3 MATERIALI E METODI

3.1 Tecniche standard

3.1.1 Mappe autoorganizzanti di Kohonen

L'architettura di una mappa autoorganizzante di T. Kohonen (Self Organizing Maps Algorithm, *SOMA*) [Kohonen, 1984 ; Kohonen, 1988] è relativamente semplice : uno strato di unità, completamente interconnesse tra di loro, alle quali viene sottoposto un ingresso N-dimensionale, sotto forma di vettore a N componenti.

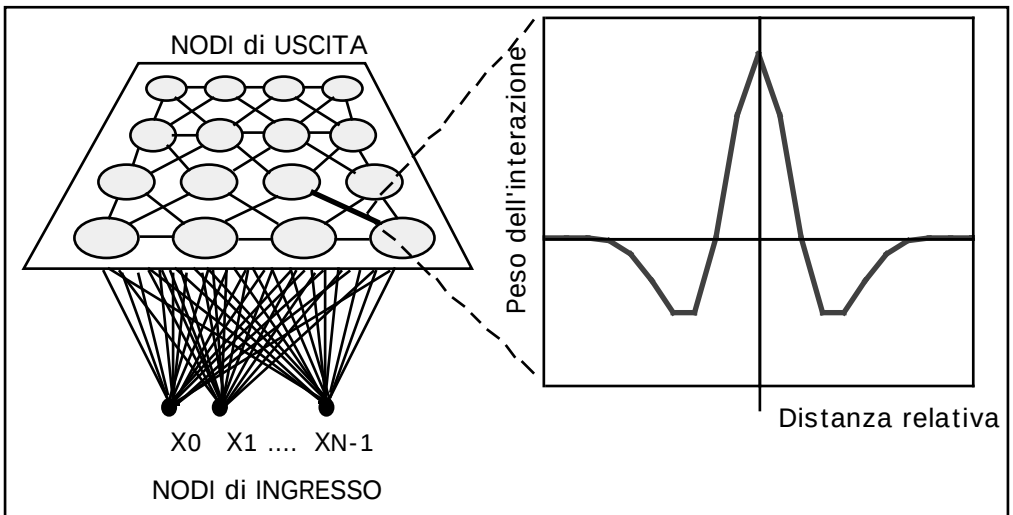


Figura 3.1 Una mappa autoorganizzante di Kohonen.
Ogni nodo di ingresso è connesso, con un peso adattivo, ad ogni nodo di uscita. I nodi di uscita sono interconnessi tra di loro con un peso il cui valore dipende dalla distanza relativa: i nodi vicini interagiscono più fortemente dei nodi lontani. La figura nell'inserto rappresenta una sezione della funzione F(x,y) di interazione laterale delle unità nello strato di uscita. La sua espressione analitica è quella del Laplaciano di una gaussiana.

Il modo nel quale quale viene ottenuta la classificazione rende questo sistema connessionista appartenente alla famiglia delle reti ad *apprendimento* **competitivo* : per ciascuno degli ingressi solo una delle unità di uscita della rete (la *classe*) avrà un valore alto.

Nell'algoritmo *SOMA* i termini di accoppiamento di interazione laterale non sono adattivi, dipendendo dalla sola distanza, e sono, per semplicità, costanti nel tempo. I pesi adattivi

* Con il termine *apprendimento* ci si riferisce a quella serie di regole che vengono seguite per assegnare ad un dato in ingresso (generalmente multivariato) una rappresentazione semplice in uscita (generalmente bivariata). La determinazione delle relazioni variabili $\mathbf{w}_{ij}$ ingresso/uscita avviene in modo adattivo : sono gli esempi utilizzati a determinare le regole di associazione tra un determinato ingresso e la relativa configurazione di risposta del sistema.

w_{ij}, invece, sono quelli che accoppiano ogni unità dello strato di uscita con le componenti del vettore descrivente il dato in ingresso: si assume che tutte le unità di uscita ricevano simultaneamente il dato di ingresso, e che tale grandezza sia a valori continui. Ogni unità avrà quindi un numero di pesi adattivi pari alla dimensione del vettore di ingresso.

La funzione di attivazione $S_{ij}(t)$ dell'unità i,j-sima terrà conto perciò sia dei valori relativi alle componenti del dato di ingresso, mediati dai pesi adattivi, che di quelli dovuti alle attività delle altre unità dello strato di uscita, mediati questi ultimi dalla funzione F(x,y) di feedback laterale

$$S_{ij}(t) = \varphi\left(\eta_{ij}(t) + \sum_{n=-k}^{n=+k} \sum_{m=-K}^{m=+K} F(n,m) S_{i+n,j+m}(t-1)\right) \qquad (3.1)$$

dove il primo termine $\eta_{ij}(t)$ rappresenta il valore di ingresso, mentre il secondo tiene conto degli effetti di feedback laterale : le sommatorie sono da intendersi estese a tutta la rete (-k +k, -K +K). La funzione di attivazione φ è generalmente una sigmoide, ed è importante dire che lo stesso Kohonen non pone particolari vincoli nella scelta della funzione F(x,y) di feedback laterale. E' sufficiente che essa risponda a caratteristiche abbastanza generali, affinchè si ottenga il risultato sperato : una parte centrale eccitatoria ed una periferica inibitoria.

Per quel che riguarda l'equazione di modifica dei pesi variabili (l'equazione di *apprendimento*) , il requisito fondamentale deve essere quello di portare i vettori **w** dei pesi N-dimensionali a riprodurre, ordinatamente e in modo ottimale, l'insieme dei vettori associati ai dati in ingresso. La distanza $d(\mathbf{x},\mathbf{w}_i)$ deve decrescere monotonamente, e la variazione $\delta\mathbf{w}_i$ deve essere tale da verificare

$$[\ \mathrm{grad}_{\mathbf{w}_i}\ d(\mathbf{x},\mathbf{w}_i)\ ]^T \cdot \delta\mathbf{w}_i < 0 \qquad (3.2)$$

L'equazione di apprendimento utilizzata da Kohonen nel *SOMA* [Kohonen,1984] appartiene alla classe generica del tipo

$$d\mathbf{w}/dt \equiv \mathbf{w}' = \Phi(\mathbf{x};\mathbf{w};S)\mathbf{x} - \Theta(\mathbf{x};\mathbf{w};S)\mathbf{w} \qquad (3.3)$$

$\mathbf{x}$ = ingresso = $(x_1 ,,, x_n)^T$
S = uscita = $\mathbf{w}^T\mathbf{x}$
$\mathbf{w}$ = peso = $(w_1 ,,, w_n)^T$

dove Φ e Θ sono funzioni scalari, eventualmente non lineari, dell' ingresso **x**, della risposta S e dello stato stesso del peso **w**. Sostanzialmente le variazioni dei pesi avverranno soltanto nella direzione di **x**, **w** o una loro combinazione lineare. Partendo, poi, dal fatto che abbiamo definito l'attività S delle unità della rete dipendente dal prodotto $\mathbf{w}^T\mathbf{x}$, si possono considerare le funzioni Φ e Θ proprio come $\Phi = \Phi(S)$ e $\Theta = \Theta(S)$, e si può ritenere che le variazioni dei pesi siano proporzionali a tali funzioni di S. Il requisito principale per una legge che descrive un sistema fisico è che ne garantisca la stabilità, e cioè che per **x**(t) limitati rimanga finita la soluzione **w**(t), per ogni t. Si deve, inoltre, considerare non significativa la situazione per la quale $w(t) \rightarrow 0$ per $t \rightarrow \infty$.

Esistono diverse possibili soluzioni del tipo (3.2) [Kohonen,1984 (cap. 4)], non tutte di interesse notevole: ci si limiterà, quindi, ad una discussione, per sommi capi, della particolare soluzione adottata nel modello studiato :

$$d\mathbf{w}/dt \equiv \mathbf{w}' = \alpha S \mathbf{x} - \beta S \mathbf{w} = \alpha \mathbf{x}\mathbf{x}^T\mathbf{w} - \beta \mathbf{w}\mathbf{w}^T\mathbf{x} \qquad \alpha, \beta > 0 \tag{3.4}$$

(n.b. : il prodotto $\mathbf{w}\mathbf{w}^T$ è una matrice n x n)

Se chiamiamo **X** il valore aspettato di **x** condizionato da **w**, e cioè

$$E\{ \mathbf{x} \mid \mathbf{w} \} = \mathbf{X}$$

e C_{xx} l'elemento della matrice di correlazione di **x**, ottenuto da

$$E\{ \mathbf{x}\mathbf{x}^T \mid \mathbf{w} \} = C_{xx}$$

allora otteniamo una espressione della (3.3) come equazione differenziale di Bernoulli di secondo grado

$$\langle \mathbf{w}' \rangle = \alpha C_{xx}\mathbf{w} - \beta(\mathbf{X}^T\mathbf{w})\mathbf{w} \tag{3.5}$$

Come si vede, una possibile soluzione stazionaria di $\langle \mathbf{w}' \rangle = 0$ è quella che si ottiene per $\mathbf{w}^* = 0$. Si dimostra anche che un qualsiasi autovettore della matrice di correlazione C_{xx} rappresenta un punto fisso del sistema : se $\mathbf{c}_i$ è un autovettore con autovalore λ_i, allora l'eventuale soluzione sarà $\mathbf{w}^* = k\mathbf{c}_i$, con k costante scalare. Infatti

$$C_{xx}\,\mathbf{c}_i = \lambda_i \mathbf{c}_i$$

$$0 = k\alpha\lambda_i\mathbf{c}_i - k^2\beta(\mathbf{X}^T\mathbf{w})\mathbf{c}_i$$

$$k = \frac{\alpha\lambda_i}{\beta(\mathbf{X}^T\mathbf{w})}$$

e quindi

$$\mathbf{w}^{*} = \frac{\mathbf{c}_i \, \alpha\lambda_i}{\beta(\mathbf{X}^T\mathbf{w})} \tag{3.6}$$

Si dimostra pure [Kohonen,1984 (cap. 4)], però, che non tutti i punti fissi rappresentano soluzioni stabili ; comunque, la "traiettoria" $\mathbf{w}(t)$ sarà rallentata in corrispondenza di un punto fisso e, se il prodotto scalare tra l'autovettore $\mathbf{c}_{max}$ associato all'autovalore massimo e il vettore $\mathbf{w}$ si mantiene positivo per ogni t, si dimostra che la soluzione tenderà a convergere verso l'autovettore di C_{xx} avente il massimo autovalore. In questo modello, infine, esiste la probabilità non nulla che $\mathbf{w}(t)$ converga a zero anche per ingressi non nulli. Da quanto si è detto si ricava la seguente proposizione

> Se gli ingressi x_i sono variabili stocastiche con proprietà statistiche stazionarie, allora i valori $\mathbf{w}_i$, in accordo con l'equazione (3.6), convergeranno a valori asintotici tali che il vettore $\mathbf{w}$ rappresenterà l'autovettore di C_{xx} associato al maggior autovalore.

