Mining di pattern complessi di superficie da strutture molecolari*

Lorenzo Baldacci, Matteo Golfarelli e Stefano Rizzi

DEIS, Università di Bologna - Viale Risorgimento, 2 - 40136 Bologna

Sommario Nell'ambito della bioinformatica riveste particolare interesse il ruolo giocato nel processo biologico dalle proteine, che fungono da trasmettitori e ricevitori di informazioni regolando i meccanismi che determinano il funzionamento dei sistemi organici. Recenti studi hanno evidenziato una stretta correlazione tra le caratteristiche della superficie delle proteine e il loro modo di interagire. In questo lavoro presentiamo un approccio originale alla classificazione delle proteine sulla base di caratteristiche di superficie. Il lavoro si focalizza in particolare sulla definizione di pattern di superficie e sulla descrizione dell'algoritmo di mining per la ricerca di pattern ricorrenti.

1 Introduzione

Dopo che, alla fine del secolo scorso, è stato completato il sequenziamento del genoma umano, l'interesse della ricerca bioinformatica si è spostato sulle proteine: la nuova sfida è determinare il loro comportamento all'interno delle cellule. Alla base di tale interesse è il ruolo giocato dalle proteine nel processo biologico; esse infatti fungono da trasmettitori e ricevitori di informazioni innescando e regolando gran parte dei meccanismi che determinano il corretto funzionamento dei sistemi organici. È quindi ovvio che conoscere le funzioni svolte dalle diverse proteine è fondamentale in settori quali la medicina, la farmaceutica e la chimica.

Per comprendere il funzionamento delle proteine è necessario classificarle, e per fare ciò è indispensabile poterle comparare per identificare gli elementi comuni. Una proteina è costituita da una catena di aminoacidi che ne individua la cosiddetta *struttura primaria*. Ogni catena polipeptidica, a causa delle forze elettriche generate dai propri atomi, si ripiega in una caratteristica struttura spaziale detta *struttura terziaria*. Esistono inoltre dei motivi ricorrenti nella conformazione spaziale chiamati elementi di *struttura secondaria*. Le principali strade per la classificazione percorse sino a oggi sono l'*allineamento sequenziale*, che applica tecniche di string matching [1] alla struttura primaria, e l'*allineamento strutturale*, che opera sulla struttura secondaria e terziaria considerando quindi nel suo complesso la struttura tridimensionale della proteina [2].

Una nuova tendenza in fatto di comparazione e classificazione si basa sull'assunto che l'interazione tra proteine è determinata fondamentalmente dalle caratteristiche di superficie (forma, potenziale elettrico, ecc.): in altre parole, la

* This work was partially funded by the IST programme of the European Community, Future and Emerging Technologies under the IST-2001-33058 PANDA project.

superficie della proteina rappresenta l'interfaccia attraverso cui essa interagisce con l'ambiente esterno. Uno degli ambiti di ricerca in cui, su questa base, si sono ottenuti ottimi risultati è quello del *docking biomolecolare* [3], nel quale ci si chiede se due proteine potranno interagire tra loro formando un contatto energetico stabile. La capacità di interagire dipende dall'esistenza sulle due proteine di aree di sufficiente grandezza che presentino caratteristiche compatibili, ad esempio carica elettrica positiva e forma convessa da una parte, carica elettrica negativa e forma concava dall'altra. L'utilizzo di una singola area ai fini di classificazione (cfr. [4]) non è tuttavia sempre efficace, poiché le proprietà dell'intera proteina potrebbero essere determinate da un insieme di regioni non necessariamente limitrofe.

Scopo del progetto di ricerca che abbiamo intrapreso è determinare, dato un database (DB) di proteine di cui non sono note le funzioni, l'insieme dei *pattern di superficie complessi* che si presentano frequentemente, al fine di permettere la classificazione delle proteine. Tale risultato, come illustrato in Figura 1, richiede (1) l'utilizzo di tecniche di clustering per determinare, a partire da una rappresentazione dettagliata della superficie proteica, un insieme di regioni caratteristiche; (2) di definire una rappresentazione sintetica e al tempo stesso efficace della superficie proteica; e (3) di applicare a questa rappresentazione appropriate tecniche di data mining per determinare i pattern frequenti.

