

Recupero di dati ed elaborazione di segnali e immagini per
bioinformatica

*MODULO: Riconoscimento e Recupero dell'informazione per
Bioinformatica*

Il clustering

Manuele Bicego

Corso di Laurea in Bioinformatica
Dipartimento di Informatica - Università di Verona

Nota preliminare

- ⇒ Il clustering è stato studiato in molte aree scientifiche, come la biologia, la psichiatria, la psicologia, l'archeologia, la geologia, la geografia, il marketing....
 - ⇒ Ogni area ha la sua terminologia
 - ⇒ Qui ci focalizziamo sulla prospettiva della Pattern Recognition
- ⇒ Altri termini tipicamente utilizzati per il clustering:
unsupervised learning, numerical taxonomy, vector quantization, learning by observation...

Una definizione possibile

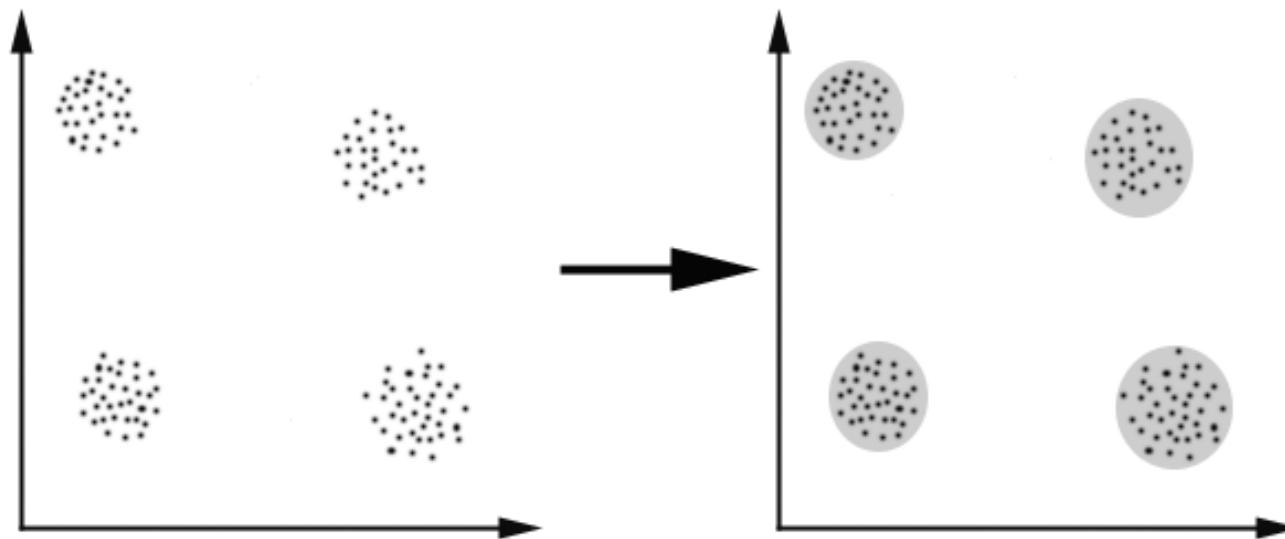
[Jain *et al.*, ACM Computing Surveys, 1999]

- Il clustering rappresenta l'organizzazione di un insieme di patterns (entità/oggetti) in gruppi sulla base della similarità
 - ⇒ pattern che appartengono allo stesso gruppo sono tutti **simili tra loro**;
 - ⇒ pattern di gruppi diversi sono invece **differenti tra di loro**

I gruppi si chiamano **clusters**

- Il processo è completamente “non supervisionato”
 - ⇒ Non è data nessuna informazione a priori sui gruppi

Clustering



Esempio di clustering: un insieme di punti in uno spazio 2D sono raggruppati in 4 clusters

Applicazioni

- Il clustering è stato utilizzato con successo in moltissimi contesti, principalmente per analisi esplorative (*exploratory data analysis*)
 - Fondamentale quando c'è poca informazione a priori (o nessuna)
 - Utile per inferire relazioni tra gli oggetti del problema, in modo da avere un'idea (anche preliminare) della loro organizzazione

Vediamo alcuni esempi di applicazioni del clustering...

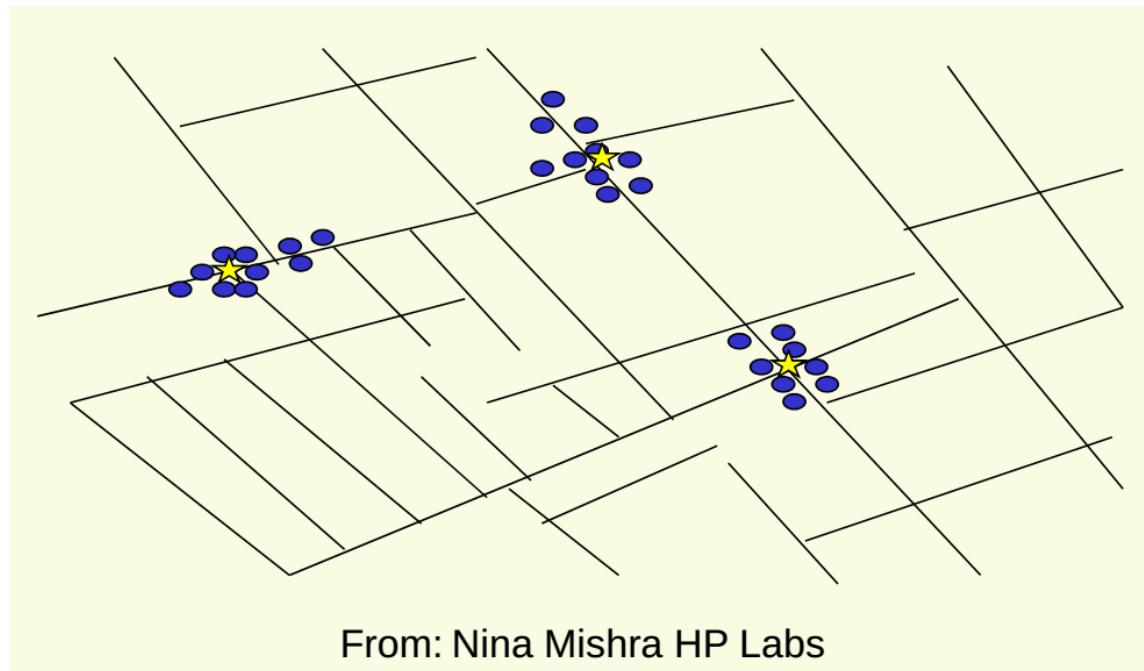
La prima (?) applicazione del clustering

- Nel 1850, John Snow, un fisico di Londra, durante un'epidemia di colera ha disegnato su una mappa la posizione dei casi e delle morti
- L'obiettivo principale era di capire se esisteva una possibile relazione tra le morti e le diverse parti della città



La prima (?) applicazione del clustering

- Guardando la mappa il fisico fu in grado di vedere che i casi erano tutti raggruppati attorno a determinati incroci
- In quegli incroci c'erano tombini con acqua inquinata!



Il clustering ha permesso di:

I. Capire il problema!

II. Suggerire la soluzione!

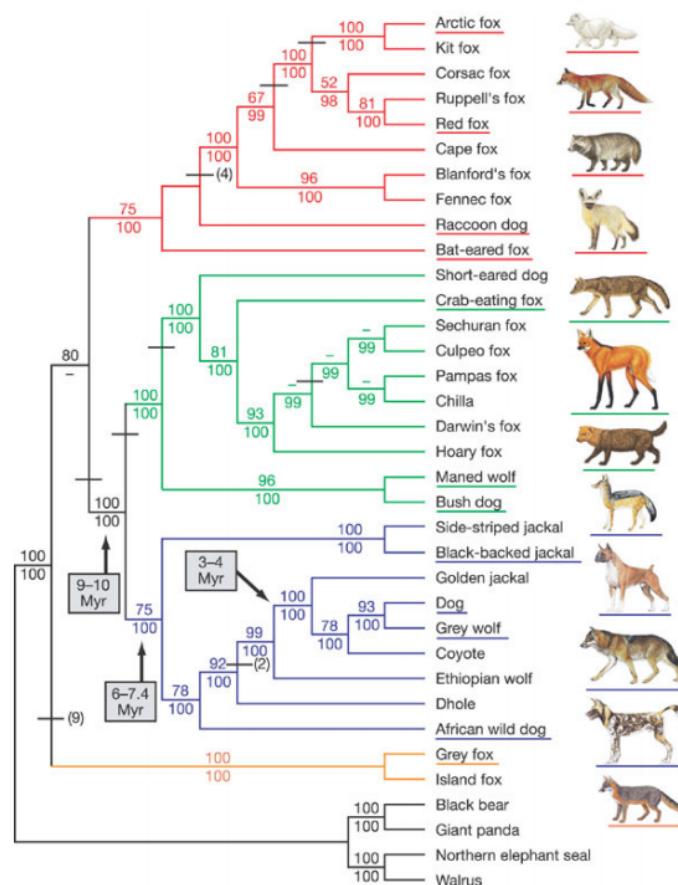
Altre applicazioni

- Segmentazione di immagini: dividere l'immagine in zone “simili”



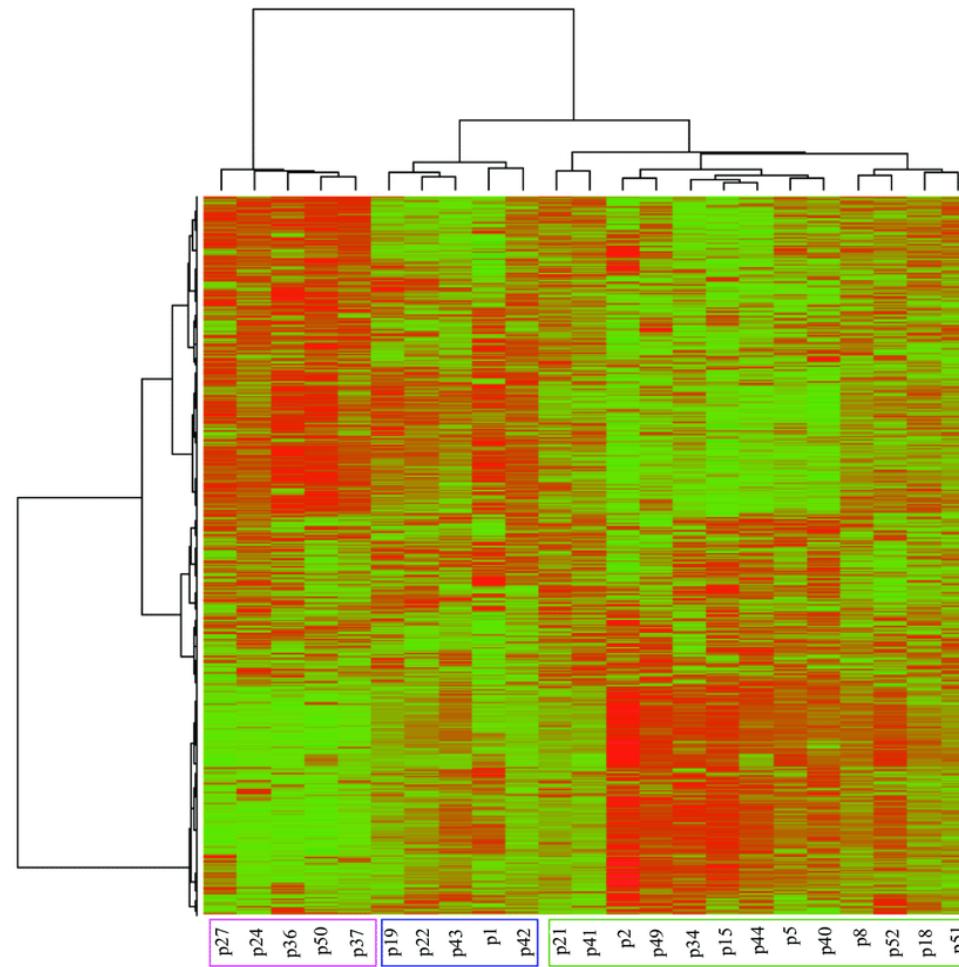
Altre applicazioni

- Bioinformatica: inferire le relazioni genealogiche tra gli organismi (filogenesi) tramite clustering di sequenze geniche o proteiche



Altre applicazioni

- Bioinformatica: trovare gruppi di geni co-regolati, cioè geni con un pattern di attivazione simile



Altre applicazioni

- Analisi di Social Networks: creare gruppi di utenti sulla base di interessi/comportamenti comuni (e.g. tweets simili su Twitter)



Altre applicazioni

- Molte altre!!!!
 - Astronomia, organizzazione di databases, marketing, psicologia, profiling di utenti web, monitoraggio ambientale, clustering di linguaggi, ...

Il Clustering è difficile!!

Una definizione possibile

[Jain et al., ACM Computing Surveys, 1999]

- ⇒ Il clustering rappresenta l'organizzazione di un insieme di patterns (entità/oggetti) in gruppi sulla base della similarità
- ⇒ pattern che appartengono allo stesso gruppo sono tutti **simili tra loro**;
- ⇒ pattern di gruppi diversi sono invece **differenti tra di loro**

I gruppi si chiamano **clusters**

- ⇒ Il processo è completamente “non supervisionato”
 - ⇒ Non è data nessuna informazione a priori sui gruppi

Qual’è la similarità più appropriata?
Su che base definisco che due oggetti sono simili?