Studi sullo stato stazionario del modello di Kohonen sono stati sviluppati da H. Ritter e K. Schulten, nei quali si ottiene anche una espressione esplicita del fattore di ingrandimento locale della mappa (e cioè della proprietà della mappa di assegnare alla decodifica del dato di ingresso un numero di unità legato alla importanza statistica di quest'ultimo) [Ritter & Schulten, 1986]. Sempre Ritter e Schulten ne hanno studiato le proprietà di convergenza e le fluttuazioni dalla situazione di equilibrio, descrivendo il processo di apprendimento per mezzo di una equivalente equazione di Fokker-Planck [Ritter & Schulten, 1988].

L'effetto dell'applicazione di tale algoritmo di classificazione ad un set set di dati multivariati è la progressiva organizzazione dei valori di attivazione $S_{ij}(t)$ delle unità di uscita della rete in "bolle", e cioè in raggruppamenti di neuroni attivi intorno al più attivo di tutti * (per quel determinato ingresso $\mathbf{x}_i(t)$).

Una precisazione da fare è che, nel processo di auto-organizzazione appena descritto, non sono le unità a spostarsi sullo strato di uscita (Fig. 3.1) : sono le loro attività che diventano spazialmente correlate. L'effetto di tale organizzazione sarà, quindi, visibile come bolla di attività nello spazio fisico bidimensionale (Fig. 3.2).

Si è poi visto che la dimensione di tale bolla è in qualche modo legata al rapporto E/I tra la parte eccitatoria e quella inibitoria della F(x,y), e, più precisamente, la bolla avrà il raggio tanto minore quanto maggiore sarà il contributo inibitorio. Il verificarsi del meccanismo di "clusterizzazione" è abbastanza dipendente dalla scelta di alcuni parametri, quali appunto il rapporto E/I, la larghezza della F(x,y) e la sua forma : essi sono pertanto da determinarsi empiricamente.

---

* In linea con le caratteristiche fondamentali degli algoritmi ad apprendimento competitivo.

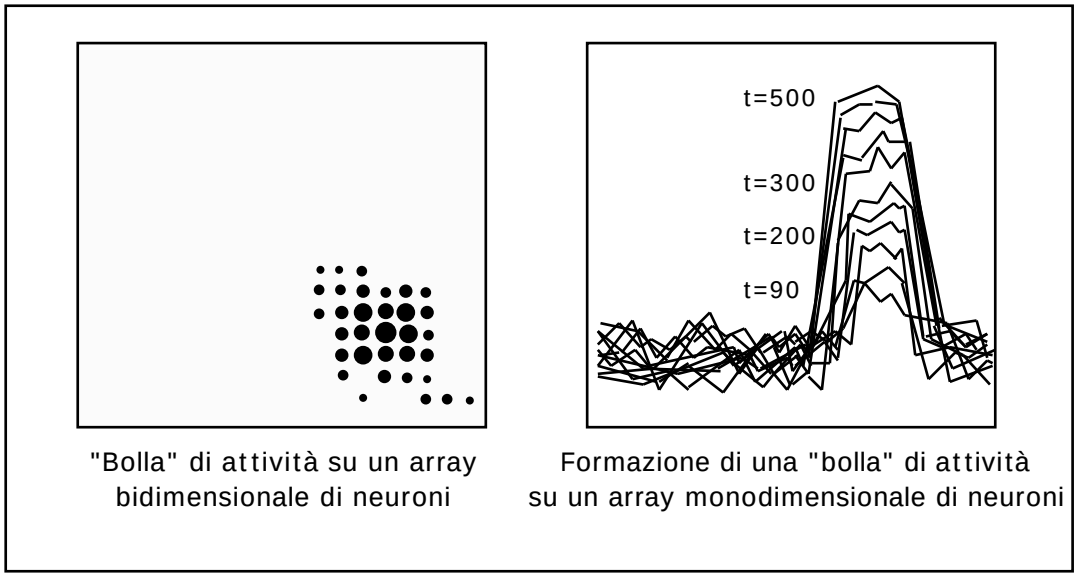


Figura 3.2 - La localizzazione della risposta ad un determinato ingresso e la conservazione di topologia in una mappa auto-organizzante.

Tornando alla discussione dell'algoritmo *SOMA*, è possibile introdurre delle varianti, a partire dalla (3.4), che permettono di realizzare un algoritmo semplificato. Una volta, infatti, osservata la capacità del modello di creare delle rappresentazioni auto-organizzate, si è passato allo studio di espedienti computazionali che semplificassero l'algoritmo, mantenendo inalterate, però, le peculiarità del "clustering" e della conservazione della topologia, e che, soprattutto, consentissero di ottenere una rappresentazione spazialmente ordinata con una lunghezza di correlazione significativamente grande.

L'equazione di attivazione (3.1) tende, come si è visto, a stabilizzare l'attività S_j delle unità ad un valore "alto" o "basso", in funzione della propria risposta al dato in ingresso e dell' attività delle unità adiacenti. In pratica si verifica che, una volta assegnati i valori iniziali (generalmente casuali) dei pesi di connessione, le unità della rete che formeranno la *bolla* ad attività alta sono quelle che - insieme con i vicini in un intorno di dimensioni determinate dalla forma della F(x,y) di feedback laterale - hanno una attività di gruppo massima. Ciò permette di fare delle semplificazioni : si può imporre che la *bolla* si formi intorno al neurone che, da solo, ha attività iniziale massima. Una ulteriore semplificazione consiste nel considerare tale attività come funzione di un criterio di similitudine nello spazio vettoriale n-dimensionale. La scelta più semplice, ma non per questo riduttiva, è quella della distanza euclidea, adottata in molti modelli ad apprendimento competitivo : con essa si può calcolare il "matching score" tra i due vettori senza che essi debbano essere normalizzati. L'unità che avrà quindi attività iniziale massima sarà quella che misurerà distanza euclidea minima e, cioè, che realizzerà, con il proprio vettore dei pesi **w**, il "best match" con il vettore di ingresso **x**

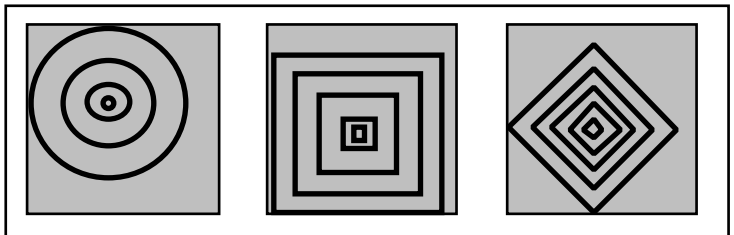
$$\| \mathbf{x} - \mathbf{w}_c \| = \min_i \| \mathbf{x} - \mathbf{w}_i \| \quad (3.7)$$

Dopo aver individuato l'unità "best match", si avvia, in un suo intorno N_c, il processo di variazione dei pesi : la correlazione a largo raggio che si osserva nelle mappe autoorganizzanti dipende proprio dalla scelta di tale intorno. Le unità che subiranno un adattamento dei pesi, alla presentazione di un ingresso, saranno quelle che misureranno una distanza (ora la distanza non è più nello spazio parametrico N-dimensionale dei pesi, ma nello spazio fisico 2-dimensionale dello strato di uscita) dell'unità "best match" minore di un raggio di interazione R(t), decrescente nel tempo. Avremo così un intorno $N_c(t)$ la cui dimensione iniziale dovrà essere tale da coinvolgere, nei primi processi, tutte le unità dello strato di uscita. L'andamento temporale può essere rappresentato da una funzione lineare, esponenziale, etc. : la scelta, pure in questo caso, non è estremamente vincolante. Eccone alcuni esempi

$$R(t) = \max\left\{ R_0 + \frac{(R_{\min} - R_0)t}{T_0} \; , \; R_{\min} \right\} \quad (3.8a)$$

$$R(t) = \max\left\{ R_0 \frac{-t}{T_0} \; , \; R_{\min} \right\} \quad (3.8b)$$

Il valore di T_0 stabilisce una costante di tempo del processo di *clusterizzazione* che verrà discussa tra breve. Il valore di R_{min} stabilisce il raggio minimo di interazione, in genere unitario o nullo, che si vuole mantenere tra le unità nella fase asintotica dell'apprendimento. In alcuni casi è importante mantenere un raggio minimo di interazione non minore di due unità, allo scopo di garantire una certa plasticità anche nella fase avviata del programma di apprendimento e permettere alle rete di riassestarsi, qualora si verificassero mutazioni significative nell'insieme di dati da classificare. La scelta della forma geometrica dell'intorno è, anche questa, abbastanza arbitraria e condizionata da fattori di semplicità computazionale. Eccone alcune possibili



Esempi di intorni temporalmente decrescenti

Nelle applicazioni realizzate nel corso di questo lavoro, la scelta è caduta sull'intorno circolare per raggi di interazione grandi (per motivi di isotropia), passando poi all'intorno quadrato (assai più maneggevole computazionalmente) al rimpiccolirsi del raggio.

