WAKSMAN
JRNL        TITL   CRYSTAL STRUCTURE OF THE LARGE FRAGMENT OF THERMUS
JRNL        TITL 2 AQUATICUS DNA POLYMERASE I AT 2.5-A RESOLUTION:
JRNL        TITL 3 STRUCTURAL BASIS FOR THERMOSTABILITY
JRNL        REF    PROC.NAT.ACAD.SCI.USA         V.  92  9264 1995
JRNL        REFN   ASTM PNASA6  US ISSN 0027-8424                0040
REMARK   1
REMARK   2
REMARK   2 RESOLUTION. 2.5  ANGSTROMS.

      [Sono state eliminate 332 righe di tipo REMARK]

DBREF  1KTQ    290   503  SWS    P19821   DPO1_THEAQ     290    503
DBREF  1KTQ    513   832  SWS    P19821   DPO1_THEAQ     513    832
SEQADV 1KTQ             SWS  P19821    GLY   504 GAP IN PDB ENTRY
SEQADV 1KTQ             SWS  P19821    LYS   505 GAP IN PDB ENTRY
SEQADV 1KTQ             SWS  P19821    THR   506 GAP IN PDB ENTRY
SEQADV 1KTQ             SWS  P19821    GLU   507 GAP IN PDB ENTRY
SEQADV 1KTQ             SWS  P19821    LYS   508 GAP IN PDB ENTRY
SEQADV 1KTQ             SWS  P19821    THR   509 GAP IN PDB ENTRY
SEQADV 1KTQ             SWS  P19821    GLY   510 GAP IN PDB ENTRY
SEQADV 1KTQ             SWS  P19821    LYS   511 GAP IN PDB ENTRY
SEQADV 1KTQ             SWS  P19821    ARG   512 GAP IN PDB ENTRY
SEQRES   1    543  SER PRO LYS ALA LEU GLU GLU ALA PRO TRP PRO PRO PRO
SEQRES   2    543  GLU GLY ALA PHE VAL GLY PHE VAL LEU SER ARG LYS GLU
SEQRES   3    543  PRO MET TRP ALA ASP LEU LEU ALA LEU ALA ALA ALA ARG
SEQRES   4    543  GLY GLY ARG VAL HIS ARG ALA PRO GLU PRO TYR LYS ALA
SEQRES   5    543  LEU ARG ASP LEU LYS GLU ALA ARG GLY LEU LEU ALA LYS
SEQRES   6    543  ASP LEU SER VAL LEU ALA LEU ARG GLU GLY LEU GLY LEU
SEQRES   7    543  PRO PRO GLY ASP ASP PRO MET LEU LEU ALA TYR LEU LEU
SEQRES   8    543  ASP PRO SER ASN THR THR PRO GLU GLY VAL ALA ARG ARG
SEQRES   9    543  TYR GLY GLY GLU TRP THR GLU GLU ALA GLY GLU ARG ALA
SEQRES  10    543  ALA LEU SER GLU ARG LEU PHE ALA ASN LEU TRP GLY ARG
SEQRES  11    543  LEU GLU GLY GLU GLU ARG LEU LEU TRP LEU TYR ARG GLU
SEQRES  12    543  VAL GLU ARG PRO LEU SER ALA VAL LEU ALA HIS MET GLU
SEQRES  13    543  ALA THR GLY VAL ARG LEU ASP VAL ALA TYR LEU ARG ALA
SEQRES  14    543  LEU SER LEU GLU VAL ALA GLU GLU ILE ALA ARG LEU GLU
```

```
SEQRES  15    543  ALA GLU VAL PHE ARG LEU ALA GLY HIS PRO PHE ASN LEU
SEQRES  16    543  ASN SER ARG ASP GLN LEU GLU ARG VAL LEU PHE ASP GLU