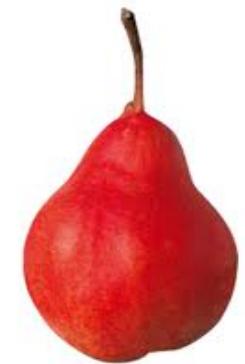
Cambiare la similarità cambia il risultato!

Problema 1: Il concetto di cluster è vago e mal posto

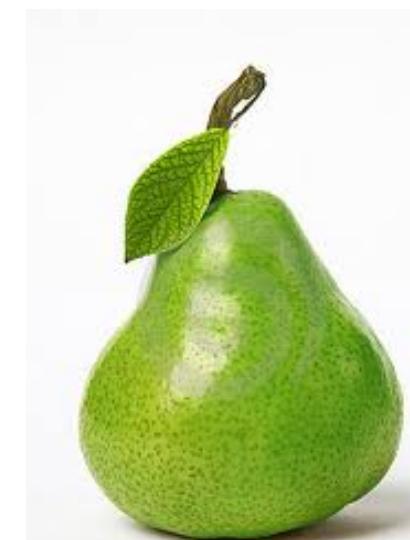
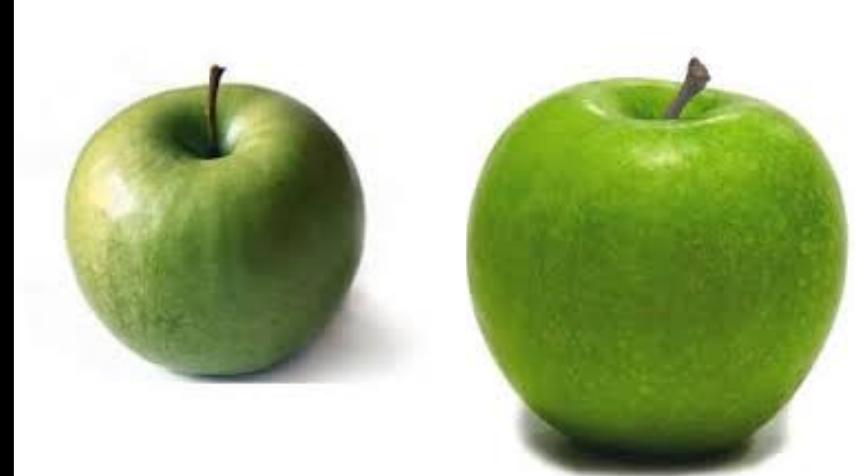
Esempio: Oggetti da clusterizzare



Ci sono 2 gruppi: mele e pere



Altra possibilità: frutta rossa e frutta verde

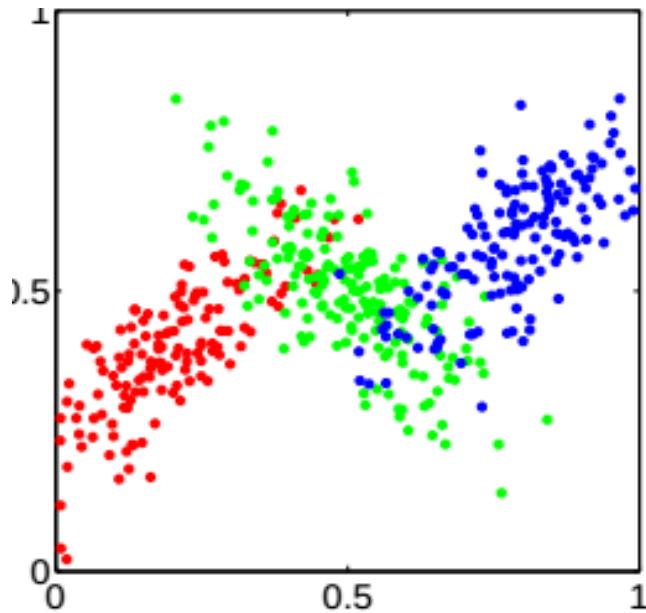


Il Clustering è difficile!!

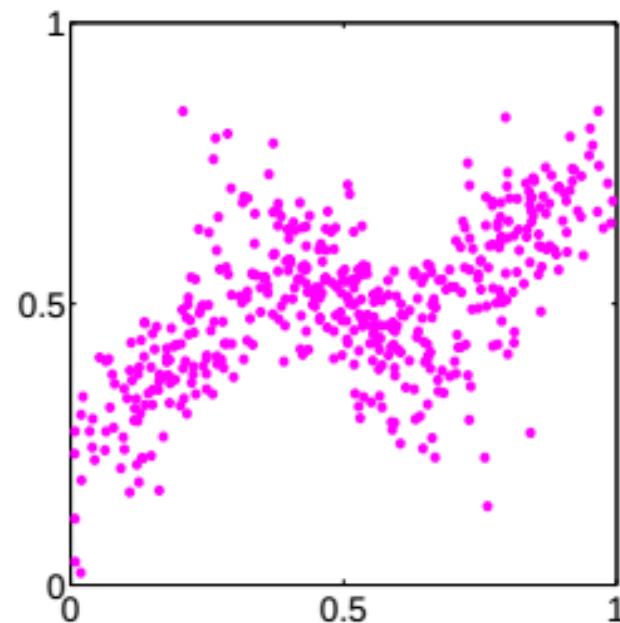
- Quindi: il concetto di gruppo è definito in modo vago e assolutamente soggettivo
 - Dipendentemente dalle misure di similarità utilizzate cambia il risultato
 - Il risultato può cambiare anche a seconda della metodologia utilizzata per fare clustering (più chiaro in seguito)

Il Clustering è difficile!!

- **Problema 2:** occorre derivare il modello senza alcuna informazione a priori (al limite il numero dei gruppi)



Informazione a priori: le etichette
Facile: 3 Gaussiane!



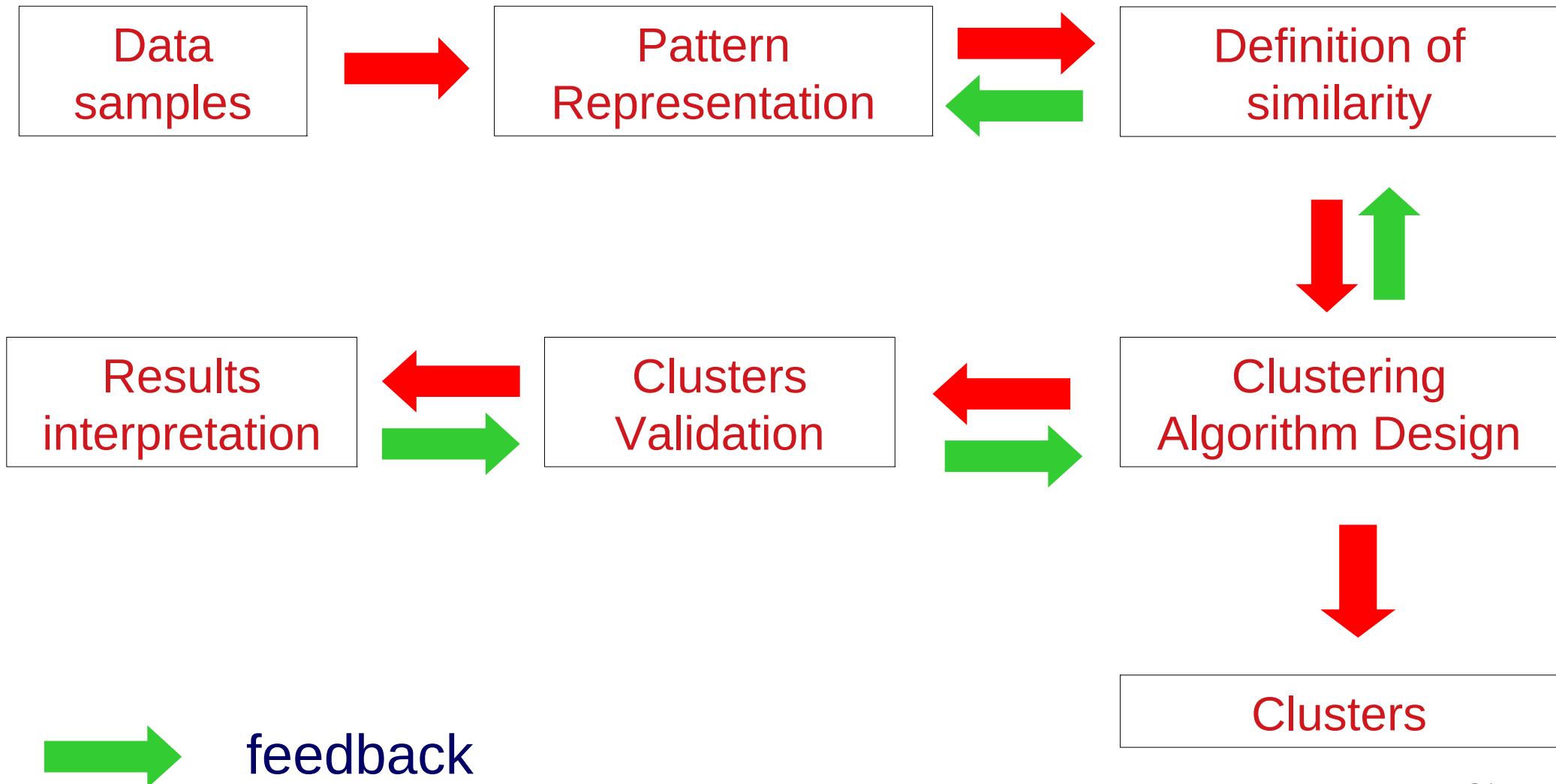
Nessuna informazione a priori
Decisamente più difficile!

Il Clustering è difficile!!

- **Problema 3:** non c'è il “ground truth” (cioè non sappiamo il risultato): la validazione “quantitativa” del clustering è molto difficile:
 - I cluster che ho trovato sono significativi? E come posso misurare questo?
 - Quanti gruppi ho?
 - *Esiste effettivamente una struttura di clustering? (cluster tendency)*
- Soluzione tipica: conoscenze derivanti dal settore applicativo

Un tipico sistema di
clustering

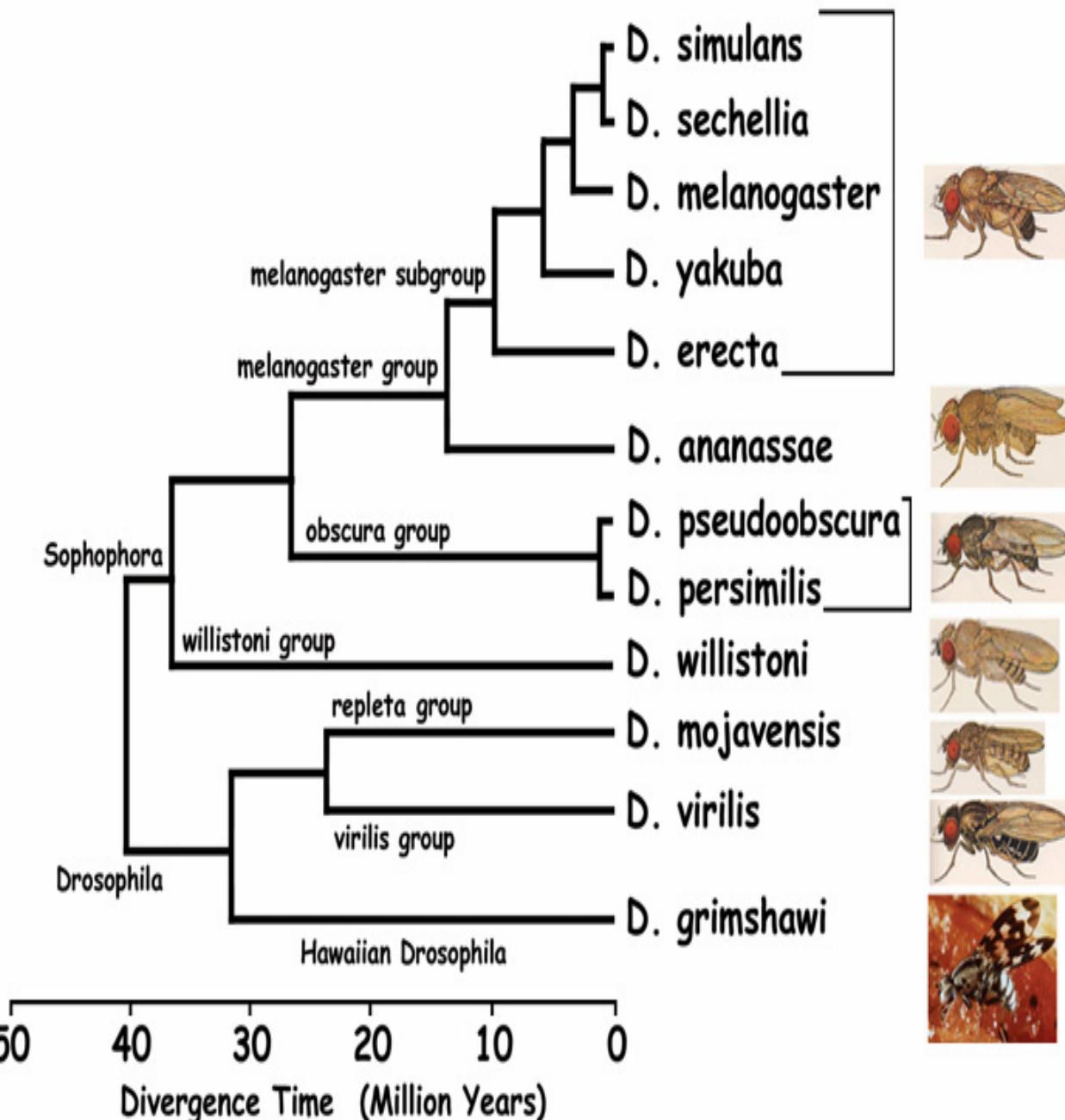
Un tipico sistema di clustering



Esempio: la filogenesi

Vediamo i vari passi
con un esempio
guida: la filogenesi

Filogenesi: inferire
le relazioni
genealogiche tra gli
organismi
⇒ clustering di sequenze
geniche o proteiche