Anche per l'equazione di apprendimento possono essere fatte delle semplificazioni. Partendo dalla

$$d\mathbf{w}/dt = \alpha\, S\, \mathbf{x} - \beta(S)\, \mathbf{w}$$

possiamo, nel limite di saturazione per il quale l'attività S si stabilizza su valori alti (dentro la bolla) o bassi (fuori dalla bolla), assegnare alla funzione non lineare monotona $\beta(S)$ degli analoghi valori di saturazione. Riscalando le variabili **x** e **w** abbiamo la possibilità di definire $S \in \{0,1\}$ e $\beta(S) \in \{0,\alpha\}$, e quindi di riscrivere l'equazione di apprendimento come

$$d\mathbf{w}/dt = \alpha\,(\mathbf{x} - \mathbf{w}) \qquad (3.9a)$$

se $S = 1$ e $\beta = \alpha$ (dentro la bolla)

$$d\mathbf{w}/dt = 0 \qquad (3.9b)$$

se $S = 0$ e $\beta = 0$ (fuori dalla bolla)

Si è osservato, inoltre, che per avere buoni risultati nell'auto-organizzazione, dobbiamo fare in modo che anche il guadagno plastico α sia una funzione monotona decrescente nel tempo, anche questa determinata sulla base di prove empiriche. Unico vincolo, come gia si è detto, è che sia compresa tra 0 e 1. Una scelta tra le più comuni è

$$\alpha(t) = \alpha_0\,(1 - t / T_0) \qquad (3.10)$$

$\alpha_0 = 0.1 \div 0.9$

Nel corso delle simulazioni si è visto che i migliori risultati si sono ottenuti con valori di $\alpha_0 \sim 0.1$. Il valore T_0, trovato anche nella (3.8) che descrive l'andamento temporale del raggio R(t) di interazione, regola la durata della fase di prima organizzazione della rete, durante la quale il raggio decrescerà dal valore di massimo ricoprimento R_0 al valore R_{min} di interazione con i primi vicini. Si è notato, sempre empiricamente, che migliori risultati si ottengono se, una volta raggiunto il raggio minimo, si continua il programma di apprendimento per un tempo $T_1 \geq 10T_0$. In questa seconda fase, nella quale ogni gruppo di unità che risponde ad un certo ingresso si specializza, il termine di guadagno plastico α viene mantenuto costante, o al più leggermente decrescente nel tempo, intorno ad un valore che, nei casi studiati, è di $\alpha \sim 0.02 \div 0.06$.

Nel caso pratico, il termine temporale t (così come i termini T_1 e T_0) è, in realtà, un contatore che misura la maturazione della rete in base al numero progressivo di dati in ingresso elaborati : assume pertanto valori discreti. Risultati accettabili nel processo di auto-

organizzazione si ottengono se il programma di apprendimento prevede complessivamente almeno qualche migliaio di ingressi presentati (anche ricorsivamente) alla rete (e cioè $T_1+T_0 \sim 10^3 \div 10^4$). Tali valori dipendono dalla dimensione della rete stessa.

Passando alle differenze finite si ottiene, così, la forma ultima delle equazioni di apprendimento semplificate

$$\| \mathbf{x} - \mathbf{w}_c \| = \min_i \| \mathbf{x} - \mathbf{w}_i \|$$

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t)[\, \mathbf{x}(t) - \mathbf{w}(t) \,] \qquad \text{per i} \quad N_c(t) \tag{3.11a}$$

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) \qquad \text{per i} \quad Nc(t) \tag{3.11b}$$

Una alternativa possibile al sistema appena visto prevede l'introduzione di una funzione scalare $H_{ci} = H_{ci}(t)$, che modula la variazione dei pesi dell'unità i-esima, in funzione della distanza dall' unità c-esima di "best match"

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + H_{ci}(t)\, [\, \mathbf{x}(t) - \mathbf{w}(t) \,] \tag{3.12}$$

Una delle funzioni che possono descrivere la $H_{ci}(t)$ di modulazione del guadagno può essere la

$$H_{ci}(t) = H_0 \exp\left(-\frac{\|r_i - r_c\|^2}{s^2(t)}\right) \tag{3.13}$$

che descrive una curva a "campana", centrata in r_c, la cui larghezza è controllata dalla funzione s(t), decrescente nel tempo.

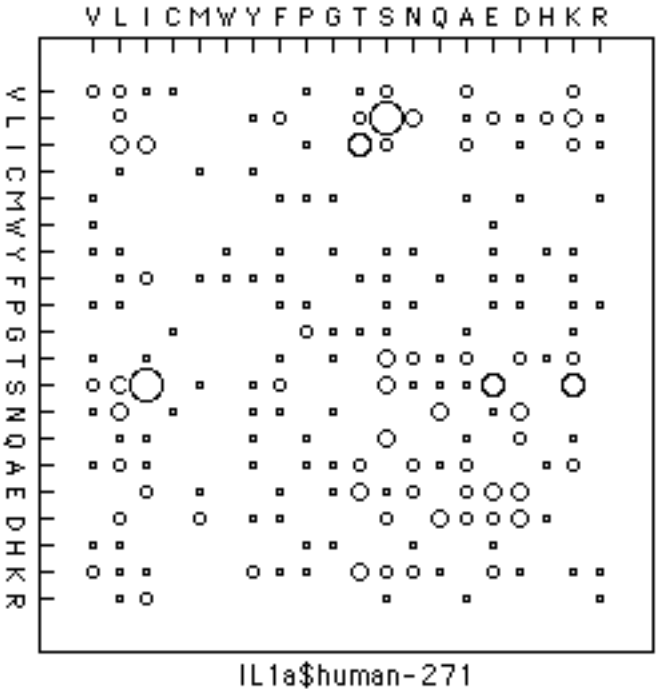
Il diagramma a blocchi dell'algoritmo *SOMA* è riportato nello Schema 3.1.

I risultati che si ottengono utilizzando l'algoritmo semplificato appena descritto, sia per la (3.12) che la (3.13) (più onerosa dal punto di vista computazionale), evidenziano le proprietà già note di questo modello : il mapping a conservazione di topologia, la riduzione di dimensionalità e la selezione delle dimensioni del segnale di ingresso più rilevanti dal punto di vista informativo (quelle a massima varianza).

Schema 3.1

Algoritmo *SOMA* di Kohonen

L'algoritmo*SOMA*, è, per le proprietà appena descritte, ben indicato per risolvere il problema della mappatura di strutture "multidimensionali" come le sequenze proteiche. Il problema maggiore che nasce con l'utilizzo di questo metodo è quello legato alla necessità di disporre di patterns (strutture primarie di proteine) codificati in modo tale da essere rappresentati da vettori, tutti con lo stesso numero di componenti. E' invece noto che le proteine possono essere formate da catene di aminoacidi diversamente lunghe (indicativamente, da 50 a 2000 residui). Una codifica rispondente a tali specifiche è stata proposta da E. A. Ferràn e P. Ferrara [Ferràn & Ferrara, 1991]. Essa consiste, sostanzialmente, in una trasformazione che tiene conto delle frequenze di composizione bipeptidica di ogni proteina: viene costruita una matrice $\mathbf{X}_{20*20}$ di 400 componenti i cui elementi corrispondono alle frequenze relative di tutte le possibili coppie ordinate di aminoacidi della sequenza corrispondente. Ciascun vettore di 400 componenti, codificante ciascuna proteina, verrà utilizzato, una volta normalizzato, per la costituzione dell'insieme di patterns di addestramento della rete neuronale *SOMA*. Nel caso particolare i due autori hanno utilizzato un insieme di dieci precursori di interleuchina-1 , relativi a quattro specie differenti (umana, bovina, del topo e del coniglio) e di tre tipi differenti (**a**lpha,**b**eta e **r**eceptor). Nella figura seguente è raffigurato il risultato della codifica secondo il metodo della frequenza dipeptidica per il precursore dell'interleuchina-1 **a**lpha umana.



Rappresentazione grafica della matrice pattern associata alla struttura primaria della proteina umana interleuchina-1 alpha umana (precursore). Il raggio del cerchio in ogni posizione (x,y) della matrice è proporzionale alla frequenza della coppia xy di aminoacidi nella sequenza. Il cerchio più grande corrisponde a 5 coppie di bipeptidi (LS e SI).

Struttura primaria :

MAKVPDMFED LKNCYSENEE DSSSIDHLSL NQKSFYHVSY GPLHEGCMDQ SVSLSISETS
KTSKLTFKES MVVVATNGKV LKKRRLSLSQ SITDDDLEAI ANDSEEEIIK PRSAPFSFLS
NVKYNFMRII KYEFILNDAL NQSIIRANDQ YLTAAALHNL DEAVKFDMGA YKSSKDDAKI
TVILRISKTQ LYVTAQDEDQ PVLLKEMPEI PKTITGSETN LLFFWETHGT KNYFTSVAHP
NLFIATKQDY WVCLAGGPPS ITDFQILENQ A

Il processo di apprendimento e formazione della mappa segue le classiche fasi (già descritte) dell'algoritmo di autoorganizzazione di Kohonen. L'obiettivo è quello di organizzare le sequenze primarie in gruppi ("clusters") o famiglie in base alle omologie di sequenza. La scelta di un algoritmo non supervisionato è dovuta al fatto che non è nota a priori la composizione ed il numero di questi "clusters". Le mappe topologiche ottenute potrebbero essere utili per organizzare estesi data-base di proteine e per classificare nuove sequenze.

C'è da sottolineare che i risultati, assai interessanti, esposti da Ferràn e Ferrara nel loro articolo sono stati ottenuti (e replicati nel corso di questo lavoro) utilizzando un gruppo di proteine già appartenenti alla stessa classe. Le mappe (4*4) ottenute con l'algoritmo *SOMA* hanno evidenziato una separazione abbastanza netta tra i gruppi **a**lpha, **b**eta ed **r**eceptor, indipendentemente dalla specie. La figura seguente mostra schematicamente alcune delle mappe ottenute dai due autori per diversi valori dei parametri di apprendimento della rete (è interessante vedere che, al variare di tali parametri varia anche la risoluzione della classificazione).

Il valore "Hscore" riportato è sostanzialmente un "punteggio di omologia" per la particolare mappa ottenuto con il calcolo della percentuale di identità tra ciascuna coppia di proteine per mezzo dell'algoritmo di Needleman-Wunsch [Needleman & Wunsch, 1970].