SEQRES  17    543  LEU GLY LEU PRO ALA ILE GLY LYS THR GLU LYS THR GLY
SEQRES  18    543  LYS ARG SER THR SER ALA ALA VAL LEU GLU ALA LEU ARG
SEQRES  19    543  GLU ALA HIS PRO ILE VAL GLU LYS ILE LEU GLN TYR ARG
SEQRES  20    543  GLU LEU THR LYS LEU LYS SER THR TYR ILE ASP PRO LEU
SEQRES  21    543  PRO ASP LEU ILE HIS PRO ARG THR GLY ARG LEU HIS THR
SEQRES  22    543  ARG PHE ASN GLN THR ALA THR ALA THR GLY ARG LEU SER
SEQRES  23    543  SER SER ASP PRO ASN LEU GLN ASN ILE PRO VAL ARG THR
SEQRES  24    543  PRO LEU GLY GLN ARG ILE ARG ARG ALA PHE ILE ALA GLU
SEQRES  25    543  GLU GLY TRP LEU LEU VAL ALA LEU ASP TYR SER GLN ILE
SEQRES  26    543  GLU LEU ARG VAL LEU ALA HIS LEU SER GLY ASP GLU ASN
SEQRES  27    543  LEU ILE ARG VAL PHE GLN GLU GLY ARG ASP ILE HIS THR
SEQRES  28    543  GLU THR ALA SER TRP MET PHE GLY VAL PRO ARG GLU ALA
SEQRES  29    543  VAL ASP PRO LEU MET ARG ARG ALA ALA LYS THR ILE ASN
SEQRES  30    543  PHE GLY VAL LEU TYR GLY MET SER ALA HIS ARG LEU SER
SEQRES  31    543  GLN GLU LEU ALA ILE PRO TYR GLU GLU ALA GLN ALA PHE
SEQRES  32    543  ILE GLU ARG TYR PHE GLN SER PHE PRO LYS VAL ARG ALA
SEQRES  33    543  TRP ILE GLU LYS THR LEU GLU GLU GLY ARG ARG ARG GLY
SEQRES  34    543  TYR VAL GLU THR LEU PHE GLY ARG ARG ARG TYR VAL PRO
SEQRES  35    543  ASP LEU GLU ALA ARG VAL LYS SER VAL ARG GLU ALA ALA
SEQRES  36    543  GLU ARG MET ALA PHE ASN MET PRO VAL GLN GLY THR ALA
SEQRES  37    543  ALA ASP LEU MET LYS LEU ALA MET VAL LYS LEU PHE PRO
SEQRES  38    543  ARG LEU GLU GLU MET GLY ALA ARG MET LEU LEU GLN VAL
SEQRES  39    543  HIS ASP GLU LEU VAL LEU GLU ALA PRO LYS GLU ARG ALA
SEQRES  40    543  GLU ALA VAL ALA ARG LEU ALA LYS GLU VAL MET GLU GLY
SEQRES  41    543  VAL TYR PRO LEU ALA VAL PRO LEU GLU VAL GLU VAL GLY
SEQRES  42    543  ILE GLY GLU ASP TRP LEU SER ALA LYS GLU
FORMUL   2  HOH   *440(H2 O1)
HELIX    1   1 PRO    316  TRP    318  5                                    3
HELIX    2   2 PRO    338  ASP    344  1                                    7
HELIX    3   3 ALA    353  ARG    362  1                                   10
HELIX    4   4 PRO    373  LEU    380  1                                    8
HELIX    5   5 PRO    387  TYR    394  1                                    8
HELIX    6   6 ALA    402  GLU    421  1                                   20
HELIX    7   7 GLU    424  GLU    432  1                                    9