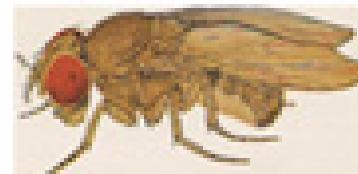


Rappresentazione dei Pattern

- Descrizione digitale del pattern
(già vista)
- Concetti di tipo di pattern, tipo di dato, preprocessing, estrazione di features, selezione di features...
- Anche per il clustering la rappresentazione è cruciale

Esempio: la filogenesi

- Insetti da clusterizzare



- Dati grezzi: le sequenze di DNA relative ad un determinato gene

```
CAGATCTTGACGATCCAAAGTGGTTCATTGGCTTAGATGAAG  
TACCGATCTTGACGATCCAAAGTCATTGGCTTAGATGAAG  
CAGATCTTGACGATCCAAAGTGGTTCATTGGCTTAGATGAAG  
CAGATCTTCACGATCCAAAGTGGTTCATTGGCTTAGAT
```

- Pre-processing: allineamento delle sequenze

```
CA--GATCTTGACGATCCAAAGTGGTTCATTGGCTTAGATGAAG  
TACCGATCTTGACGATCCAAAG----TCATTGGCTTAGATGAAG  
CA--GATCTTGACGATCCAAAGTGGTTCATTGGCTTAGATGAAG  
CA-GATCTTCACGATCCAAAGTGGTTCATTGGCTTAGAT----
```

Similarità

- La misura di similarità è fondamentale.
 - la maggior parte degli algoritmi di clustering dipende strettamente dalla definizione di questa misura
 - Dovrebbe essere fatta in modo da inglobare la maggior quantità possibile di informazione a priori.
- Esistono molte definizioni diverse
 - ⇒ dipendentemente dal dominio, dal tipo di rappresentazione, dalla conoscenza a priori
- Concetti equivalenti: similarità / dissimilarità

Esempio: la filogenesi

- La distanza tra due sequenze si potrebbe ragionevolmente legare al numero di “differenze” (e.g. sostituzioni) che ci sono tra le due sequenze (eventualmente pesate)
- ESEMPIO: misura di Jukes-Cantor

$$d(S_1, S_2) = -\log \left(1 - \frac{4}{3} p \right)$$

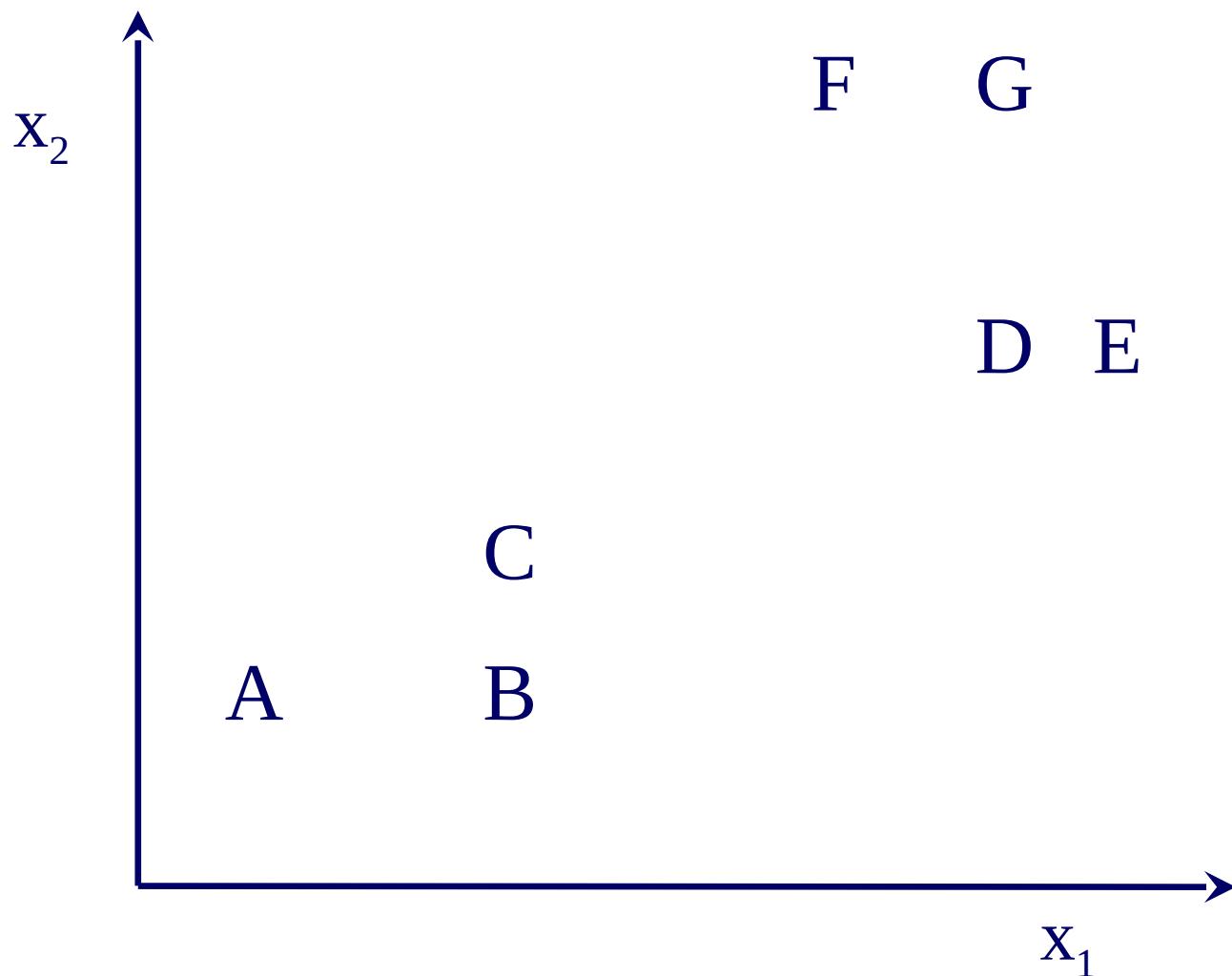
(p = proporzione di nucleotidi dove le due sequenze differiscono)

Metodologie di Clustering

- Obiettivo: trovare i gruppi data la definizione di similarità
- Non esiste un'unica metodologia appropriata per tutti i problemi
 - ⇒ la scelta di un algoritmo appropriato dipende dal dominio, dal processo di acquisizione, dalla conoscenza a priori, dalla quantità di dati a disposizione
- Ci sono molti metodi in letteratura
 - ⇒ Diversi criteri di ottimizzazione, assunzioni, modelli, requisiti computazionali
- Principale suddivisione: metodi partizionali o gerarchici

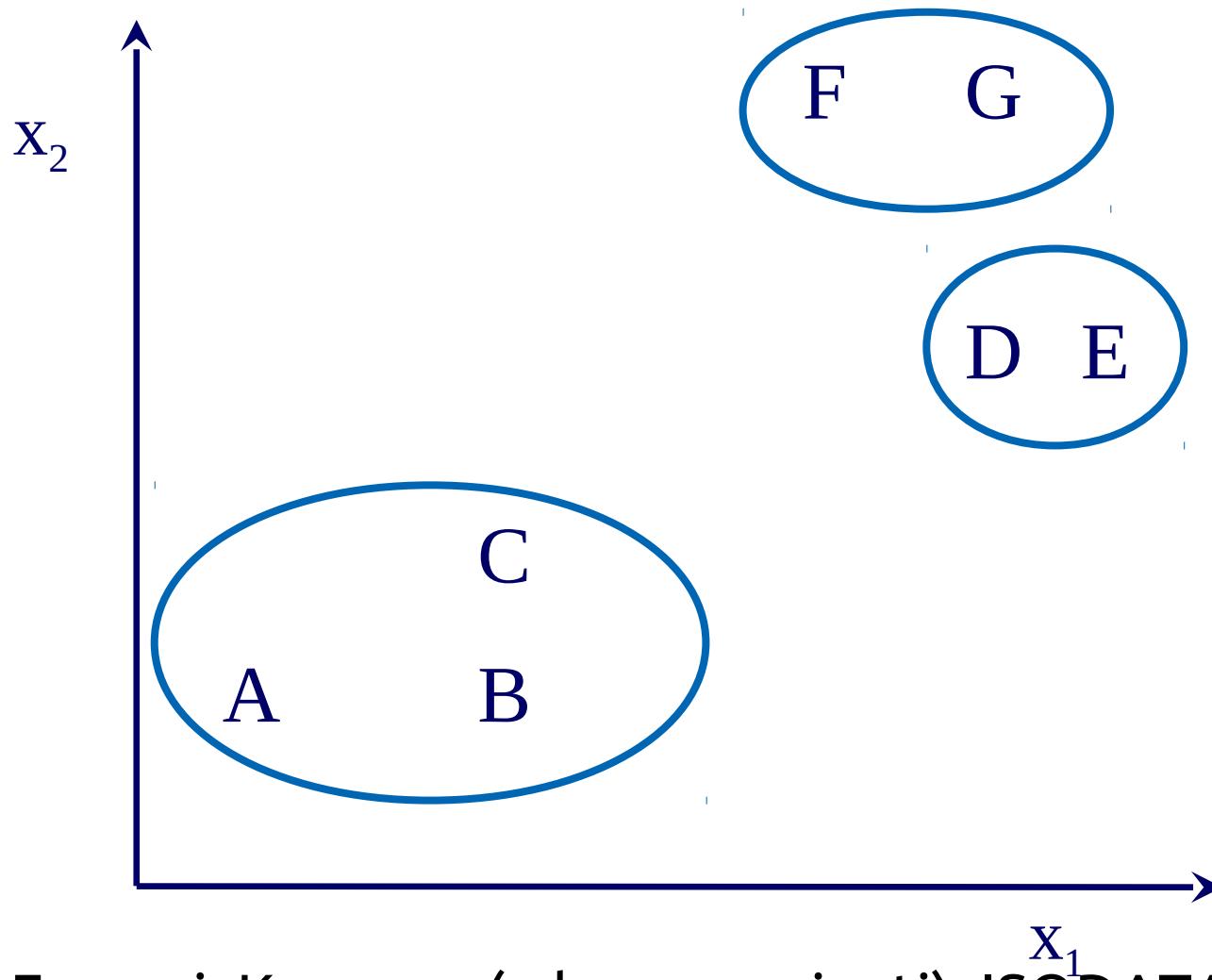
Metodi partizionali

il risultato è una singola partizione del dataset



Metodi partizionali

il risultato è una singola partizione del dataset



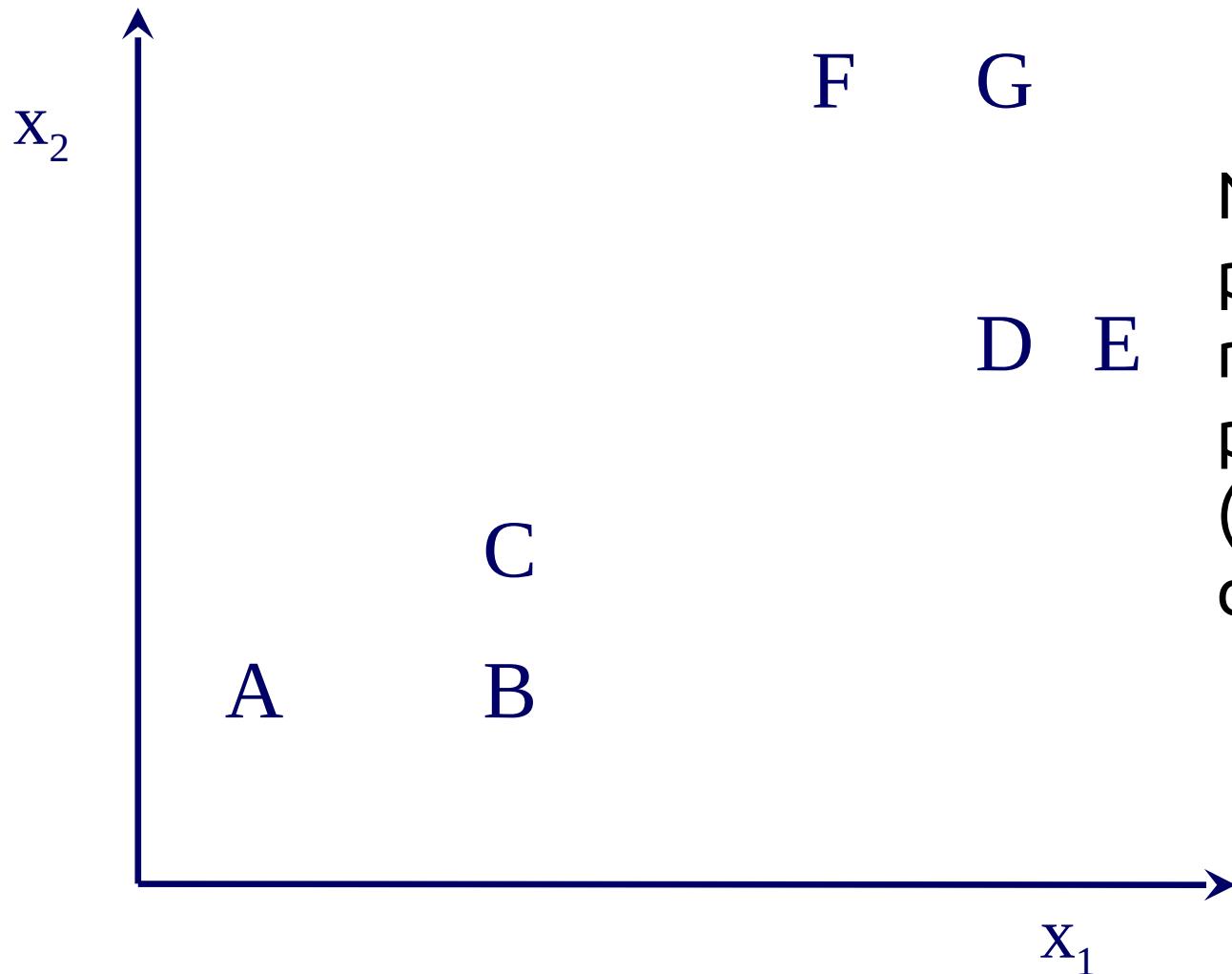
Tipicamente il numero di cluster è dato in ingresso

In questo caso
Numero clusters = 3

Esempi: K-means (e le sue varianti), ISODATA, PAM, ...