2r			
4a			2b
			2b

Hscore = 202.33

4a			
4b			2r

Hscore = 168.24

2a		b	b
a		b	
a	b		
		r	r

Hscore = 77.02

Mappe topologiche di dieci proteine ottenute per differenti valori dei parametri di apprendimento. Viene solo fornito il numero di proteine p ogni famiglia (a : IL1a ; b : IL1b ; r : IL1r).

(modificata da Ferràn & Ferrara, 1991)

Nel corso della sperimentazione per sondare l'efficienza della codifica proposta da Ferràn e Ferrara per l'algoritmo di Kohonen si è verificato che la semplice informazione sulla frequenza dipeptidica non sempre conduce alla produzione di mappe di Kohonen soddisfacenti (utilizzando insiemi di proteine delle quali sono note le caratteristiche di omologia). Questo perchè, su stessa ammissione dei due autori, la "finestra" di analisi sulla proteina (larga soltanto due peptidi consecutivi) non è sufficientemente ricca di informazione. Tale codifica rimane così di carattere meramente composizionale (composizione di dipeptidi), perdendo l'informazione

associata alla posizione dei singoli aminoacidi nella sequenza, dato sicuramente rilevante nel processo di *folding* della proteina.

Si sono presentate quindi due alternative: la prima, più naturale, è quella dell'ampliamento della finestra a raggruppamenti tri- o tetrapeptidici, ma ciò ha come conseguenza la necessità di utilizzare vettori di codifica rispettivamente di 20^3=8000 o 20^4=160000 componenti, e quindi l'enorme allungamento dei tempi di calcolo; la seconda consiste nel conservare l'informazione posizionale e, al tempo stesso, codificare gli aminoacidi con grandezze fisico-chimiche piuttosto che con simboli : ad ogni proteina viene associato, in questo modo, un *profilo* numerico. Quest'ultima soluzione presenta però il problema della differente lunghezza dei profili descriventi le proteine, dipendente appunto dal numero di residui. Nel corso di questo lavoro, per ovviare a questo problema, è stata elaborata una procedura originale di equalizzazione delle lunghezze, descritta nel § 3.2.1.

3.1.2 Analisi delle componenti principali.

L'analisi delle componenti principali (PCA) è un metodo con il quale è possibile rappresentare insiemi di oggetti, caratterizzati dall'avere una descrizione multivariata, in base a un *nuovo* insieme di variabili, le **componenti principali**, tra loro incorrelate [Kendall, 1958] [Gower, 1966]. Tale metodo prende anche il nome di *Singular Value Decomposition* [Henry & Hofrichter, 1992].

Si considerino n oggetti $\mathbf{x}_i$, ciascuno descritto da p variabili ($\mathbf{x}_i = x_{i1}, x_{i2}, ..., x_{ip}$), non necessariamente indipendenti. Per il calcolo delle componenti principali si procede innanzitutto ad una standardizzazione X_{ij} a media nulla e varianza unitaria delle variabili x_{ij} :

$$X_{ij} = \frac{x_{ij} - \bar{x}_j}{\sqrt{\sigma_j}}$$

dove $\bar{x}_j$ e σ_j sono rispettivamente il valore medio e la varianza della variabile j-esima su tutti gli n oggetti $\mathbf{x}_i$. Consideriamo quindi una combinazione lineare delle variabili standardizzate per ogni elemento dell'insieme

$$z_i = a_1 X_{i1} + a_2 X_{i2} + ... + a_p X_{ip} = \sum_{k=1}^{p} a_i X_{ik}$$

I coefficienti a_i della somma vengono determinati in modo tale da massimizzare la correlazione tra le variabili standardizzate X_{ij} e le loro combinazioni lineari z_i , espressa da

$$Q = \sum_{j=1}^{p} \left(\sum_{i=1}^{n} z_i X_{ij} \right)^2$$

a condizione che la somma $\sum_{i=1}^{n} z_i^2$ sia uguale ad uno. Il problema è equivalente a quello della massimizzazione della grandezza $Q - \lambda \sum_{i=1}^{n} z_i^2$ rispetto ai coefficienti a_i, dove λ è un moltiplicatore di Lagrange, e il risultato coincide con l'individuazione dei p autovalori $\lambda^{(1)} \ldots \lambda^{(p)}$ e dei corrispondenti autovettori $\mathbf{a}^{(1)}$, ..., $\mathbf{a}^{(p)}$ della matrice nXn di correlazione tra gli n oggetti $\mathbf{x}_i$:

$$\begin{pmatrix} r_{11} & \cdots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{p1} & \cdots & r_{pp} \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_p \end{pmatrix} = \lambda \begin{pmatrix} a_1 \\ \vdots \\ a_p \end{pmatrix}$$

dove i termini r_{ij} sono i coefficienti di correlazione calcolati secondo la formula di Pearson

$$r_{ij} = \frac{\sum_{k=1}^{n} (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^{n} (x_{ik} - \bar{x}_i)^2 \sum_{k=1}^{n} (x_{jk} - \bar{x}_j)^2}} \quad i,j = 1,...,p$$

Gli autovalori $\lambda^{(k)}$ (k=1,...,p) corrispondono ai valori di correlazione $Q^{(k)}$ con le p componenti degli autovettori $\mathbf{a}^{(k)}$. Inoltre gli autovalori, e così anche gli autovettori, sono ordinati in modo tale rappresentare percentuali di variabilità crescenti : $z^{(1)}$ rappresenta la quantità di variabilità maggiore, $z^{(2)}$ la seconda maggiore , e via dicendo.

Con questo metodo è possibile quindi rappresentare oggetti multivariati con un numero ridotto di nuove variabili, senza per questo perdere percentuali significative di informazione. Nella figura seguente è riportato un semplice esempio di come sia possibile rappresentare una grandezza bivariata con una sola variabile, conservando comunque una quantità di informazione soddisfacente.

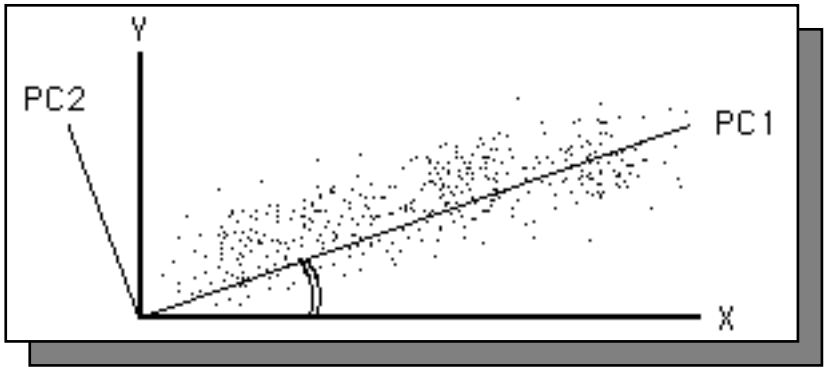
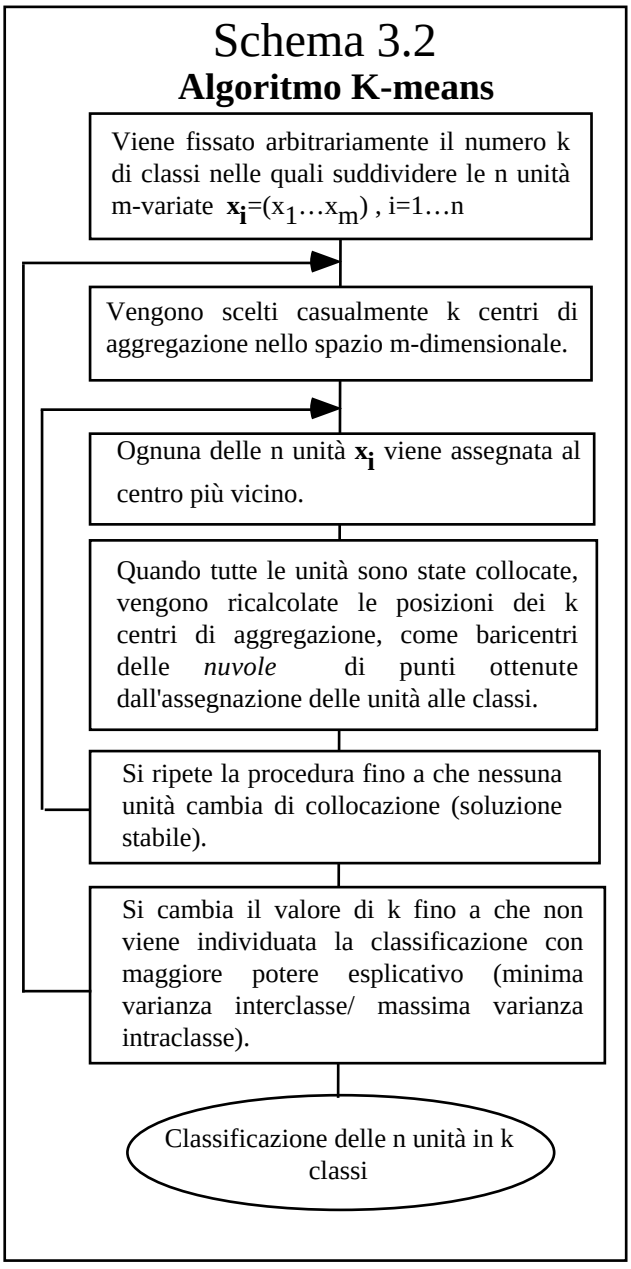


Figura 3.3 L'analisi delle componenti principali.
In figura è rappresentata una distribuzione bivariata di punti, espressi ciascuno da una coppia di variabili (x,y). Grazie all'analisi delle componenti principali (PCA) è possibile rappresentare la distribuzione bivariata con sufficiente precisione (e senza una significativa perdita di informazione) con una sola variabile lungo il nuovo asse PC1 (che descrive la maggiore quantità di variabilità dell'insieme di punti nelle coordinate originali). Il coseno dell'angolo tra l'asse PC1 e gli assi x, y rappresenta il *loading* della prima componente principale con ciascuna delle variabili originali.