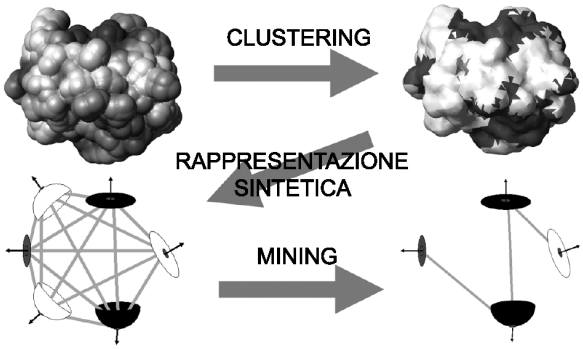

Figura 1. L'approccio proposto: dalle superfici proteiche ai pattern frequenti

I principali contributi di questo articolo sono:

- L'introduzione di una rappresentazione sintetica per le superfici proteiche basata su grafi di regioni omogenee anziché su singole regioni, e la corrispondente definizione di pattern di superficie;
- La proposta di un algoritmo originale di mining per individuare pattern di superficie frequenti.

Questi due argomenti sono trattati, rispettivamente, nelle Sezioni 2 e 3. Inoltre, la Sezione 4 discute gli ambiti applicativi delle tecniche proposte.

2 Rappresentazione sintetica delle superfici proteiche e pattern di superficie

È data una collezione di proteine di cui è nota la rappresentazione vettoriale della superficie, composta da mesh triangolari etichettate con un insieme di caratteristiche puntuali quali il potenziale elettrico e il fattore di idrofobicità. Nello studio delle interazioni tra proteine, il passaggio a una rappresentazione meno dettagliata è incoraggiato, oltre che da vincoli di carattere computazionale, anche dalle considerazioni di seguito elencate:

- *Dimensione delle aree di interazione.* Biologi e chimici concordano sul fatto che, affinché due proteine possano interagire, la porzione di superficie compatibile deve essere all'incirca il 5% dell'intera superficie, mentre le mesh normalmente utilizzate coprono in media un'area non superiore allo 0,01%.
- *Modalità di interazione.* Durante la fase di interazione le proteine dimostrano un'elevata flessibilità che permette loro di modificare parzialmente la forma della superficie. Risulta quindi inutile modellare con esattezza dettagli che non incidono sulle possibilità di interazione.
- *Informazioni spaziali.* Le mesh triangolari non presentano singolarmente l'informazione spaziale di concavità/convessità che caratterizza invece regioni più ampie e che è un fattore fondamentale per l'interazione.
- *Omogeneità delle caratteristiche.* L'interazione tra proteine avviene per regioni con caratteristiche di superficie omogenee.

Ciò suggerisce di adottare una rappresentazione basata, anziché su mesh, su *regioni omogenee* di superficie. La loro individuazione può essere effettuata utilizzando tecniche di clustering che si basano sul vettore di caratteristiche associato ad ogni mesh. Il processo di clustering prevede inizialmente una stima delle proprietà geometriche della superficie, effettuata partendo dalla rappresentazione discreta tramite mesh e calcolando sui punti di superficie gli indici di curvatura media e gaussiana. Questi ultimi vengono poi utilizzati insieme alle caratteristiche puntuali per la ricerca, attraverso un approccio di tipo *boundary-based* [5], di cluster corrispondenti a regioni di superficie connessa e con caratteristiche omogenee.