Metodi gerarchici

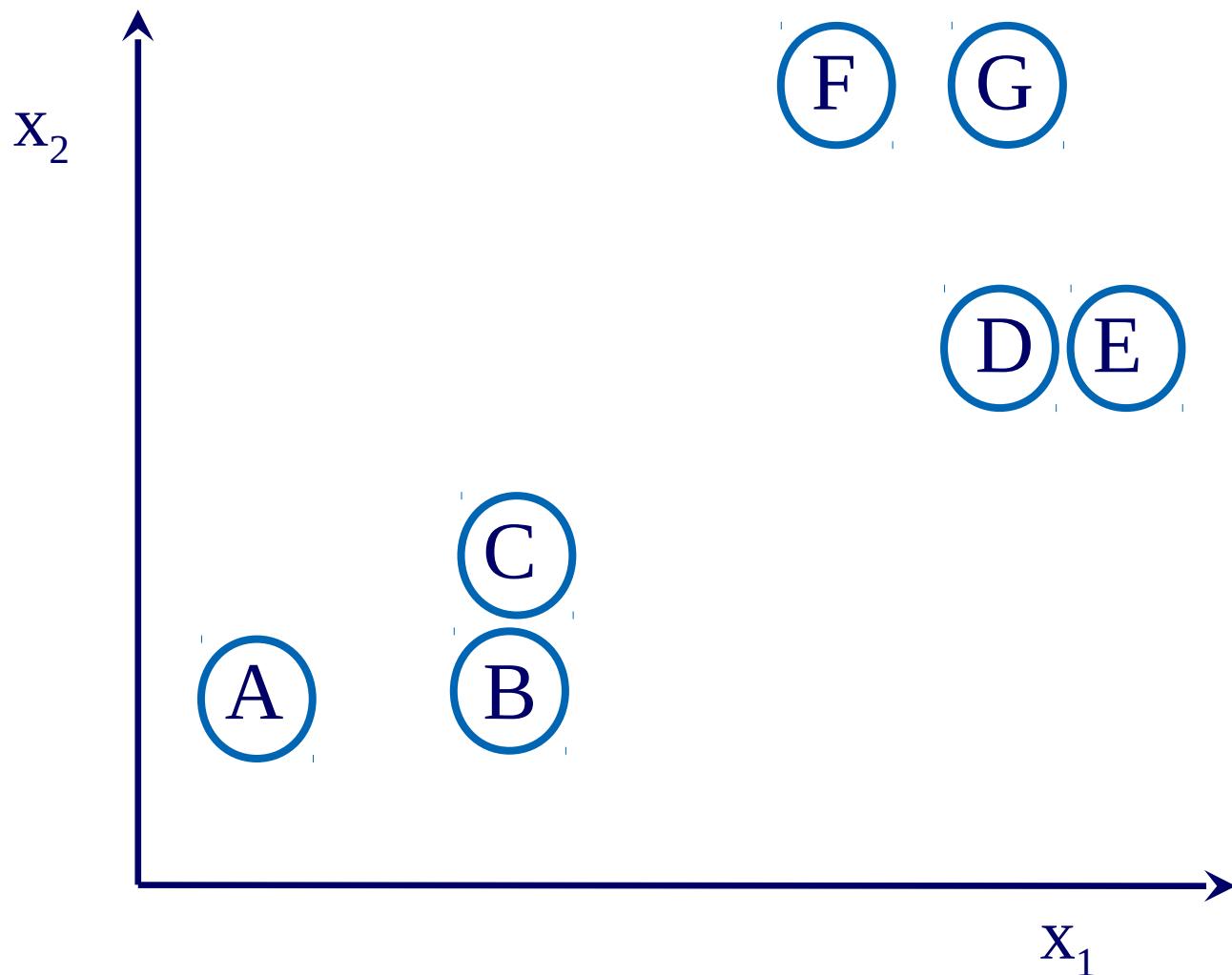
il risultato è una serie di partizioni innestate
(una gerarchia di partizioni):



Nella gerarchia di partizioni, ogni livello i rappresenta una partizione più “generale” (cioè con meno clusters) del livello $i-1$

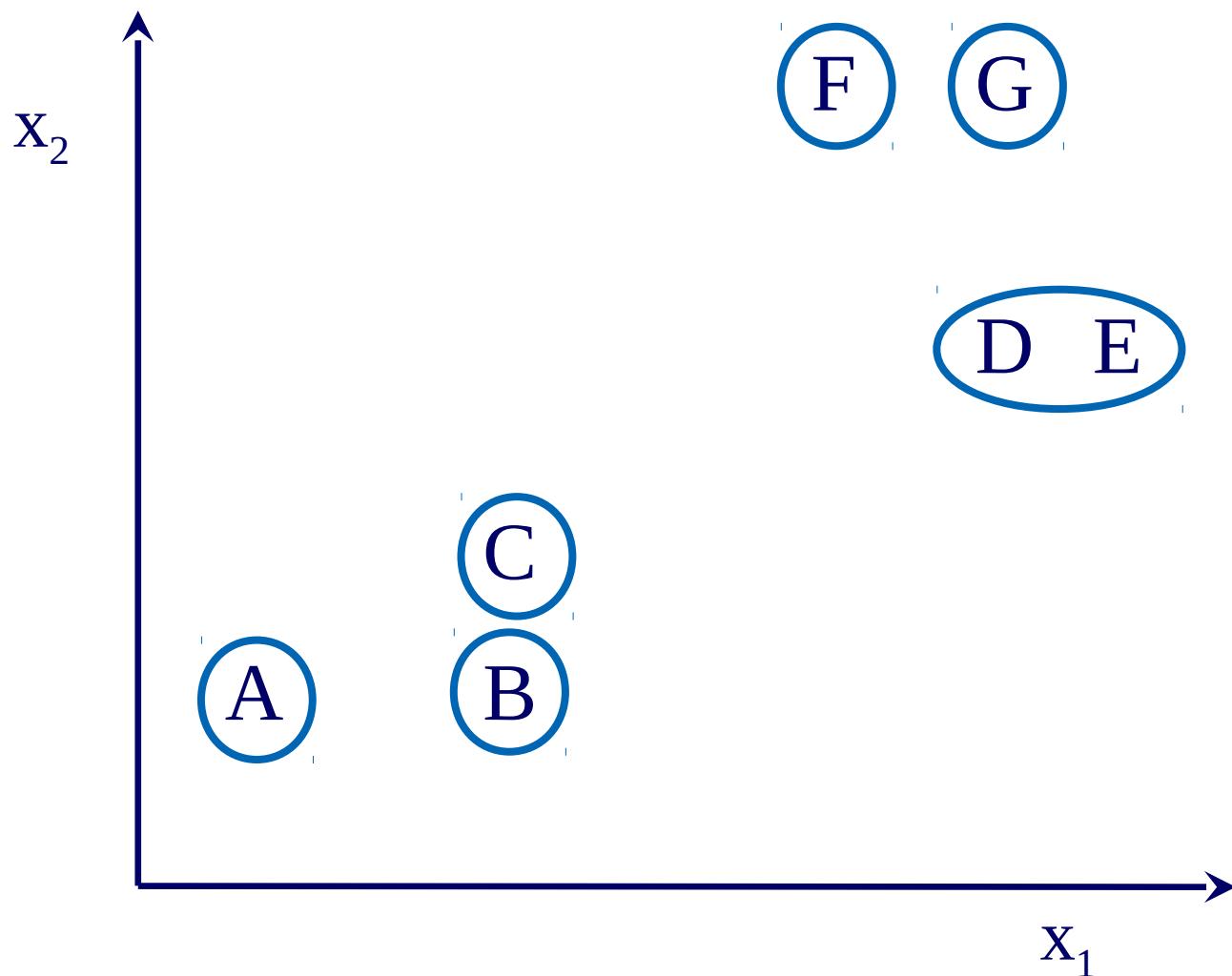
Metodi gerarchici

Livello 0 (più specifico): ogni oggetto è in un singolo cluster (7 oggetti → 7 clusters)