3.1.3 L'analisi dei *clusters*

Con tale nome vengono denominate un gruppo di procedure finalizzate alla identificazione di eventuali aggregazioni (*clusters*), o classi, presenti in un insieme di dati multivariati. Le metodiche di analisi dei *clusters* sono differenti, ma tutte sono caratterizzate da un obiettivo comune, che è quello di suddivedere un insieme di unità (eventualmente multivariate) in classi in modo tale tutti gli individui appartenenti ad una determinata classe si *somiglino* tra loro (in base ad una scelta, spesso cruciale, di un tipo di distanza) e siano, al tempo stesso, differenti da quelli delle alte classi.

Schema 3.2
Algoritmo K-means



In altri termini, si trova la classificazione per la quale viene contemporaneamente minimizzata la varianza *intraclasse* e massimizzata qualla *interclasse* (eventualmente minimizzando il rapporto tra le due). Per semplicità verrà descritta qualitativamente solo una di queste tecniche, nota con il nome di *"k-means"*, la cui applicazione, molto diffusa, è di semplice comprensione e realizzazione (vedi Schema 3.2).

Le classi costruite con la procedura k-means risulteranno massimamente compatte e massimamente separate. Al crescere di k, partendo almeno da k=2, si passa dalla individuazione di raggruppamenti sulla grande scala alla determinazione di strutture più fini, arrivando, quando k uguaglia il numero n di unità, alla discriminazione della singola unità (con perdita di significato). Se non si dovesse trovare un minimo, al variare di k, del rapporto tra varianze di interclasse e intraclasse, ciò

significherebbe che nell'insieme di dati non è possibile ritrovare alcuna clusterizzazione.

Nella Figura 3.4 viene riportato un esempio di classificazione di unità 2-variate (un insieme di punti (x,y) nel piano) per differenti valori di k (k=2 e k=4, per i quali si ottengono i migliori risultati di classificazione).

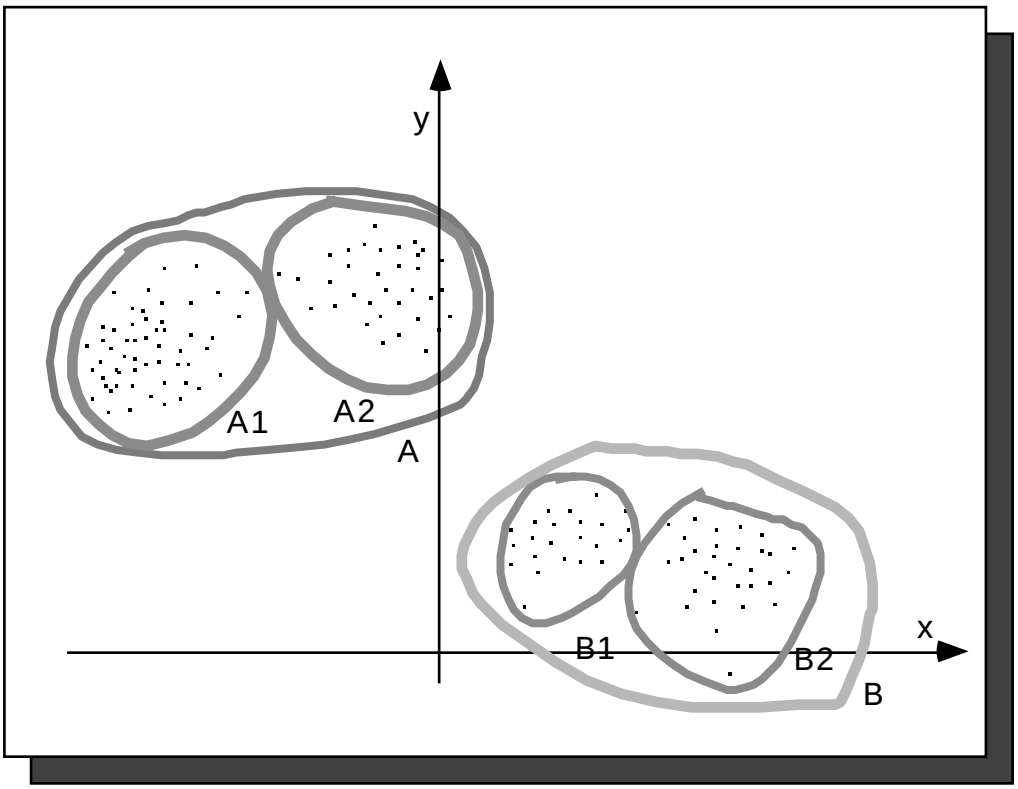


Figura 3.4 - La scelta del numero di clusters
Nella figura è riportato un insieme di n punti (x,y) 2-variati. Per i valori di k=2 e k=4 è possibile trovare delle classi (rispettivamente, A-B, e A1-A2-B1-B2) con le quali si ottiene due efficienti partizione dell'insieme (rispettivamente, a grande e piccola scala).

Una possibile interpretazione del significato di *cluster* è quella di *attrattore* di un sistema dinamico, attorno al quale si trovano concentrate intere famiglie di stati. Inoltre, una importante peculiarità comune a tutte le tecniche di *cluster analysis* è quella di individuare delle classificazioni in insiemi di dati senza la necessità di alcuna conoscenza a priori (si pensi alle mappe autoorganizzanti non supervisionate dell'algoritmo *SOMA* descritte prima nel § 3.1.1). Le classi "naturali" nelle quali un insieme di dati multivariati può essere organizzato in modo ordinato emergono spontaneamente dall'analisi della topologia dello spazio multidimensionale nel quale sono descritte. Anche in questo caso svolgono un ruolo cruciale il criterio di codifica (e quindi la selezione delle variabili effettivamente informative) e la definizione di un criterio di distanza, che permetta di quantificare una eventuale *somiglianza* tra dati differenti.

3.2 Tecniche appositamente sviluppate

3.2.1 La codifica numerica delle strutture primarie e l'algoritmo del "Letto di Procuste".

Come si è accennato nei paragrafi precedenti, l'utilizzo dell'algoritmo di classificazione *SOMA* di Kohonen impone una restrizione: tutti i vettori descriventi i dati in ingresso devono avere lo stesso numero di elementi. Questa limitazione è cruciale se teniamo conto del fatto che gli *oggetti* da classificare sono, in questo caso, strutture primarie di proteine.

Alternativamente alla soluzione proposta in letteratura [Ferràn & Ferrara, 1991] basata sulla codifica delle strutture primarie con vettori contenenti le 20x20 frequenze dipeptidiche, si propone innanzitutto l'utilizzo di grandezze fisico-chimiche a descrizione delle caratteristiche dei singoli aminoacidi ; si ritiene, poi, cruciale la conservazione dell'informazione sulla posizione dei singoli aminoacidi nella sequenza.

Per quel che riguarda il primo punto è stata utilizzata una tabella di valori, pubblicata in letteratura [Schneider & Wrede, 1993] e riportata qui di seguito, relativa a sette grandezze fisico-chimiche importanti per la codifica degli aminoacidi :

Tabella 3.1
Proprietà fisico-chimiche dei 20 aminoacidi naturali.

	hydropho	volume	surfarea	hydrophil	bulkiness	refractiv	polarity	
Ala	1.60	88.60	115.00	-0.50	11.50	4.34	0.00	**A**
Arg	-12.30	173.40	225.00	3.00	14.28	26.66	52.00	**R**
Asn	-4.80	117.70	160.00	0.20	11.68	12.00	49.70	**N**
Asp	-9.20	111.10	150.00	3.00	12.82	13.28	3.38	**D**
Cys	2.00	108.50	135.00	-1.00	13.46	35.77	1.48	**C**
Gln	-4.10	143.90	180.00	0.20	13.57	17.26	49.90	**Q**
Glu	-8.20	138.40	190.00	3.00	14.45	17.56	3.53	**E**
Gly	1.00	60.10	75.00	0.00	3.40	0.00	0.00	**G**
His	-3.00	153.20	195.00	-0.50	13.69	21.81	51.60	**H**
Ile	3.10	166.70	175.00	-1.80	21.40	19.06	0.13	**I**
Leu	2.80	166.70	170.00	-1.80	21.40	18.78	0.13	**L**
Lys	-8.80	168.60	200.00	3.00	15.71	21.29	49.50	**K**
Met	3.40	162.90	185.00	-1.30	16.25	21.64	1.43	**M**
Phe	3.70	189.90	210.00	-2.50	19.80	29.40	0.35	**F**
Pro	-0.20	122.70	145.00	0.00	17.43	10.93	1.58	**P**
Ser	0.60	89.00	115.00	0.30	9.47	6.35	1.67	**S**
Thr	1.20	116.10	140.00	-0.40	15.77	11.01	1.66	**T**
Trp	1.90	227.80	255.00	-3.40	21.67	42.53	2.10	**W**
Tyr	-0.70	193.60	230.00	-2.30	18.03	31.53	1.61	**Y**
Val	2.60	140.00	155.00	-1.50	21.57	13.92	0.13	**V**

La scelta, nella codifica, di una grandezza piuttosto che un'altra assume quindi un ruolo determinante per l'ottenimento di una *descrizione* della struttura primaria che contenga la

massima quantità di informazione e la minima ridondanza (per contenere il costo computazionale e per enfatizzare le differenze significative tra le proteine).

A tal fine è stata utilizzata la tecnica dell'Analisi delle Componenti Principali (già descritta nel paragrafo § 3.1.2) sull'insieme delle sette proprietà fisico-chimiche. Il risultato, riportato nella tabella seguente, mostra che con le sole prime due componenti principali si è in grado di descrivere più dell' 84% della variabilità associata all'insieme originale, ottenendo così una significativa diminuzione della ridondanza e, al tempo stesso, una compattazione della codifica.