Le proprietà delle proteine dipendono non solo dall'insieme di regioni che ne caratterizzano la superficie, ma anche dalla loro disposizione e orientazione relative. La rappresentazione sintetica che proponiamo per la proteina D_i è pertanto un grafo completamente connesso i cui nodi sono le regioni $d_1^i, \ldots, d_m^i$ risultanti dal clustering, ciascuna descritta dal valore medio delle caratteristiche superficiali e dall'ampiezza della regione. (Figura 2.a). L'arco tra due regioni d_1^i, d_2^i esprime la loro posizione relativa mediante: (1) la lunghezza γ del vettore che congiunge il baricentro delle due regioni, (2) gli angoli α_1 e α_2 formati da tale vettore con i versori delle regioni e (3) l'angolo β tra i due versori calcolato proiettandoli sul piano ortogonale al vettore che li congiunge (Figura 2.b) [6].

Siamo ora in grado di definire un *pattern* P^i appartenente alla proteina D_i come un sottografo completamente connesso del grafo che ne rappresenta sinteticamente la superficie; chiameremo *livello* di un pattern il numero di regioni che

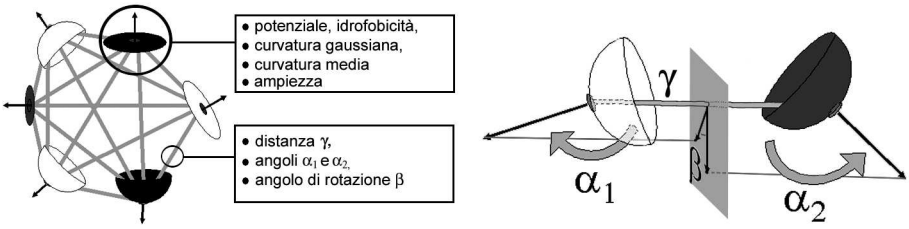

Figura 2. (a) Grafo delle regioni di una proteina; (b) Informazioni necessarie a calcolare la posizione relativa di due regioni

lo costituiscono. Si noti che le informazioni sugli archi sottendono una metrica Euclidea e pertanto la rappresentazione tramite grafo completamente connesso risulta ridondante. Di conseguenza, la ricerca di pattern frequenti non verrà affrontata come mining di sottografi [7] che, richiedendo l'uso di tecniche di determinazione di isomorfismi, renderebbe il problema intrattabile all'aumentare delle dimensioni degli oggetti trattati [8].

3 Mining di pattern complessi

L'algoritmo di mining proposto in questa sezione, partendo dal DB di proteine $\mathcal{D}$ rappresentate in forma sintetica, ricerca i pattern frequenti composti da due o più regioni. Un pattern P è *frequente* se esiste in $\mathcal{D}$ un insieme, con cardinalità maggiore o uguale a una soglia *minsupp*, di pattern simili a P. Tale insieme è detto *supporto* di P. La funzione di similarità, che non specificheremo in questo lavoro, deve considerare ovviamente sia le informazioni puntuali delle regioni, sia la loro disposizione relativa.

La determinazione di pattern frequenti di cui sono ignoti e variabili il livello e la composizione dà luogo a uno spazio di ricerca di dimensione esponenziale. Nell'ambito del data mining sono state proposte numerose tecniche, la cui applicabilità dipende dalle caratteristiche del dominio applicativo [9]. In particolare, la densità/sparsità dei dati e il livello di correlazione tra gli elementi sono fattori chiave di scelta. Ad esempio, la classe di algoritmi di *mining orizzontale a livelli* (a cui appartiene APriori [10]) identifica dapprima tutti i pattern frequenti con cardinalità 1 e quindi genera selettivamente i pattern candidati con cardinalità via via superiore sfruttando il vincolo antimonotono di frequenza dei sotto-pattern[1] ed eventualmente altri vincoli propri del dominio applicativo. Ogni iterazione prevede una fase di generazione e una fase di validazione sul DB dell'effettiva ampiezza del supporto dei candidati.