HELIX    8   8 GLU    434  THR    447  1                                   14
HELIX    9   9 VAL    453  ALA    478  1                                   26
HELIX   10  10 ARG    487  PHE    495  1                                    9
HELIX   11  11 VAL    518  ALA    521  1                                    4
HELIX   12  12 PRO    527  TYR    545  5                                   19
HELIX   13  13 LEU    549  LEU    552  5                                    4
HELIX   14  14 LEU    581  ASN    583  5                                    3
HELIX   15  15 PRO    589  ALA    597  1                                    9
HELIX   16  16 ILE    614  SER    623  1                                   10
HELIX   17  17 GLU    626  GLN    633  1                                    8
HELIX   18  18 ILE    638  PHE    647  1                                   10
HELIX   19  19 PRO    656  VAL    669  5                                   14
HELIX   20  20 ALA    675  GLU    681  1                                    7
HELIX   21  21 GLU    688  SER    699  1                                   12
HELIX   22  22 PRO    701  ARG    717  1                                   17
HELIX   23  23 PRO    731  GLU    734  5                                    4
HELIX   24  24 LYS    738  MET    775  1                                   38
HELIX   25  25 LYS    793  GLU    808  5                                   16
HELIX   26  26 TRP    827  ALA    830  1                                    4
SHEET    1   A 3 ARG   331  ARG    334  0
SHEET    2   A 3 LEU   321  ARG    328 -1  N  ARG   328   O  ARG   331
SHEET    3   A 3 PHE   306  LEU    311 -1  N  VAL   310   O  LEU   322
SHEET    1   B 4 VAL   819  GLY    824  0
SHEET    2   B 4 TRP   604  TYR    611 -1  N  ASP   610   O  GLU   820
SHEET    3   B 4 GLU   786  PRO    792 -1  N  ALA   791   O  LEU   605
SHEET    4   B 4 ARG   778  VAL    783 -1  N  VAL   783   O  GLU   786
SHEET    1   C 2 TYR   719  GLU    721  0
SHEET    2   C 2 ARG   727  TYR    729 -1  N  ARG   728   O  VAL   720
CISPEP   1 TRP    299    PRO    300          0        -0.31
CISPEP   2 ASP    578    PRO    579          0       -10.40
CRYST1  109.400  136.800   45.600  90.00  90.00  90.00 P 21 21 2     4
ORIGX1      1.000000  0.000000  0.000000        0.00000
ORIGX2      0.000000  1.000000  0.000000        0.00000