Tabella 3.2
Analisi delle componenti principali sulle proprietà fisico-chimiche riportate nella Tabella 3.1
(Nella prima riga, con EV(%) è riportato il valore percentuale di variabilità spiegata da ciascuna componente)

		PC-1	PC-2	PC-3	PC-4	PC-5	PC-6	PC-7
	EV(%)	*50.04*	*34.73*	*7.43*	*5.29*	*1.90*	*0.47*	*0.14*
A	***Ala***	-1.13	-0.91	-0.12	0.28	-0.47	-0.40	0.95
R	***Arg***	0.21	2.35	0.18	-0.32	0.00	-0.04	-0.48
N	***Asn***	-0.67	0.79	-1.58	0.77	0.11	-1.58	0.44
D	***Asp***	-0.95	0.80	1.96	-0.65	0.12	-1.35	-0.99
C	***Cys***	-0.01	-0.66	-0.53	-2.55	3.19	0.08	0.21
Q	***Gln***	-0.16	0.90	-1.44	0.71	0.13	-0.14	0.05
E	***Glu***	-0.34	0.97	2.08	-0.67	-0.68	0.23	1.81
G	***Gly***	-2.12	-0.96	-0.87	-1.06	-1.38	0.35	-2.06
H	***His***	0.18	0.85	-1.88	0.37	0.02	-0.13	0.99
I	***Ile***	0.72	-0.81	0.42	1.13	0.33	0.30	-0.85
L	***Leu***	0.68	-0.81	0.44	1.16	0.43	0.11	-1.83
K	***Lys***	0.03	1.86	0.21	0.60	0.59	2.10	-1.13
M	***Met***	0.50	-0.61	-0.13	-0.23	-0.84	2.17	1.05
F	***Phe***	1.29	-0.66	-0.16	-0.23	-0.52	0.87	0.28
P	***Pro***	-0.34	-0.48	0.82	0.91	0.49	-0.31	0.38
S	***Ser***	-1.27	-0.64	-0.09	-0.36	-0.58	0.75	0.54
T	***Thr***	-0.44	-0.67	0.35	0.59	0.29	0.04	0.91
W	***Trp***	2.25	-0.27	-0.27	-1.12	-0.75	-0.59	-0.52
Y	***Tyr***	1.31	-0.12	0.01	-0.93	-1.50	-1.62	-0.07
V	***Val***	0.29	-0.91	0.60	1.61	1.03	-0.85	0.32

Nella Figura 3.5 sono riportati i 20 aminoacidi codificati con le prime due componenti principali, mentre nella tabella successiva sono riportati, invece, i valori di correlazione delle componenti principali con le variabili originali : si può osservare che la prima componente è fortemente (anti)correlata (-0.940) con il volume dell'aminoacido, mentre la seconda lo è con l'idrofobicità (0.953). Ciò sta a significare **a)** che le due proprietà da sole sono in grado di descrivere efficientemente l'insieme dei venti aminoacidi, e che **b)** sono sostanzialmente incorrelate tra di loro (*ortogonali*).

Figura 3.5 - I venti aminoacidi fondamentali codificati dalla 1° (PC1) e dalla 2° (PC2) componente principale, calcolate su un insieme di sette proprietà fisico-chimiche (riportate in Tab. 3.1).

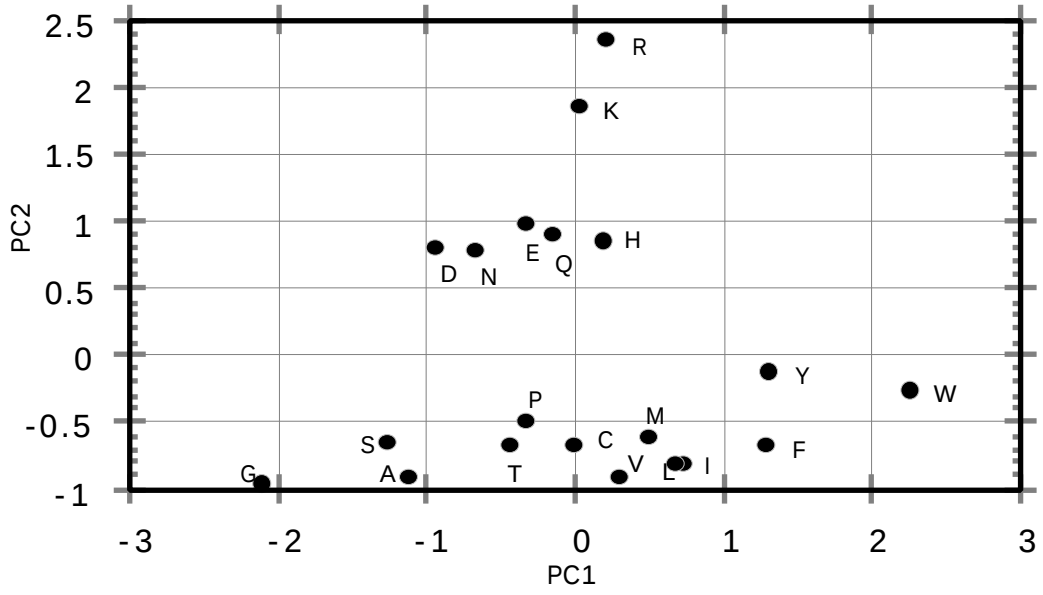


Tabella 3.3
***Loadings* delle componenti principali rispetto alle variabili originali.**

	PC-1	PC-2	PC-3	PC-4	PC-5	PC-6	PC-7
EV(%)	*50.04*	*34.73*	*7.43*	*5.29*	*1.90*	*0.47*	*0.14*
Hydrophob.	0.231	**0.953**	0.865	-0.560	0.857	0.863	-0.047
Volume	**-0.940**	0.239	0.466	0.736	-0.146	0.188	0.821
Surf_Area	-0.209	0.025	0.020	0.357	0.285	-0.071	-0.512
Hydrophil.	0.052	0.067	-0.023	-0.017	0.362	-0.423	0.229
Bulkiness	0.023	-0.142	-0.172	0.064	0.180	0.192	0.096
Refractivity	0.120	0.068	-0.012	0.113	-0.028	0.006	0.016
Polarity	0.030	-0.063	0.067	0.015	0.007	-0.003	0.003

In questo modo è stato individuato un criterio di codifica numerica della sequenza aminoacidica tale da **a)** conservare l'informazione posizionale, **b)** tenere conto delle caratteristiche fisico-chimiche degli aminoacidi, importanti nel processo di *folding*, e **c)** minimizzare la ridondanza di informazione (dovuta alla presenza di proprietà fisico-chimiche correlate).

Per l'utilizzo di questi descrittori numerici delle strutture primarie con l'algoritmo *SOMA* rimane quindi da superare il problema della diversità in lunghezza (i.e. in aminoacidi) dei vettori associati alle diverse proteine.

Una soluzione al problema dell'equalizzazione delle lunghezze è stata quindi proposta con lo sviluppo di un algoritmo originale denominato *Algoritmo del Letto di Procuste* * (Procust's Bed Algorithm, *PBA*) [Colosimo & Sirabella, 1993a, 1993b, 1994].

La caratteristica principale dell'algoritmo *PBA* è quella di *allungare* o *accorciare* il profilo numerico associato ad una struttura primaria mantenendone la *forma* il più possibile inalterata.

Una stima dell'affidabilità dell'algoritmo è stata fatta modificando (inserendo ed eliminando casualmente dei residui) delle strutture primarie di proteine. Riportando poi alla lunghezza iniziale (con l'algoritmo *PBA*) i profili numerici associati alle sequenze modificate è stato possibile calcolare il coefficiente di correlazione con il profilo originale. E' così possibile stimare, sulla base del valore correlazione, la percentuale di *allungamento* o *accorciamento* massima accettabile.

Lo Schema 3.1 e la Figura 3.6 seguenti illustrano in modo semplificato il funzionamento dell'algoritmo *PBA* .

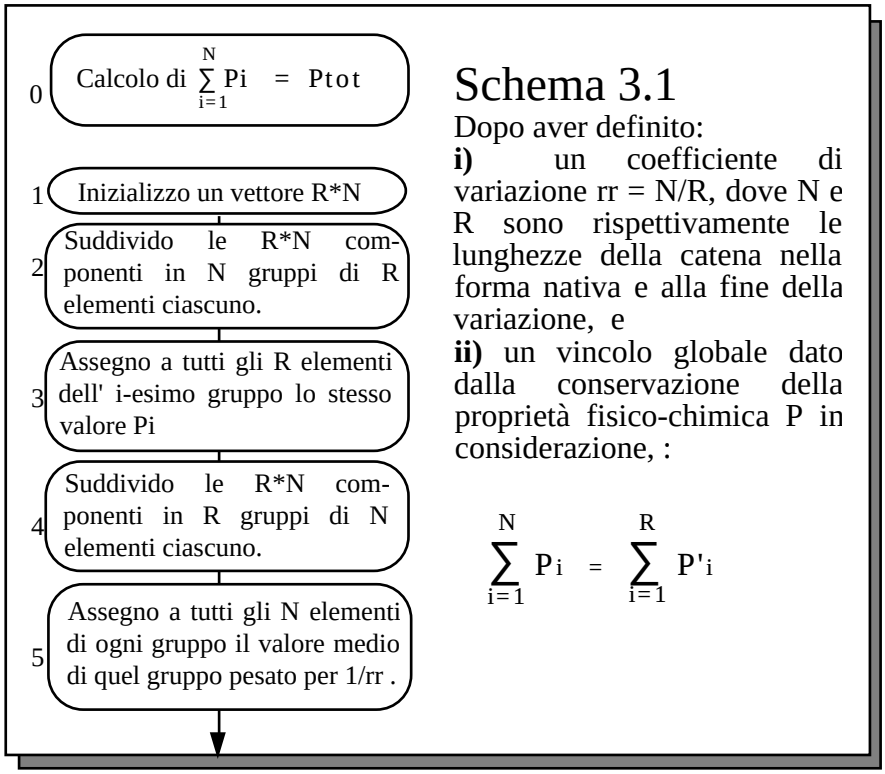
Un indice quantitativo della similitudine tra due profili (A,B) della stessa lunghezza (L) è fornito, come accennato, dall'indice di cross - correlazione (C.C.I.) calcolato in base alla formula di Pearson:

$$C.C.I._{(A,B)} = \frac{\sum_{i=1}^{L} (A_i - \overline{A})\ (B_i - \overline{B})}{[\sum_{i=1}^{L} (A_i - \overline{A})^2 \sum_{i=1}^{L} (B_i - \overline{B})^2\ ]^{1/2}}$$

dove A_i e B_i si riferiscono ai valori delle proprietà fisico-chimiche dell' i-esimo aminoacido sui profili A e B, rispettivamente, e $\overline{A}$, $\overline{B}$ sono i valori medi delle tali proprietà sull'intero profilo.