Al contrario, gli algoritmi di *mining verticale* [11] evitano l'accesso al DB mantenendo un più ampio insieme di strutture dati accessorie che permettono

[1] Un vincolo è antimonotono se, essendo soddisfatto da un certo insieme, lo è anche da ogni suo sottoinsieme.

```
for each D_i ∈ 𝒟
{  ℒ_1^i = GetFrequentRegions(D_i);
   MineFrequentPatterns(ℒ_1^i);
}
```

Figura 3. Ciclo principale di mining

di determinare direttamente il supporto di un certo pattern. La prima famiglia di algoritmi è più adatta a domini applicativi sparsi (ossia con dimensione dei pattern ridotta) poiché in questi il numero di accessi al DB è comunque limitato, mentre ovviamente la seconda è più adatta per domini densi.

Il problema trattato in questo articolo non può essere affrontato direttamente tramite tecniche di mining orizzontale o verticale, necessita piuttosto dell'utilizzo congiunto di entrambe. Inoltre, esso richiede la messa a punto di soluzioni originali per il trattamento delle seguenti specificità:

- *Relazione di similarità tra pattern*: la relazione utilizzata durante il confronto tra pattern, e in particolare nel calcolo del supporto, non è l'uguaglianza bensì la similarità.
- *Presenza di vincoli spaziali tra le regioni di un pattern*: parte delle informazioni che caratterizzano i pattern non sta sulle singole regioni bensì sugli archi, ossia i pattern sono caratterizzati anche dalla disposizione spaziale delle regioni.

La prima considerazione implica che non esiste un insieme di regioni di riferimento, ma ogni regione è *unica* nel DB. Inoltre, la similarità non è una proprietà transitiva, ossia $P \sim Q \wedge Q \sim S \not\Rightarrow P \sim S$. Da queste riflessioni si ricava che il concetto di frequenza è dipendente dal pattern (e dalla proteina) utilizzato come prototipo e che quindi il processo di mining dovrà essere innescato fissata una proteina di riferimento.

In Figura 3 è riportato il ciclo principale dell'algoritmo proposto: la procedura `GetFrequentRegions`(D_i) determina l'insieme $\mathcal{L}_1^i$ delle singole regioni (pattern di livello 1) della proteina D_i che sono frequenti, mentre `MineFrequentPatterns`($\mathcal{L}_1^i$) innesca il processo di mining vero e proprio.

Se da un lato la dipendenza del risultato del mining dallo specifico pattern usato come prototipo rende il problema computazionalmente più complesso, dall'altro consente di utilizzare in `MineFrequentPatterns` una maggior quantità di strutture dati accessorie dato che il numero di pattern frequenti per ogni proteina sarà ovviamente molto limitato rispetto a quello dell'intero DB. Si sottolinea che la codifica riportata in Figura 3 ha puramente carattere esplicativo: in fase di implementazione risulta infatti possibile elaborare contemporaneamente gruppi di proteine al fine di ridurre il numero di iterazioni, essendo la dimensione dei gruppi funzione dello spazio disponibile in memoria centrale.

In Figura 4 è riportato il nucleo centrale di pseudo-codice per l'algoritmo, mentre in Tabella 1 è riassunta la legenda dei simboli per facilitarne l'interpretazione. Useremo le lettere maiuscole da P a T per denotare i pattern e le

```
1.  MineFrequentPatterns(L_1^i)
2.  {  for (k = 2, L_{k-1}^i ≠ ∅, k++) do
3.     {  C = CandidatePatterns(L_{k-1}^i);
4.        for each D_j ∈ D do
5.           for each P^i ∈ C; D_j ∈ Prot(P^i) do
6.           {  for each Q^j ∈ Supp_j(P^i) do
7.                 if Simil(Q^j,P^i)< σ
8.                    Supp(P^i)\ = {Q^j};
9.           }
10.       L_k^i = {P^i ∈ C; | Supp(P^i) |≥ minsupp};
11.    }
12. }
```