ORIGX3      0.000000  0.000000  1.000000        0.00000
SCALE1      0.009141  0.000000  0.000000        0.00000
SCALE2      0.000000  0.007310  0.000000        0.00000
SCALE3      0.000000  0.000000  0.021930        0.00000
```

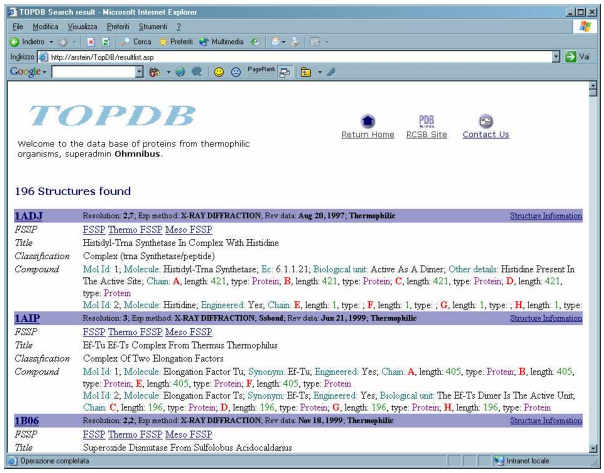
C ASPETTO DELL'APPLICAZIONE



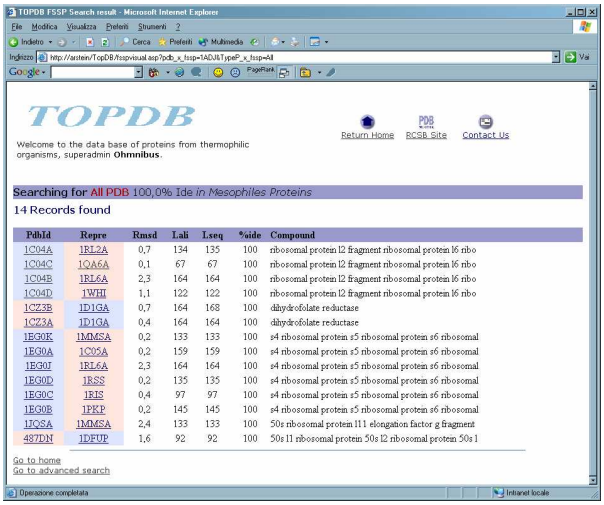
Home Page



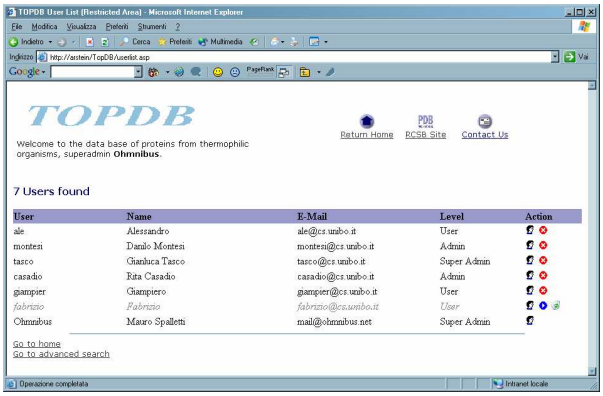
Ricerca Avanzata (dettaglio)



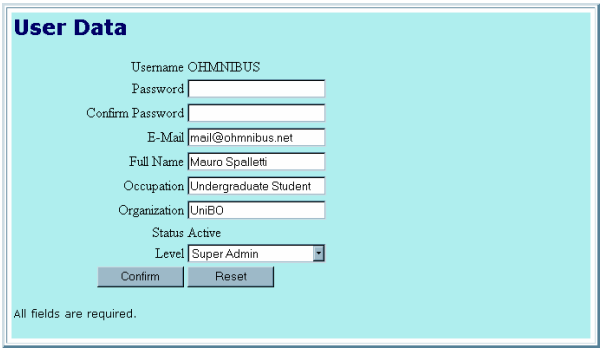
Risultati Ricerca



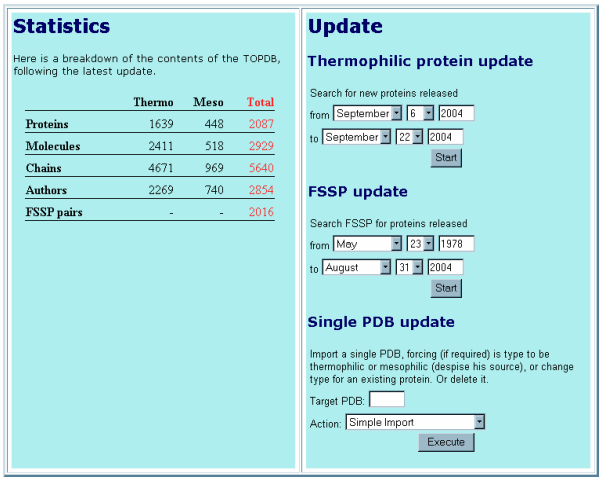
Risultati Ricerca per FSSP



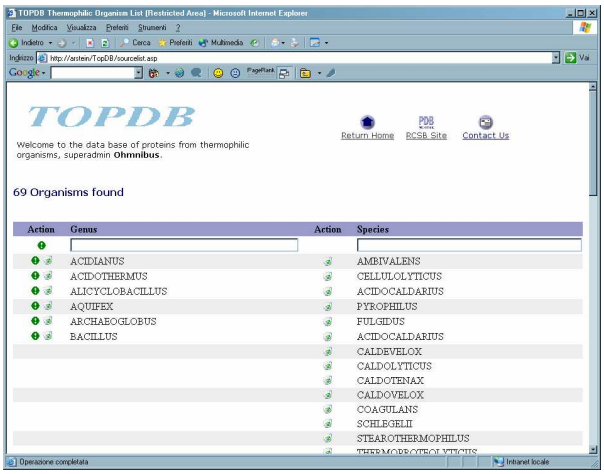
Amministrazione Utenti



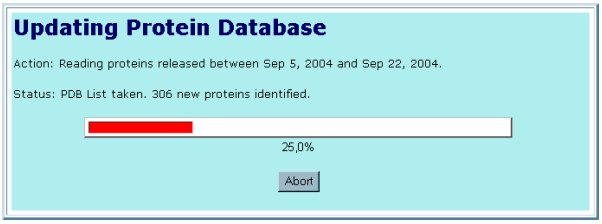
Gestione Account (dettaglio)



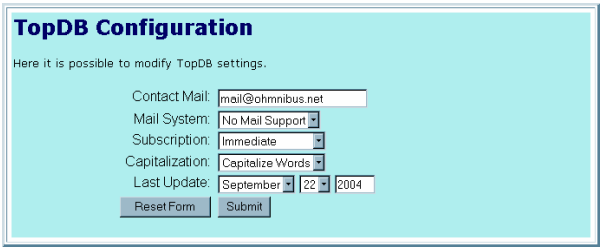
Statistiche ed Aggiornamenti (dettaglio)



Amministrazione Organismi Termofili



Stato dell'aggiornamento (dettaglio)



Configurazione (dettaglio)

BIBLIOGRAFIA

[ACPT99] Atzeni P., Ceri S., Paraboschi S., Torlone R., "Analisi delle Ridondanze", in: *Basi di dati*, Seconda edizione, Milano, McGraw-Hill, 1999, 233-236

[Ald72] Alden R.A., Birktoft J.J., Kraut J., Robertus J.D., Wright C.S., "Atomic coordinates for subtilisin BPN*", 1972, http://www.pdb.mdc-berlin.de/pdb/cgi/explore.cgi?pdbId=1SBT, 17 settembre 2004

[Anf61] Anfinsen C., Haber E., Sela M., White F., "The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain", *Proceedings of the National Academy of Sciences*, 47, 1961, 1309-1314

[Anf73] Anfinsen C., "Principles that govern the folding of protein chains", *Science*, 181, 1973, 223-230

[Bar91] Baross J.A., *Hyperthermophilic Archaea: Implications for the Origin and Early Evolution of Life at Submarine Hydrothermal Vents*, Eos, 1991

[BBBW72] Brock T.D., Brock K.M., Belly R.T., Weiss R.L., *Sulfolobus: a new genus of sulfur-oxidizing bacteria living at low pH and high temperature*, Arch Microbiol, 1972

[FraWol94] Frauenfelder H., Wolynes P.G., "Biomolecules Where the physics of complexity and simplicity meet", *Physics Today*, February, 1994, 47-58

[HåkBer99] Håkon W.L., Bert B., "Cascading Style Sheets, level 1", 1999, http://w3c.org/TR/CSS1, 17 settembre 2004

[Hol95] Holm L. ed altri, "The Dali Server", 1995, http://www.ebi.ac.uk/dali/index.html, 08 settembre 2004

[Hol01] Holm L. ed altri, "Dali Domain Dictionary v.3.1beta", 2001, http://www.ebi.ac.uk/dali/domain/3.1beta/, 08 settembre 2004