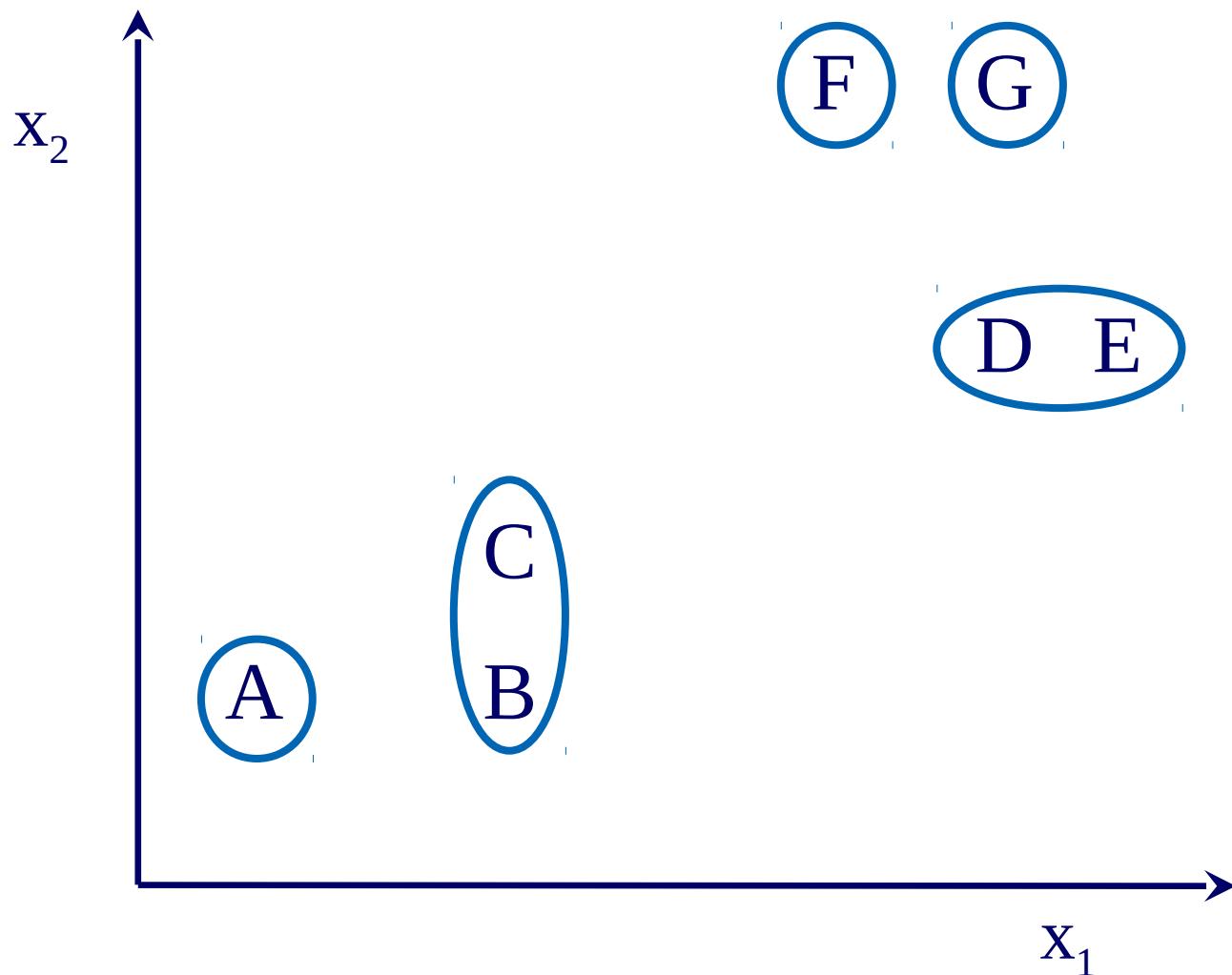
Metodi gerarchici

Livello 1: 6 clusters



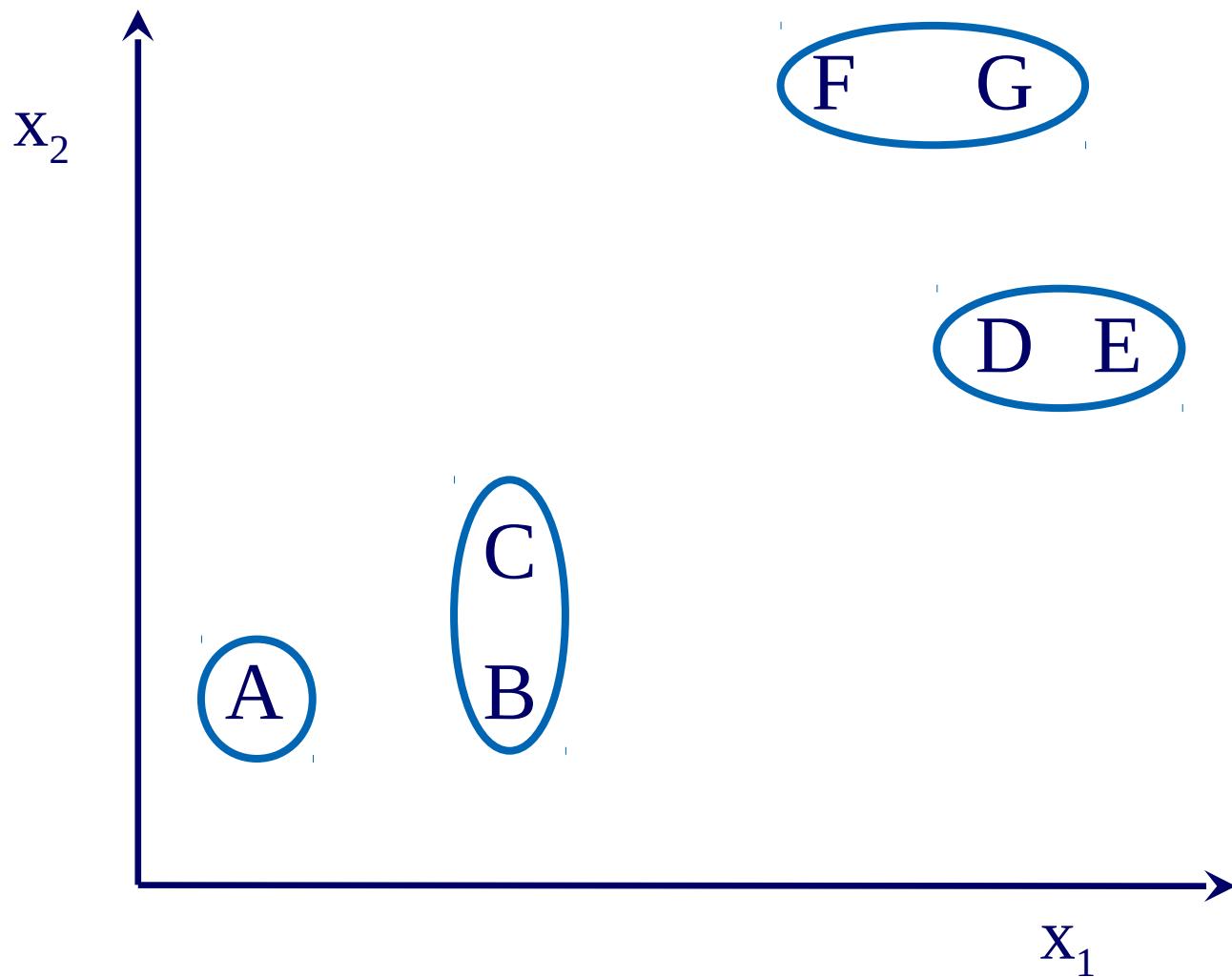
Metodi gerarchici

Livello 2: 5 clusters



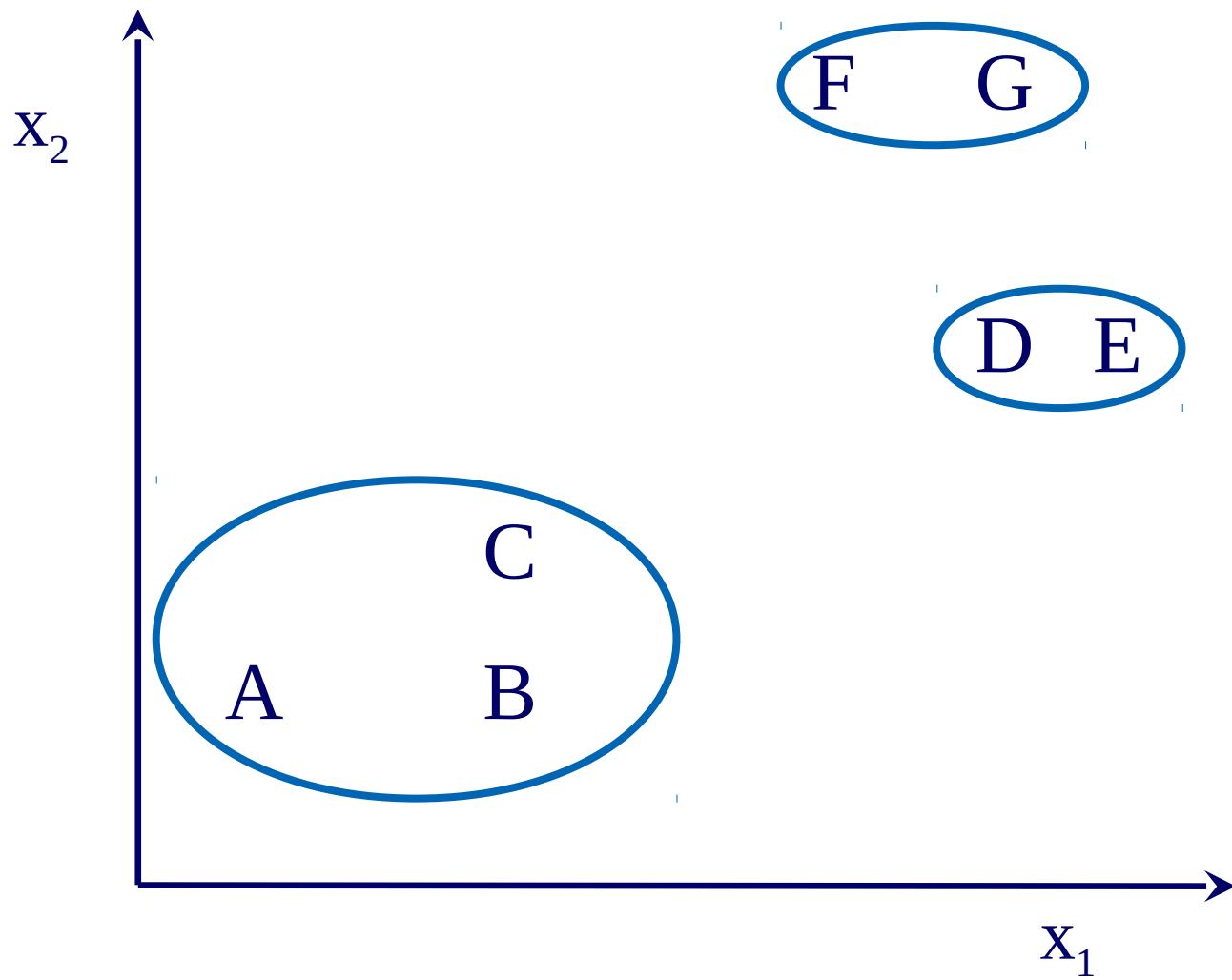
Metodi gerarchici

Livello 3: 4 clusters



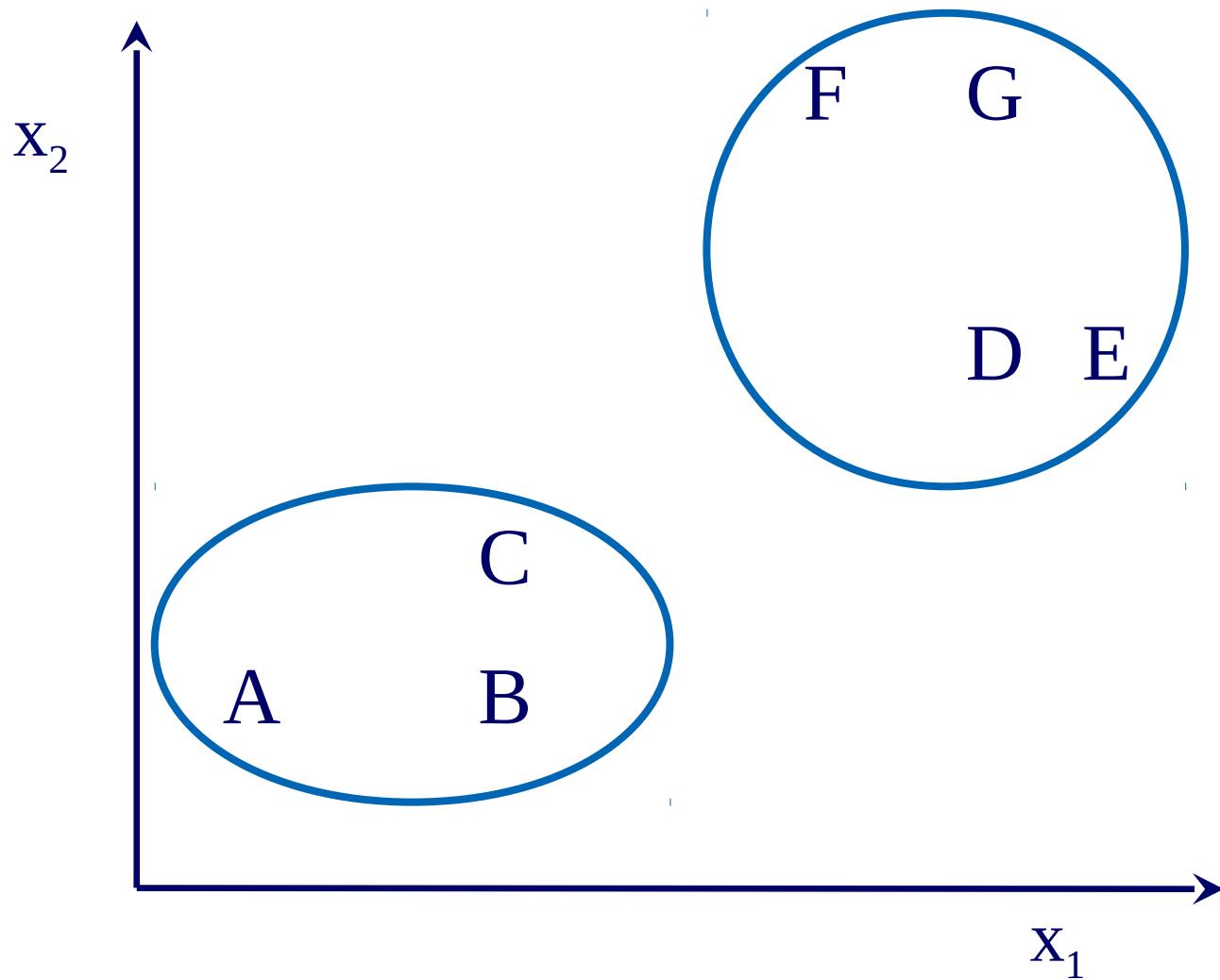
Metodi gerarchici

Livello 4: 3 clusters



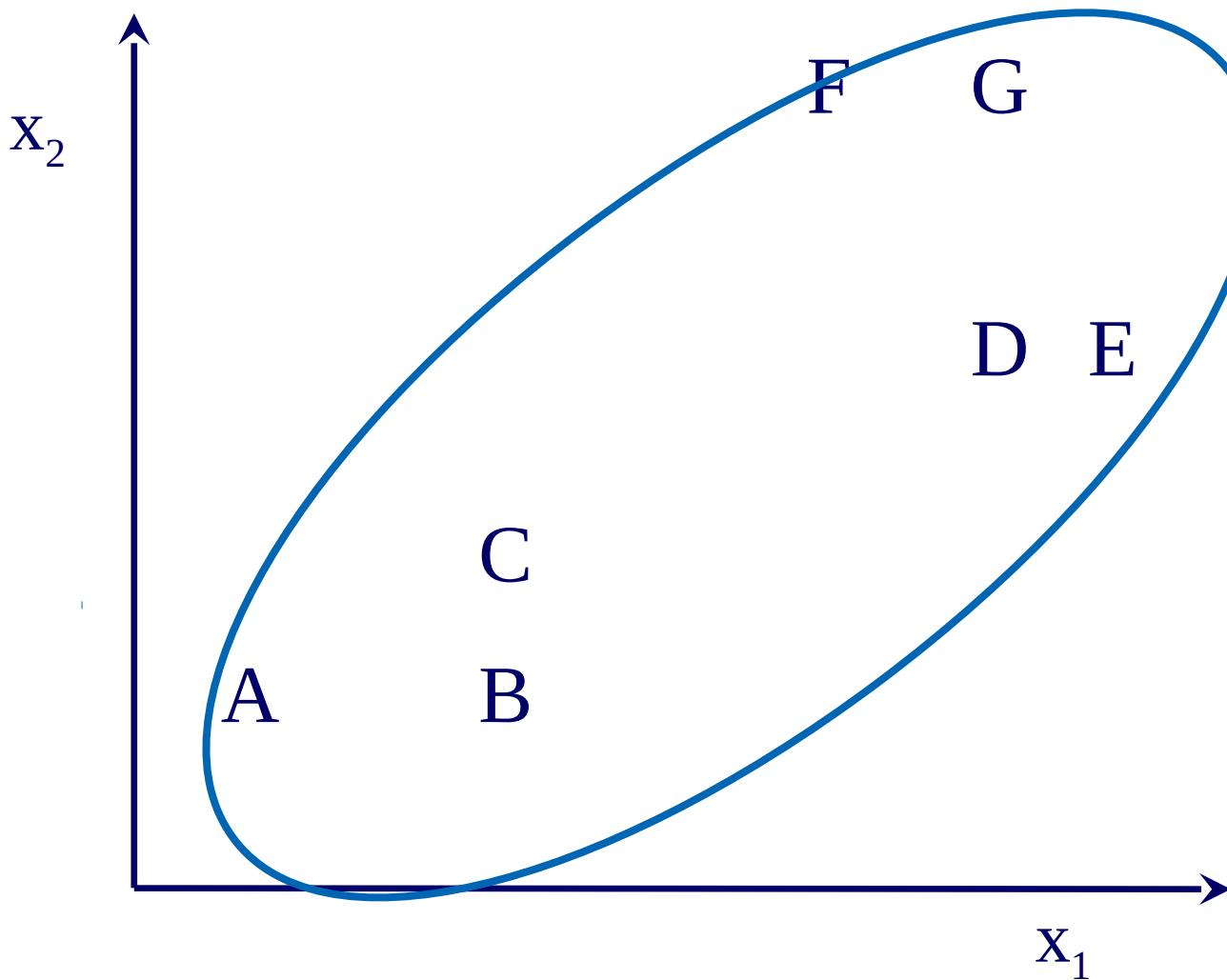
Metodi gerarchici

Livello 5: 2 clusters



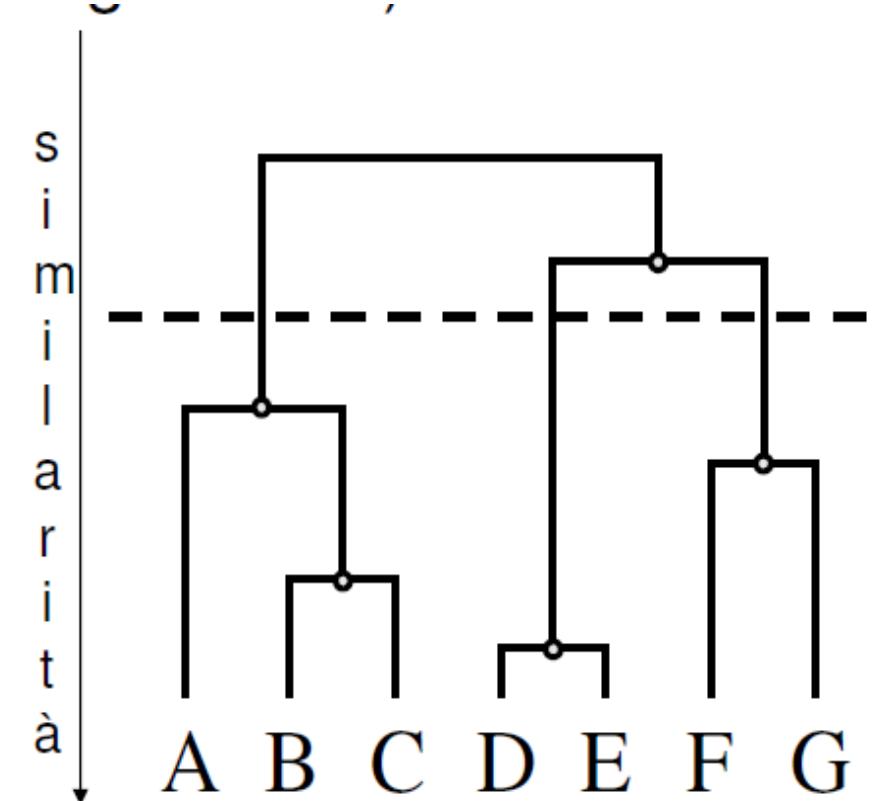
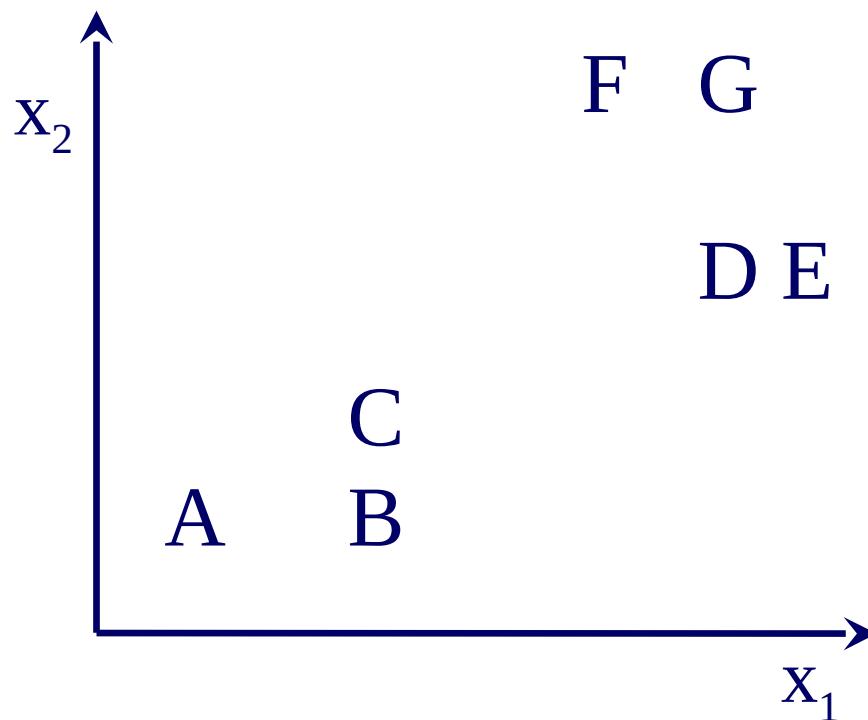
Metodi gerarchici

Livello 6 (più generale possibile): tutti gli oggetti sono nello stesso cluster



Metodi gerarchici

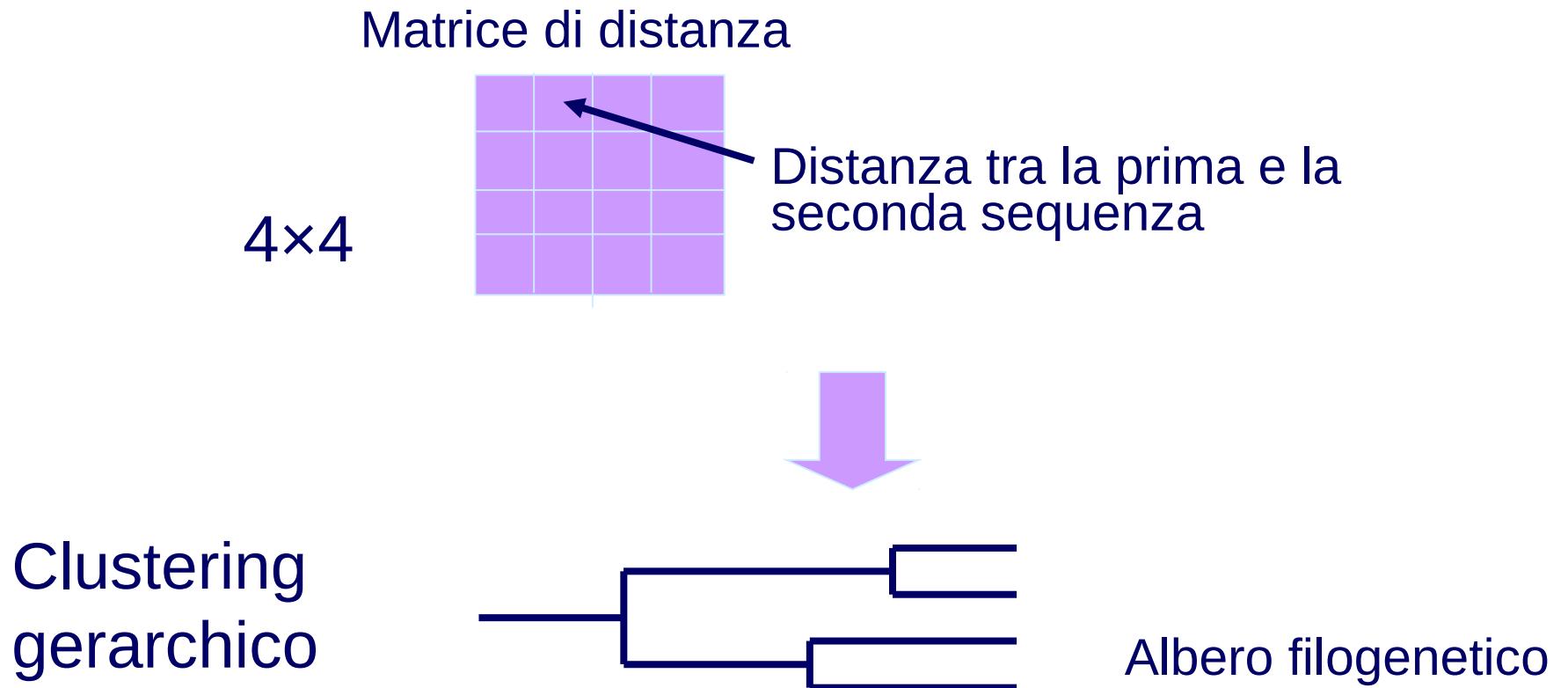
Le partizioni innestate possono essere rappresentate da un albero binario detto “dendrogramma”)



Esempi: Complete Link, Single Link, Ward, ...

Esempio: la filogenesi

- Clustering: data in ingresso la distanza tra tutte le coppie di sequenze



Validazione del clustering

- La validazione del clustering è fondamentale:
 - Ogni algoritmo di clustering genera SEMPRE un risultato: esiste qualche giustificazione per il clustering o i dati sono senza struttura?
 - Approcci differenti tipicamente portano a differenti clustering: i cluster che determino sono ottimali? E in che senso?