Figura 4. Algoritmo di Mining

Tabella 1. Legenda dei simboli utilizzati nell'algoritmo

$\mathcal{D} = \{D_1, \ldots, D_n\}$	Database delle proteine
$\mathcal{L}_i^k$	Pattern frequenti di livello k della proteina D_i
$\mathcal{C}$	Insieme dei pattern candidati
$P^i = (p_1^i, \ldots, p_k^i)$	Pattern di livello k della proteina D_i
$Supp(P)$	Pattern che formano il supporto del pattern P
$Supp_j(P)$	Sottoinsieme dei pattern in $Supp(P)$ che appartengono a D_j
$Prot(P)$	Insieme di proteine cui appartiene almeno un pattern in $Supp(P)$
σ	Soglia di similarità tra pattern

corrispondenti minuscole per denotare le loro regioni; per evidenziare il fatto che un pattern/regione appartiene alla proteina D_i, lo decoreremo con l'apice i. L'algoritmo procede iterativamente generando a ogni passo l'insieme $\mathcal{L}_i^k$ dei pattern frequenti di D_i di livello k via via crescente (riga 2); l'output di un passo rappresenta l'input del passo successivo. Per motivi prestazionali, l'algoritmo lavora su una rappresentazione semplificata dei pattern che consiste in una sequenza di puntatori alle regioni che ne fanno parte. Per questo motivo, dopo aver generato l'insieme $\mathcal{C}$ dei pattern potenzialmente frequenti, è necessario accedere al DB per verificare se effettivamente il vincolo di similarità è soddisfatto (riga 7): la funzione `Simil(`P`,`Q`)` carica P e Q dal DB e ne calcola la similarità, tenendo conto delle caratteristiche di superficie e relazioni spaziali tra le regioni. Si noti a questo proposito come l'accesso al DB sia ottimizzato leggendo, una sola volta, le sole proteine in cui esiste almeno un pattern appartenente a uno dei supporti dei pattern candidati (righe 4-6). Vengono inseriti tra i frequenti i pattern che presentano un supporto di cardinalità superiore alla soglia *minsupp* (riga 10).

Il cuore dell'algoritmo è rappresentato dalla procedura `CandidatePatterns` (Figura 5), che genera i pattern candidati di livello k fondendo coppie di pattern di livello $k-1$. L'algoritmo considera tutte le possibili coppie di pattern in input (riga 2) ed effettua una prima scrematura dei candidati sulla base dei vincoli che questi devono forzatamente soddisfare:

```
1.  CandidatePatterns(L^i_{k-1})
2.  {  C = ∅;
3.     for each P^i, Q^i ∈ L^i_{k-1}; Mergeable(P^i, Q^i) ∧
                                   ∧ | Prot(P^i) ∩ Prot(Q^i) | ≥ minsupp
4.     {  T^i = (p^i_1, ··· , p^i_{k-1}, q^i_{k-1});
5.        Supp(T^i) = {(r^l_1, ··· , r^l_{k-1}, s^l_{k-1}); R^l ∈ Supp(P^i) ∧ S^l ∈ Supp(P^i) ∧
                                   ∧ Mergeable(R^l, S^l) } ;
6.        if (| Supp(T^i) | ≥ minsupp)
7.           C∪ = {T^i};
8.     }
9.     return C;
10. }
```

Figura 5. Algoritmo di generazione dei pattern candidati

```
1.  Mergeable(P^i, Q^i)
2.  {  if (p^i_1 = q^i_1 ∧ ··· ∧ p^i_{k-2} = q^i_{k-2} ∧ p^i_{k-1} < q^i_{k-1})
3.        return TRUE;
4.     else return FALSE;
5.  }
```

Figura 6. Algoritmo di verifica della compatibilità tra pattern

- *Corrispondenza delle regioni*: per ottenere pattern di livello k è necessario fondere due pattern che condividano $k-2$ regioni. Questo vincolo viene verificato nella procedura `Mergeable`, presentata in Figura 6.
  La procedura garantisce inoltre la generazione non ridondante dei pattern imponendo un ordinamento lessicografico tra le regioni.
- *Upper-bound della cardinalità del supporto*: se il numero di proteine che contengono pattern comuni al supporto dei due generatori è minore di *minsupp*, il pattern generato non potrà essere frequente (riga 3). Si noti che il valore così calcolato rappresenta un upper-bound poiché non viene controllata l'effettiva corrispondenza tra i pattern all'interno della proteina. Tale verifica, computazionalmente più costosa, viene effettuata solo per i pattern che hanno superato il primo controllo (righe 5-6).