Nella Figura 3.3 vengono riportati, a titolo di esempio, gli effetti dell'applicazione dell'algoritmo *PBA* sul profilo di idrofobicità dell'azzurrina di *Pseudomonas aeruginosa*. I valori di correlazione calcolati tra il profilo originale e quello *allungato* e *accorciato* (calcolati su una media di 50 campioni differenti per ciascuna delle due situazioni) sono rispettivamente di **0.59** e **0.65**. Poiché i valori di correlazione completa e nulla corrispondono, rispettivamente, a 1 e 0, i valori ottenuti possono essere considerati abbastanza soddisfacenti, specialmente se si tiene conto che le modifiche apportate al profilo di idrofobicità per poter applicare l'algoritmo *PBA* sono, in entrambi i versi, di 10 parti su 125.

* Il nome deriva dal personaggio mitologico Procuste che, con le persone che gli erano avverse, utilizzava il proprio letto per confrontarne la lunghezza : a coloro che erano più lunghi venivano tagliati i piedi, mentre quelli che erano più corti venivano allungati.



Schema 3.1

Dopo aver definito:

i) un coefficiente di variazione rr = N/R, dove N e R sono rispettivamente le lunghezze della catena nella forma nativa e alla fine della variazione, e

ii) un vincolo globale dato dalla conservazione della proprietà fisico-chimica P in considerazione, :

$$\sum_{i=1}^{N} P_i = \sum_{i=1}^{R} P'_i$$

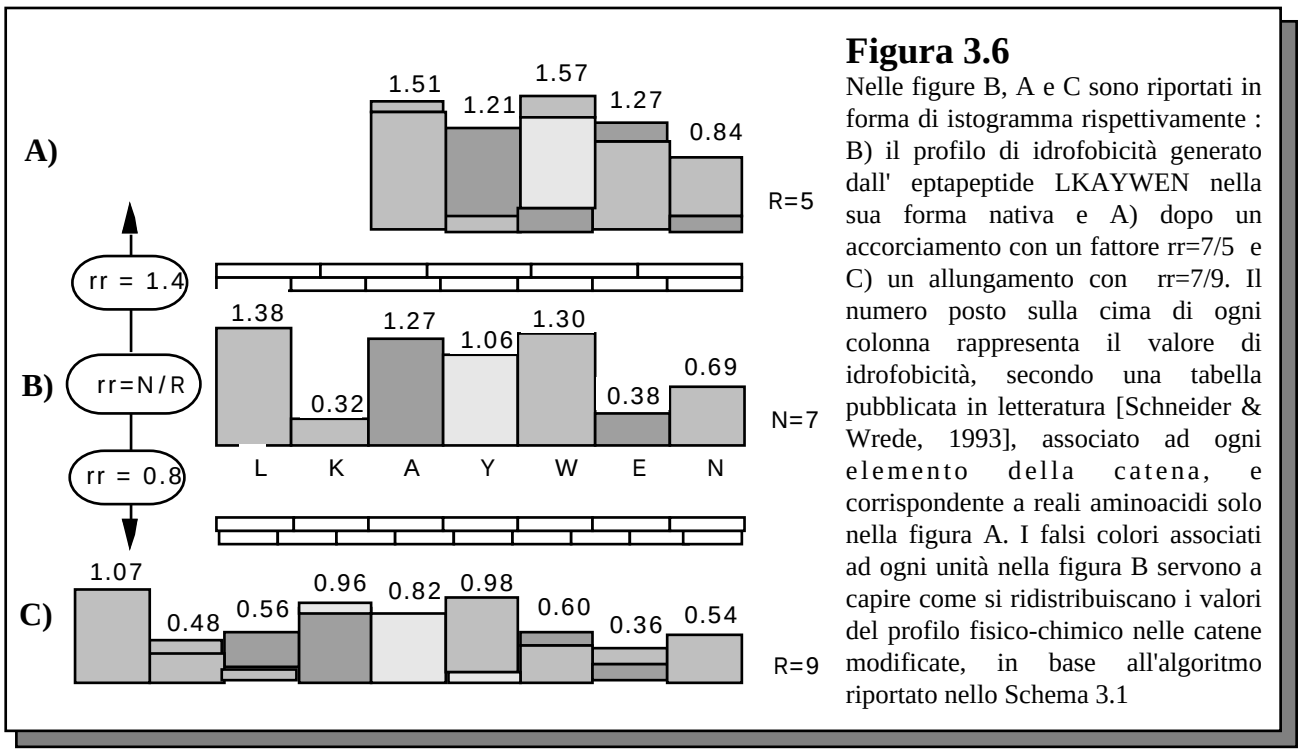
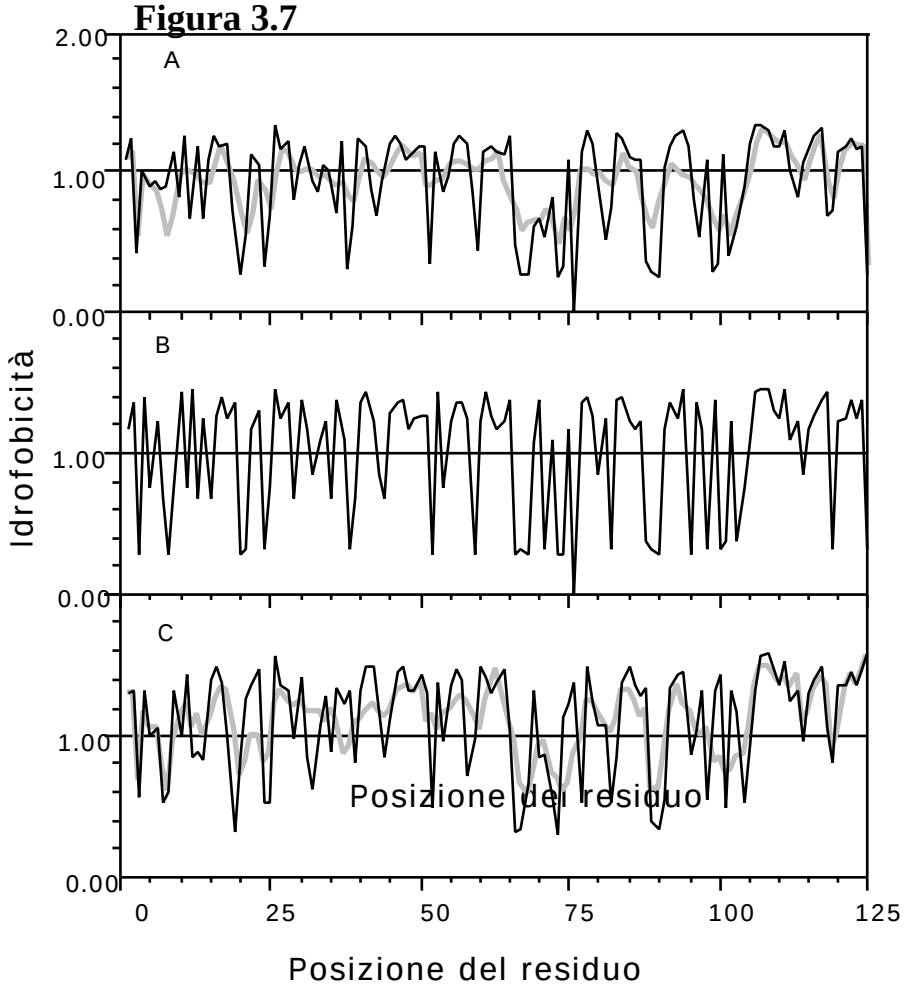


Figura 3.6

Nelle figure B, A e C sono riportati in forma di istogramma rispettivamente : B) il profilo di idrofobicità generato dall' eptapeptide LKAYWEN nella sua forma nativa e A) dopo un accorciamento con un fattore rr=7/5 e C) un allungamento con rr=7/9. Il numero posto sulla cima di ogni colonna rappresenta il valore di idrofobicità, secondo una tabella pubblicata in letteratura [Schneider & Wrede, 1993], associato ad ogni elemento della catena, e corrispondente a reali aminoacidi solo nella figura A. I falsi colori associati ad ogni unità nella figura B servono a capire come si ridistribuiscano i valori del profilo fisico-chimico nelle catene modificate, in base all'algoritmo riportato nello Schema 3.1

Figura 3.7




```
NBRF:Azpsca  Length: 128  April 20, 1994  16:06  Type: P  Check: 54
       1  AECSVDIQGN DQMQFNTNAI TVDKSCKQFT VNLSHPGNLP KNVMGHNWVL
      51  STAADMQGVV TDGMASGLDK DYLKPDDSRV IAHTKLIGSG EKDSVTFDVS
     101  KLKEGEQYMF FCTFPGHSAL MKGTLTLK
```

Figura 3.7 **Profili di idrofobicità dell'azzurrina di *Pseudomonas aeruginosa* prima e dopo l'applicazione dell'algoritmo *PBA*.**
Grafico A, linea piena: 10 residui, scelti casualmente, sono stati eliminati dalla struttura primaria dell' azzurrina di *Pseudomonas aeruginosa* e il profilo di idrofobicità risultante, lungo 115 residui, è stato riportato alla lunghezza originale di 125 residui.
Grafico A, linea tratteggiata: La procedura descritta sopra è stata ripetuta 50 volte e il profilo medio "allungato" è stato riportato nel grafico. Il valore di correlazione (CCI) tra i profili nativo e medio "allungato" è di **0.59** .
Grafico B : profilo di idrofobicità dell'azzurrina di *Pseudomonas aeruginosa* calcolato sulla base dei valori riportati nella colonna 1 della Tabella 3.1 e della sua struttura primaria (125 AA, dal precursore NBRF::Azpsca, 128 AA. Nella sequenza riportata i primi tre aminoacidi sono stati trascurati poiché presenti solo nel precursore).
Grafico C, linea piena: 10 residui, scelti casualmente, sono stati inseriti nella struttura primaria dell' azzurrina in posizioni casuali e il profilo di idrofobicità risultante, lungo 135 residui, è stato riportato alla lunghezza originale di 125 residui.
Grafico C, linea tratteggiata: La procedura descritta sopra è stata ripetuta 50 volte e il profilo medio "accorciato" è stato riportato nel grafico. Il valore di correlazione (CCI) tra i profili nativo e medio "accorciato" è di **0.65** .