- **Problema:** non c'è il “ground truth”, il processo è non supervisionato

Esempio: la filogenesi

Analisi della robustezza del clustering: BOOTSTRAP

- Vengono creati N nuovi data set (per esempio 1000) campionando casualmente N colonne (con rimpiazzo)
⇒ in questo modo ogni dataset generato contiene lo stesso insieme di specie, con alcuni dei nucleotidi duplicati e con altri rimossi
- Per ogni data set viene costruito l'albero (clustering)
- Viene calcolata la frequenza con cui ogni gruppo/sottogruppo appare nei vari alberi
- Questa indica la robustezza di un raggruppamento (se un gruppo è trovato spesso, allora è significativo)

Dataset

0123456789
seqA ACCGTTCGGT
seqB ATGGTTCAGA
seqC ATCGATCGGA

Replicate 1

1562314951
seqA CTCCGCTTTC
seqB TTGGTTATT
seqC TTCCGTAATT

Replicate 2

5234924418
seqA TCGTTCTTCG
seqB TGGTAGTTTG
seqC TCGAACAAATG

Replicate 3

5607718907
seqA TCAGGCGTAG
seqB TCAAATGAAA
seqC TCAGGTGAAG

etc

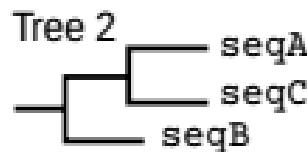
(a) Step 1

Assemble pseudo-datasets, repeat 1000 times



(b) Step 2

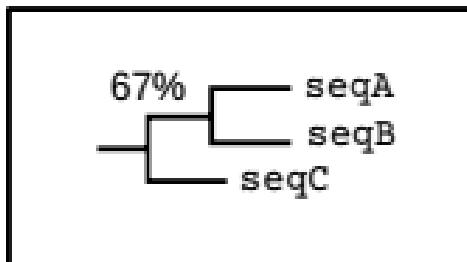
Build trees for each pseudo-dataset to give 1000 trees



etc

(c) Step 3

Tabulate results
(strict consensus tree)