Le coppie che soddisfano le precedenti proprietà vengono fuse (riga 4) per generare un pattern di livello k e viene calcolato il relativo supporto (riga 5). Si uilizza la rappresentazione esplicita del supporto, tipica degli algoritmi di mining verticali, per evitare di accedere al DB anche in fase di generazione dei candidati e per evitare il calcolo di isomorfismi tra pattern. In questa rappresentazione non ci si limita a indicare l'insieme di proteine in cui compare un pattern simile a quello in esame, ma si specifica direttamente l'insieme di sequenze di regioni che li compone rendendo possibile la verifica della corrispondenza sulla base degli identificatori. Si noti tuttavia che l'accesso al DB è comunque necessario

in `MineFrequentPatterns` poiché la similarità tra i pattern candidati e il loro supporto, garantita a livello $k-1$, non implica quella a livello k.

4 Conclusioni e sviluppi futuri

In questo lavoro è stata descritta una tecnica per l'individuazione di pattern ricorrenti su superfici proteiche, volta a permettere una classificazione significativa delle proteine sulla base delle caratteristiche superficiali. Si sottolinea che l'approccio proposto è valido in generale per tutti i domini in cui occorra eseguire una classificazione di superfici tridimensionali. In particolare, esso trova applicabilità immediata nel settore agroalimentare, in cui si fa sempre più pressante la necessità di determinare le caratteristiche e la qualità dei prodotti in maniera immediata e precisa. Poiché in questo campo le analisi devono essere effettuate utilizzando tecniche non invasive, in grado di estrarre informazioni sulle caratteristiche fondamentali dell'oggetto senza deteriorarlo, risulta estremamente promettente adottare tecniche di analisi di superficie come quelle proposte, capaci di inferire le caratteristiche dell'intero prodotto sulla base di proprietà, quali colore, dimensione, forma, presenza di imperfezioni e stato di idratazione, acquisibili dall'esterno attraverso sensori di vario genere.

Riferimenti bibliografici

1. Sagliano, A., Volpicella, M., Gallerani, R., Ceci, L.: A FastA based compilation of higher plant mitochondrial tRNA genes. NAR **26** (1998) 154–155
2. Alexandrov, N., Luethy, R.: Alignment algorithm for homology modeling and threading. Protein Science **7** (1998) 254–258
3. Srinark, T., Kambhamettu, C.: An approach for 3d segmentation on multiresolution surfaces. In: Proc. Int. Conf. Intelligent Technologies, Chiang Mai, Tailandia (2003)
4. Ferrè, F., Ausiello, G., Zanzoni, A., Helmer-Citterich, M.: Surface: a database of protein surface regions for functional annotation. NAR **32** (2004) 240–244
5. Zhang, Y., Paik, J., Koschan, A., Abidi, M., Gorsich, D.: A simple and efficient algorithm for part decomposition of 3-d triangulated models based on curvature analysis. In: Proc. Int. Conf. Image Processing, Rochester, NY (2002)
6. Kaneta, Y., Shoji, N., Ohkawa, T., Nakamura, H.: A method of comparing protein molecular surface based on normal vectors with attributes and its application to function identification. In: Proc. Joint Conf. Information Science, Research Triangle Park, NC (2002)
7. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: Proc. Int. Conf. Data Mining, San Jose, CA (2001)
8. Fortin, S.: The graph isomorphism problem. Technical report, Dept. of Computer Science, University of Alberta, Canada (1996)
9. Boulicaut, J., Jeudy, B.: Mining free itemsets under constraints. In: Proc. Int. Database Engineering & Applications Symposium, Grenoble, Francia (2001)
10. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc.VLDB, Santiago de Chile, Cile (1994)
11. Zaki, M., Gouda, K.: Fast vertical mining using diffsets. In: Proc. Int. Conf. Knowledge Discovery and Data Mining, Washington, DC (2003)