3.2.2 Algoritmo della “Buccia di cipolla”

Al fine di verificare la correttezza della classificazione delle strutture primarie di proteine ottenuta con l'utilizzo congiunto **a)** dell'algoritmo di classificazione *SOMA* di Kohonen, **b)** dei criteri di codifica studiati e **c)** dell'algoritmo di equalizzazione delle lunghezze *PBA* (tutti descritti nei capitoli precedenti), si è reso necessario lo studio di un criterio di classificazione delle proteine a partire, stavolta, dalla loro struttura tridimensionale. Si ricorda, infatti, che uno degli obiettivi principali di questo lavoro è quello di ottenere un metodo affidabile di codifica e classificazione delle strutture primarie di proteine che fornisca delle *mappe* rappresentative che siano in grado di evidenziare le relazioni esistenti tra sequenza e conformazione spaziale (nel rispetto dell'osservazione di Anfinsen). Se è corretta l'ipotesi che una appropriata codifica delle strutture primarie e una loro efficiente classificazione permettono di ottenere delle mappe di proteine nelle quali le relazioni di similitudine tra strutture primarie sono concordi con quelle esistenti tra strutture tridimensionali, allora il confronto fra mappe ottenute da strutture primarie e strutture terziarie dello stesso set di proteine può costituire un buon criterio di validazionedella correttezza delle procedure proposte.

La procedura di classificazione, ed implicitamente di codifica, delle strutture terziarie di proteine deve avere una serie di requisiti fondamentali : **a)** deve essere in grado di trattare proteine di lunghezza (numero di residui) a priori differente, **b)** deve essere indipendente dalla particolare orientazione della molecola, **c)** deve in qualche modo descrivere forma e dimensione della proteina, **d)** deve codificare in modo automatico la struttura tridimensionale, possibilmente partendo direttamente dalla lettura del *file* contenente le coordinate degli atomi costituenti la molecola in formato standard PDB (Protein Data Bank - Brookhaven National Laboratories), estremamente diffuso e presente nella maggior parte delle banche dati accessibili tramite rete telematica.

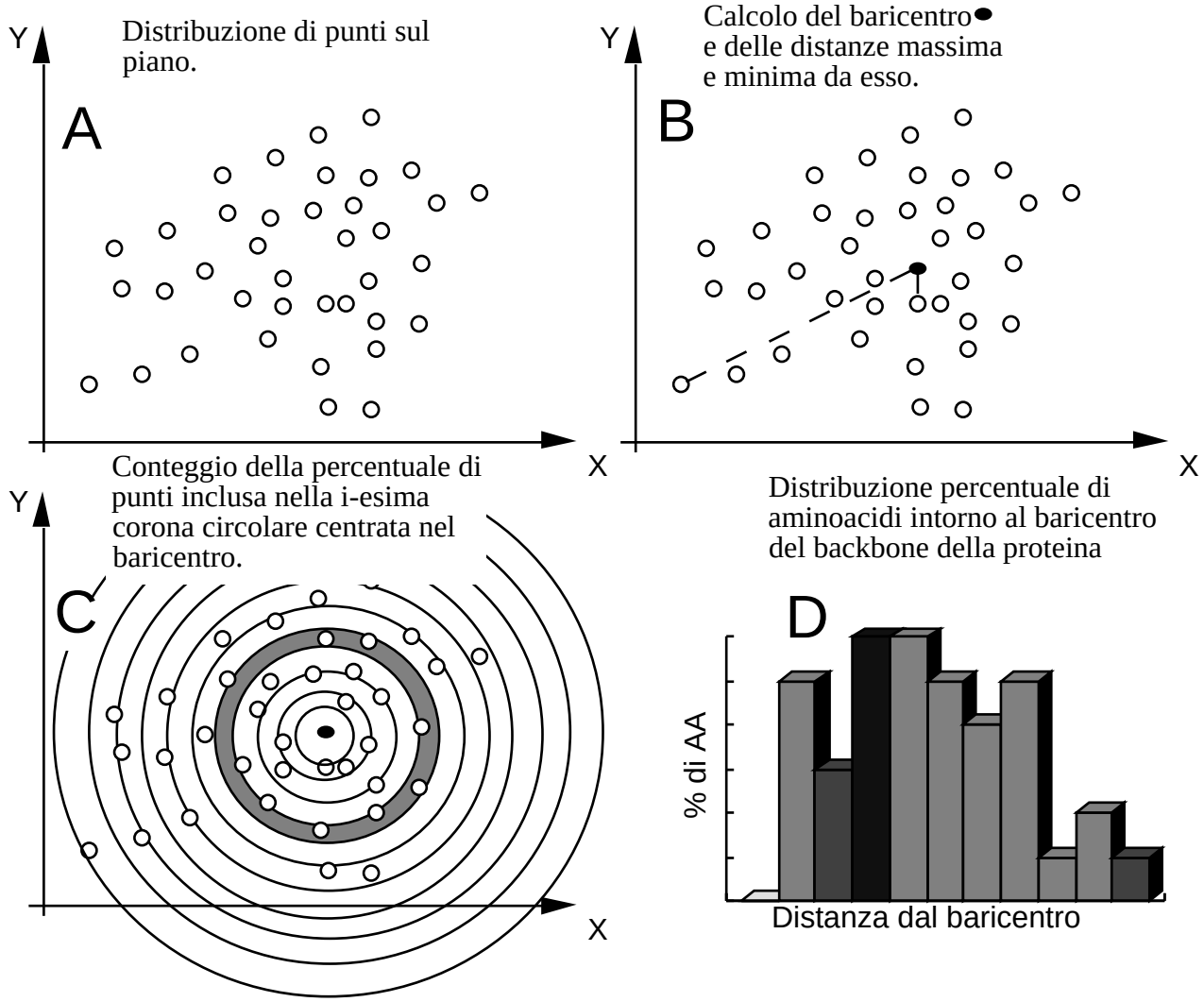
L'idea di base del algoritmo "Buccia di Cipolla" (Onion's Peel Algorithm, *OPA*) è sostanzialmente quella di descrivere la struttura terziaria di proteine come distribuzione radiale di residui intorno ad un *punto preferenziale* della molecola.

La prima semplificazione, dovuta a motivi principalmente computazionali, consiste nel limitarsi al *back-bone* della proteina, considerando quindi le sole coordinate dei carboni alfa della catena polipeptidica.

Per quel che riguarda la scelta del *punto preferenziale* , ci si è limitati al semplice calcolo del baricentro del *back-bone* che, essendo costituito da elementi identici, non è nient'altro che il centro geometrico (x_C, y_C, z_C). Per una proteina di N residui, avremo :

$$x_c = \frac{\sum_{i=1}^{N} x_i^{C\alpha}}{N} \; ; \; y_c = \frac{\sum_{i=1}^{N} y_i^{C\alpha}}{N} \; ; \; z_c = \frac{\sum_{i=1}^{N} z_i^{C\alpha}}{N} \; .$$

La distribuzione radiale di residui per una proteina di N residui viene quindi calcolata **a)** scegliendo un numero arbitrario M di intervalli (collegato alla risoluzione dell'analisi conformazionale), **b)** calcolando la massima e la minima* distanza dei residui dal baricentro, e quindi il *range* di distanze presenti, **c)** calcolando il rapporto (d=*range* / M), e, infine, conteggiando la percentuale di aminoacidi inclusi in ognuna delle M calotte sferiche concentriche (da qui il nome scelto per l'algoritmo) centrate nel baricentro e di spessore d. La figura seguente schematizza, nel caso bidimensionale, il funzionamento dell'algoritmo *OPA* :



* Non necessariamente il baricentro si trova in corrispondenza di uno dei carboni alfa del *backbone* , è anzi abbastanza frequente che esso non appartenga al *backbone* , e che quindi esista anche una distanza minima dal baricentro. E' poi possibile che determinate strutture, per così dire *cave* (sempre per quel che riguarda il *backbone*), abbiano questa distanza minima non così piccola.

Nel caso in cui ci siano più strutture terziarie di proteine da codificare e se ne voglia ottenere una classificazione, bisogna prendere ulteriori accorgimenti :

a) nell'analisi delle coordinate atomiche dei carboni-alfa bisogna calcolare per ciascuna struttura il baricentro, e ..

b) ... per ciascuna proteina bisogna calcolare la massima e la minima distanza dal baricentro.

c) Il range di distanze dal baricentro da esplorare sarà, per ciascuna struttura, quello che va dalla minore delle minime distanze alla maggiore delle massime. In questo modo si definira un riferimento unico per l'intero set di strutture terziarie di proteine.

Per quel riguarda la classificazione, una volta ottenuto per ciascuna delle N proteine un vettore ad M componenti (dove M è il numero di intervalli nei quali è stato diviso il range di distanze dal baricentro), si potrà effettuare una analisi dei *clusters*, a partire dalla matrice NxM unità-variabile ottenuta. I possibili metodi per ottenere una classificazione sono diversi : una scelta interessante si è dimostrata quella di utilizzare l'Analisi delle Componenti Principali sui vettori M-dimensionali (il numero M di intervalli, associato alla risoluzione dell'analisi conformazionale, può a priori anche essere grande). Scegliendo le prime due componenti principali, e assicurandosi che siano in grado di descrivere una percentuale accettabile di variabilità del sistema, è possibile rappresentare l'insieme di strutture terziarie analizzate su una mappa bidimensionale, da mettere quindi in relazione con la mappa bidimensionale delle strutture primarie delle medesime proteine, prodotta con gli algoritmi *PBA* e *SOMA* .