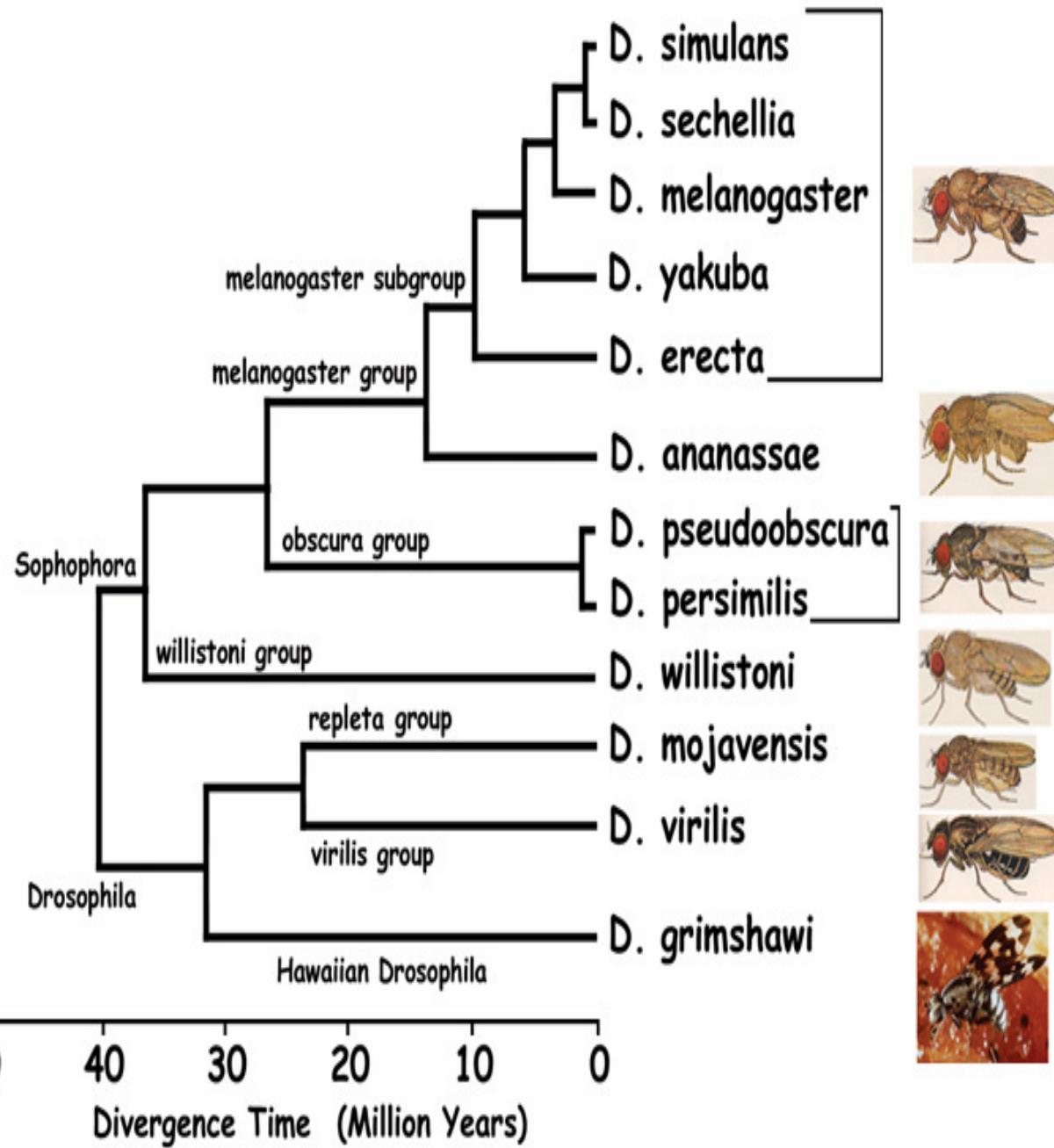
Bootstrap consensus tree

Interpretazione dei risultati

- L'obiettivo finale è quella di estrarre / recuperare conoscenza
 - ⇒ ottenere intuizioni dal data set
- Il fuoco deve essere sulla "interpretabilità" dei prodotti
 - ⇒ interpretabilità dei metodi
 - ⇒ mette a proprio agio l'utente
 - ⇒ interpretabilità delle soluzioni
 - ⇒ permette di capire gli errori

Spesso fondamentale per validare il clustering!

Esempio: la filogenesi



- *D. simulans* and *D. sechellia* sono più simili tra di loro che rispetto agli altri
- divergenza evolutiva più recente

Clustering: misure di similarità

Definizioni

- Coefficiente di similarità:
 - ⇒ indica la “forza” della relazione tra due oggetti
 - ⇒ maggiore è la somiglianza tra questi oggetti, più alto è il coefficiente di similarità
- Dissimilarità (distanza):
 - ⇒ concetto simile ma che misura le differenze tra due oggetti
- In generale si può parlare di “misure di prossimità”

\mathcal{X} Dominio del problema , $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$

$$proximity(\mathbf{x}_i, \mathbf{x}_j) = f : \mathcal{X}^2 \rightarrow \mathbb{R}$$

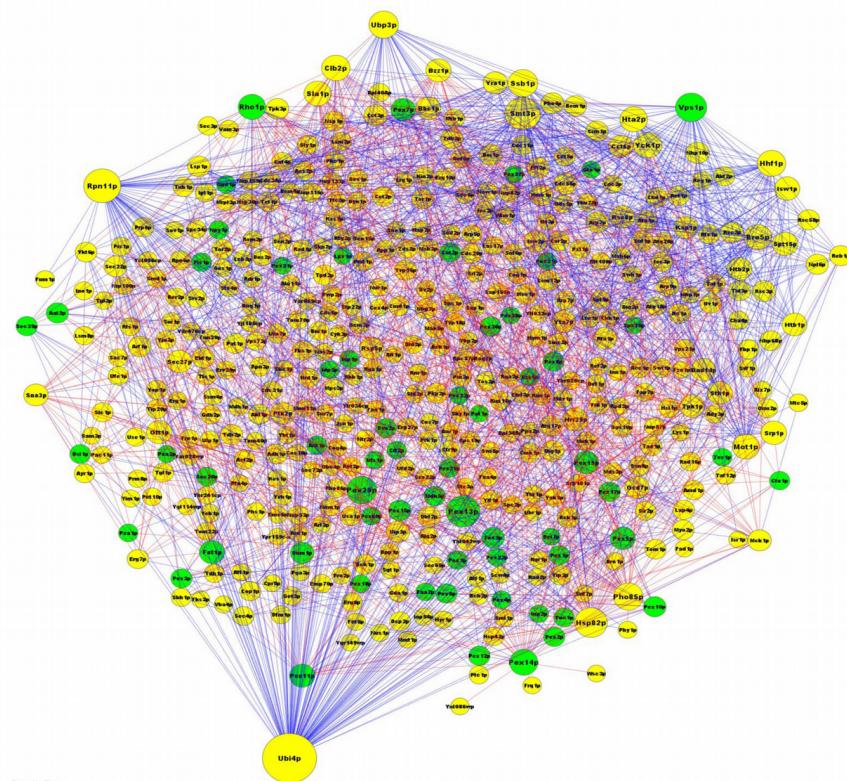
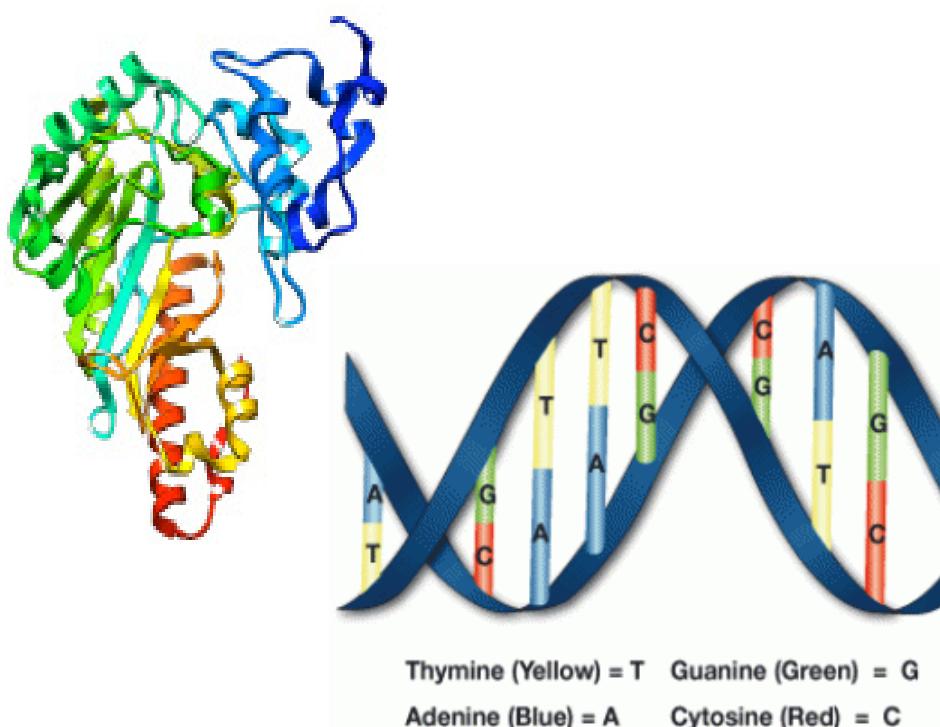
Definizioni

- Concetto di “metrica”: misura di prossimità con particolari caratteristiche
- Definizione: per una metrica (distanza) devono valere le seguenti proprietà:

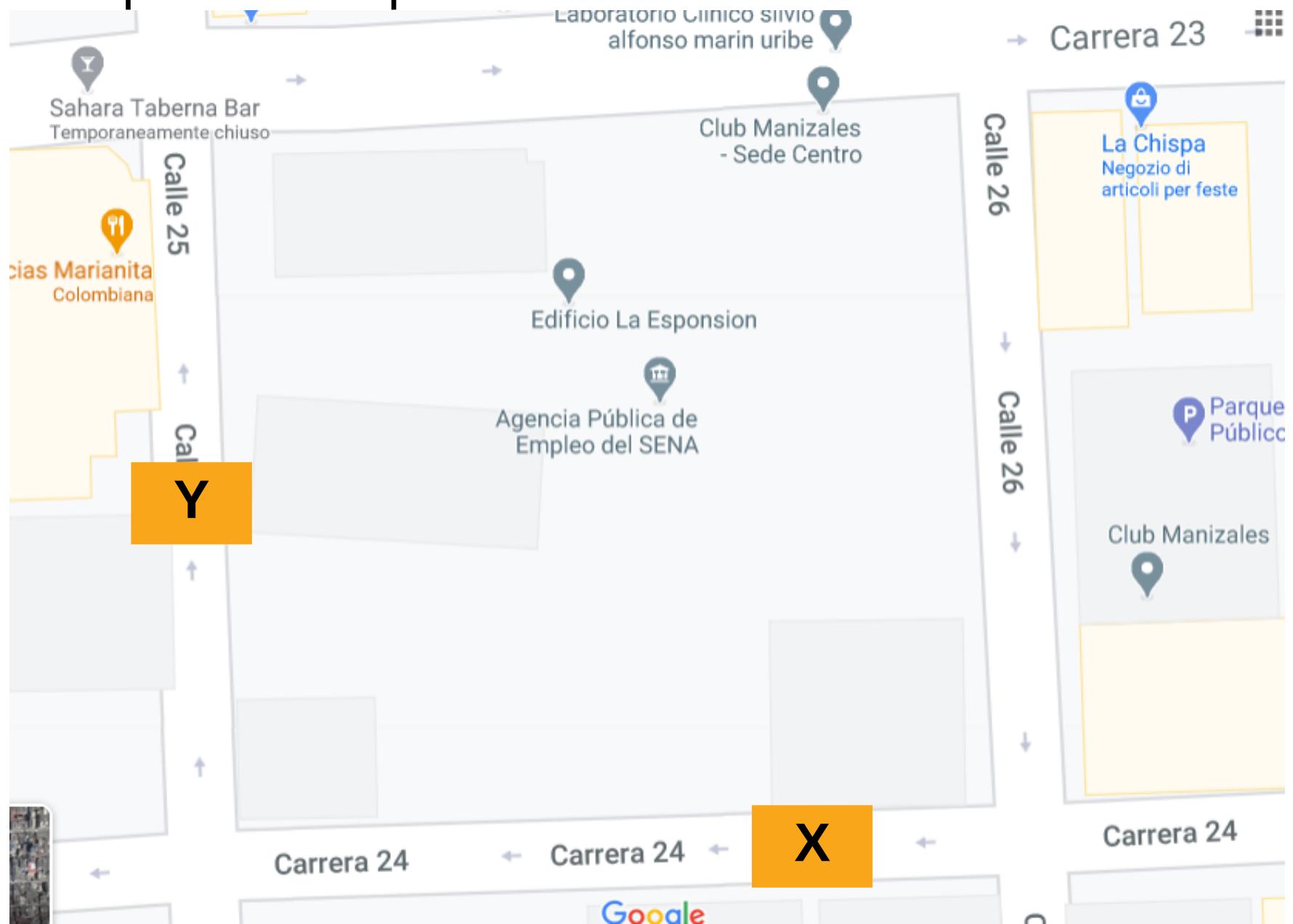
1. Positivity: $d_{ij} \geq 0$
2. Reflexivity: $d_{ii} = 0$
3. Definiteness: $d_{ij} = 0$ objects i and j are identical
4. Symmetry: $d_{ij} = d_{ji}$
5. Triangle inequality: $d_{ij} < d_{ik} + d_{kj}$

Misure non metriche

- Ci sono molte misure di prossimità che sono ragionevoli dal punto di vista dell'applicazione ma non soddisfano le proprietà delle metriche
 - Tipico in bioinformatica, specialmente quando si lavora con oggetti con struttura (sequenze, grafi etc)



Esempio: $d(X, Y)$ deve misurare la lunghezza del percorso che devo percorrere per andare da X a Y con la macchina



Problema: i sensi unici!!!

$d(X, Y)$: lunghezza del percorso verde



Ma quanto vale $d(Y,X)$?