[Lil03] Lilley C., W3C, "Separation of semantic and presentational markup, to the extent possible, is architecturally sound", 2003, http://www.w3.org/2001/tag/doc/contentPresentation-26.html, 17 settembre 2004

[Mic04] Microsoft, "Active Server Pages", 2004, http://msdn.microsoft.com/library/default.asp?url=/nhp/default.asp?contentid=28000522&frame=true, 17 settembre 2004

[Mic04b] Microsoft, "What Is VBScript?", 2004, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vtoriversioninformation.asp, 17 settembre 2004

[Mur03] Murzin A.G., Lo Conte L., Andreeva A., Howorth D., Ailey B.G., Brenner S.E., Hubbard T.J.P, Chothia C., "Structural Classification of Proteins", 2003, http://scop.mrc-lmb.cam.ac.uk/scop/index.html, 08 settembre 2004

[PDB96] PDB staff, "Protein Data Bank Contents Guide: Atomic Coordinate Entry Format Description", 1996, http://www.rcsb.org/pdb/docs/format/pdbguide2.2/guide2.2_frame.html, 17 settembre 2004

[Pem02] Pemberton S. e altri, "XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)", 2002, http://www.w3.org/TR/xhtml1/, 27 agosto 2004

[RCSB04] RCSB, "PDB Current Holdings", 2004, http://www.rcsb.org/pdb/holdings.html, 01 settembre 2004

[Ros93] Rosso L., Lobry J.R., Flandrois J.P., "An unexpexted correlation between cardinal temperatures of microbial growth highlighted by a new model", *Journal of Theoretical Biology*, 162, 1993, 447-463

RINGRAZIAMENTI

Ringrazio anzitutto i miei genitori, Pierino e Rosanna, per essere sempre stati favorevoli ad ogni mia scelta, e per avermi insegnato i più genuini valori della vita.

Ringrazio la mia ragazza, Chiara, per essermi stata sempre così vicina e per avermi arricchito di enorme gioia. Non meno, per aver arginato i miei orrori grammaticali =)

Ringrazio il prof. Montesi, la prof.ssa Casadio ed il dott. Tasco per avermi seguito durante lo svolgimento della tesi.

Ringrazio Matteo per le lunghe e corroboranti chiacchierate "informatiche" di fronte ad un bel bicchiere di birra.

Ringrazio i miei coinquilini Costante, Davide e Paolo per la forza che dimostrano nel sopportarmi, e più in generale tutti i miei amici di Bologna, di Recanati e del resto del mondo.

Ringrazio Gary Gygax, per aver gettato nel '74 le basi degli odierni giochi di ruolo, ed aver così aperto porte su mondi dove nessun uomo è mai stato prima.

Un sincero grazie a tutti.