$d(Y,X)$: lunghezza del percorso rosso



Misure non metriche

- La distanza tra X e Y è diversa dalla distanza tra Y e X!!
- Questo crea grossi problemi da un punto di vista metodologico o algoritmico
- MA: questa è una distanza “vera”, cioè una distanza che codifica perfettamente la vera natura del problema

Un esempio bioinformatico

- Distanza tra sequenze di DNA: inverso dell'alignment score basato sulla matrice di sostituzione

1 A C T G T A G G A A T C G C
↑ ↑ ↑
2 A A T G A A A G A A T C G C

- Questa misura è simmetrica se la matrice di sostituzione è simmetrica (“A → T” è uguale a “T → A”)

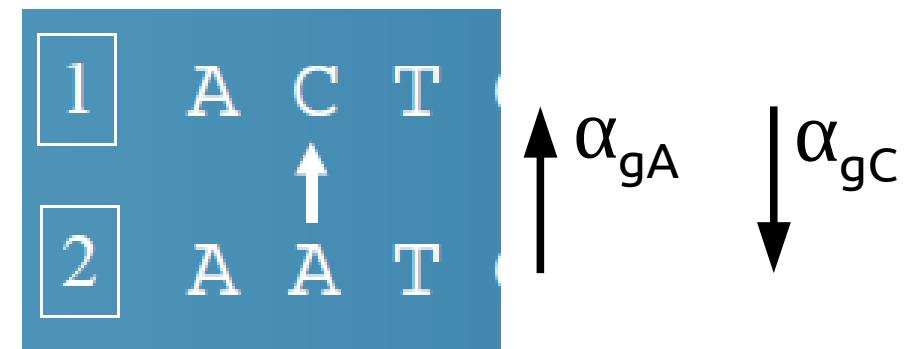
	A	T	C	G
A	-	α	α	α
T	α	-	α	α
C	α	α	-	α
G	α	α	α	-

Jukes-Cantor

Un esempio bioinformatico

- Se la matrice di sostituzione non è simmetrica, allora non è la stessa cosa allineare la sequenza 1 su 2 o allineare la sequenza 2 su 1
- Esempio: Distanza Tajima-Nei per sequenze di DNA
 - Pesa in modo diverso le sostituzioni tenendo conto della frequenza che i nucleotidi hanno all'interno delle sequenze

	A	T	C	G
A	-	α_{AT}	α_{AC}	α_{AG}
T	α_{TA}	-	α_{TC}	α_{TG}
C	α_{CA}	α_{CT}	-	α_{CG}
G	α_{GA}	α_{GT}	α_{GC}	-



Misure non metriche

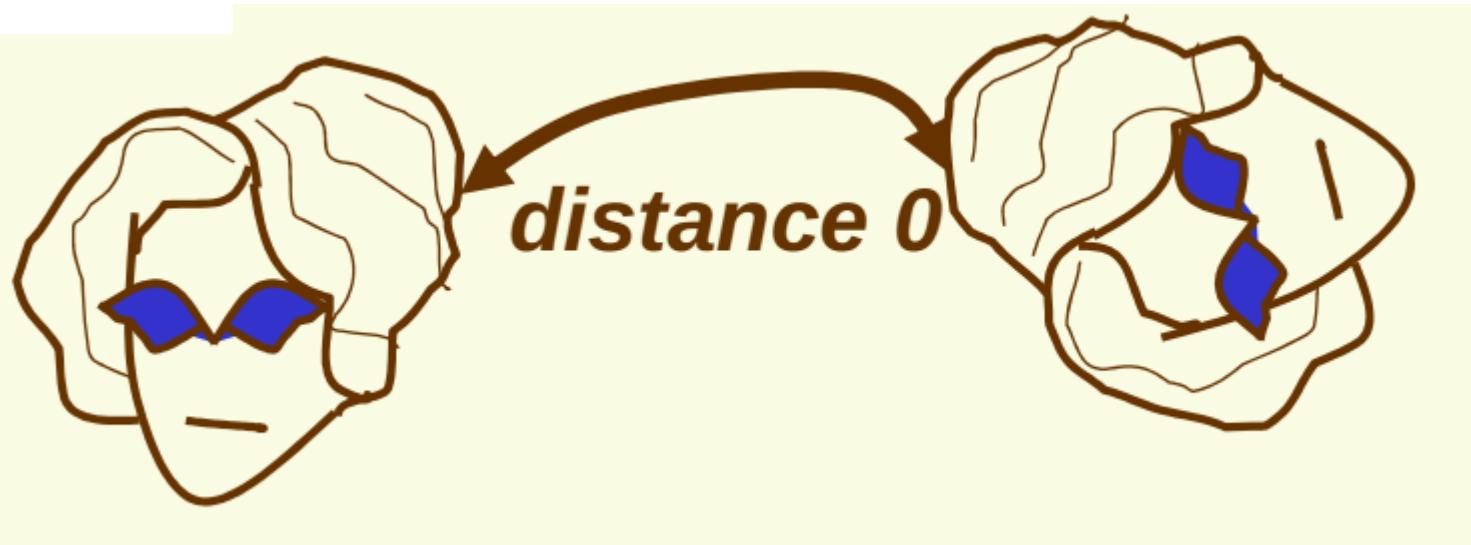
- Come detto prima, lavorare con queste misure non metriche non è “agevole” da un punto di vista metodologico o algoritmico
- Tuttavia è fondamentale disegnare algoritmi di pattern recognition basati su misure non metriche:
 - Sono molto espressive e descrivono perfettamente il contesto
 - A volte sono l'unica scelta: ci sono molti casi in cui è più facile misurare la relazione tra due oggetti che caratterizzarli con delle features
 - Esempio: Misurare la similarità tra due sequenze biologiche è facile (l'allineamento), caratterizzarle con features discriminanti è molto più difficile

Similarità: una scelta cruciale

- La scelta della misura di similarità/distanza è cruciale e influenza enormemente il risultato del clustering
- Occorre cercare di inglobare la maggior quantità possibile di informazione a priori:
 - ⇒ contesto applicativo
 - ⇒ tipo di pattern (vettore, sequenza, dati mancanti)
 - ⇒ dimensionalità del pattern
 - ⇒ scala
 - ⇒ cardinalità dell'insieme
 - ⇒ requisiti (velocità vs precisione): e.g. retrieval by content
 - ⇒ (esperienza del ricercatore)

La similarità deve essere invariante rispetto alle trasformazioni che sono “naturali” nel contesto applicativo

Esempio: se si comparano volti la distanza deve essere invariante alla rotazione



Per comparare caratteri, invece, non ci deve essere l'invarianza alla rotazione



Nota

- Similarità e dissimilarità misurano la stessa quantità da due punti di vista differenti
- Ci sono diversi modi per passare da similarità a dissimilarità

Esempi:

$d(x_i, x_j)$: distanza tra x_i e x_j

$$sim(x_i, x_j) = \frac{1}{d(x_i, x_j)}$$

Attenzione: scale diverse
(specialmente per distanze vicine allo zero!)

$$sim(x_i, x_j) = d_{max} - d(x_i, x_j) \text{ dove } d_{max} = \max_{ij} d(x_i, x_j)$$

Misure per pattern vettoriali

- Esistono molte misure diverse, a seconda che i dati siano numerici, categorici o binari
- Caso dati numerici: distanza euclidea, distanza di Manhattan, distanza Sup, distanza di Mahalanobis, distanza di Minkowski, misura coseno (similarità)

Vettori numerici

Dati due vettori $\mathbf{x} = [x_1 \dots x_L]$, $\mathbf{y} = [y_1 \dots y_L]$

- **Distanza euclidea**

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^L (x_j - y_j)^2} = [(\mathbf{x} - \mathbf{y})(\mathbf{x} - \mathbf{y})^T]^{1/2}$$

- **Distanza di Manhattan (city block distance)**

$$d(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^L |x_j - y_j|$$

⇒ Utilizzata nei circuiti dove i fili possono andare solo orizzontalmente o verticalmente

Vettori numerici

- **Maximum distance (distanza “sup”)**

$$d(\mathbf{x}, \mathbf{y}) = \max_{1 \leq j \leq L} |x_j - y_j|$$

- **Distanza di Mahalanobis**

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})\Sigma(\mathbf{x} - \mathbf{y})^T}$$

- ⇒ Effettua uno scalamento degli assi
- ⇒ Pro: invariante alle rotazioni/traslazioni/trasformazioni affini
- ⇒ Contro: calcolo della matrice di covarianza

Vettori numerici

- **Distanza di Minkowsky**

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^L (x_j - y_j)^p \right)^{\frac{1}{p}}$$

⇒ Generalizzazione della distanza euclidea ($p=2$) e di quella di Manhattan ($p=1$)

- **Similarità coseno**

$$d(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

⇒ Similarità (non distanza)

⇒ Tiene conto della lunghezza dei vettori

Misure per dati categorici

- Dati discreti (exe DNA)
⇒ simple matching dissimilarity measure: conta dove due sequenze sono diverse

$$\delta(x, y) = \begin{cases} 0 & \text{se } x = y \\ 1 & \text{se } x \neq y \end{cases}$$

$$d(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^L \delta(x_j, y_j)$$

Misure per dati binari

Dati binari: 0 o 1

- Distanza di Hamming: numero di posizioni dove i due vettori differiscono

$$d(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^L (x_j - y_j)^2$$

Esempio:

s₁: 1011101
s₂: 1001001

La distanza di Hamming è 2

Misure per dati binari

Similarità di Jaccard: misura del grado di overlap fra 2 insiemi A e B

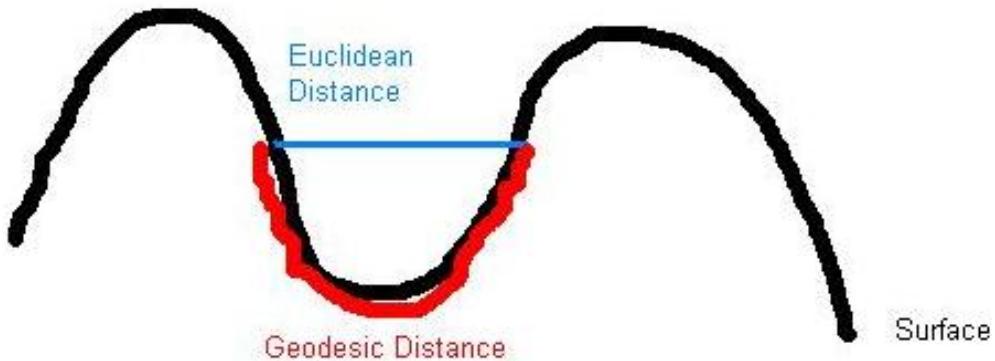
⇒ L'intersezione di A e B divisa per l'unione di A e B

$$Jsim(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

⇒ La distanza di Jaccard si ottiene facilmente facendo

$$Jdis(A, B) = 1 - Jsim(A, B)$$

Distanza geodesica



Tiene conto del “supporto”: il percorso che effettivamente occorre fare per raggiungere B partendo da A

- Misura più realistica, quindi più accurata
- Molto più difficile da misurare, occorre conoscere il supporto

Misure per pattern non vettoriali

- In caso di pattern non vettoriali occorre disegnare specifiche distanze
- Queste misure sono molto utilizzate in bioinformatica (molti oggetti si prestano ad essere descritti con rappresentazioni non vettoriali)
 - Tipicamente non soddisfano le proprietà della metrica
- Un esempio: Edit distance: la distanza tra due oggetti si misura come costo della trasformazione di un oggetto nell'altro

Edit Distance per sequenze

- Edit distance per sequenze: misura quante “modifiche” occorre effettuare sulla prima sequenza per ottenere l’altra
 - Sostituzioni, inserzioni, cancellazioni
- ESEMPIO: i simboli sono lettere, i pattern sono parole di un testo scritto.
- Possibili “modifiche”:
 - ⇒ sostituzioni: “pattern” → “pastern”
 - ⇒ inserzioni: “pattern” → “patterns”
 - ⇒ cancellazioni: “pattern” → “pttern”

Edit distance

- Ci sono molti modi per passare da una stringa all'altra. Ogni trasformazione T ha un costo, formato da tre componenti:
 - $C(T)$: costo per le cancellazioni in T
 - $I(T)$: costo per gli inserimenti in T
 - $R(T)$: costo per le sostituzioni in T
- l'Edit distance trova la trasformazione a costo minimo

$$D(S_1, S_2) = \min_{\text{all possible } T} (C(T) + I(T) + R(T))$$

Edit distance

- Distanza molto utilizzata per automatic editing e text retrieval (trovare il best match tra un pattern e un database di patterns)
- Molto utilizzata in bioinformatica per trovare la distanza tra due sequenze biologiche (i costi sono dati dalle matrici di sostituzione)
- L'algoritmo può essere disegnato in modo efficiente con la programmazione dinamica (dettagli nel cap 8.2.2 del Theodoridis)

Un esempio biologico: BLAST

- Basic Local Alignment Search Tool
 - ⇒ algoritmo per confrontare sequenze biologiche (nucleotidiche o aminoacidiche)
 - ⇒ confronta una sequenza di test con un database di sequenze, ritornando le più simili
 - ⇒ uno degli algoritmi più famosi di bioinformatica
 - ⇒ affronta un problema molto importante
 - ⇒ è computazionalmente efficiente -- la ricerca effettuata con algoritmi di programmazione dinamica è assolutamente inefficiente, vista la mole di dati presente oggigiorno – BLAST è 50 volte più veloce

Un esempio biologico: BLAST

- ⇒ IDEA: cerca di allineare due sequenze, lo score di allineamento rappresenta la misura della bontà del match
- ⇒ Assunzioni / Idee per velocizzare l'approccio
 - ⇒ non cercare l'allineamento “ottimale”
 - ⇒ non effettuare la ricerca in tutto lo spazio delle sequenze
 - ⇒ utilizzare una serie di euristiche per velocizzare l'approccio
- ⇒ Input dell'algoritmo:
 - ⇒ sequenza query (sequenza sconosciuta)
 - ⇒ sequenza target (o database)

Un esempio biologico: BLAST

PASSI dell'ALGORITMO

PASSO 1. Rimuovere le regioni di bassa complessità della sequenza query

- ⇒ regioni della sequenza con ripetizioni di pochi tipi di simbolo
- ⇒ possono confondere il programma nello trovare regioni significative

PASSO 2. Creare una lista delle “word” di N lettere della sequenza query

MDCCDC
MDCCDC
MDCCDC
MDCCDC
MDCCDC

MDCCDC
MDCCDC
MDCCDC
MDCCDC

MDCCDC
MDCCDC
MDCCDC

N = 2

N = 3

N = 4

Un esempio biologico: BLAST

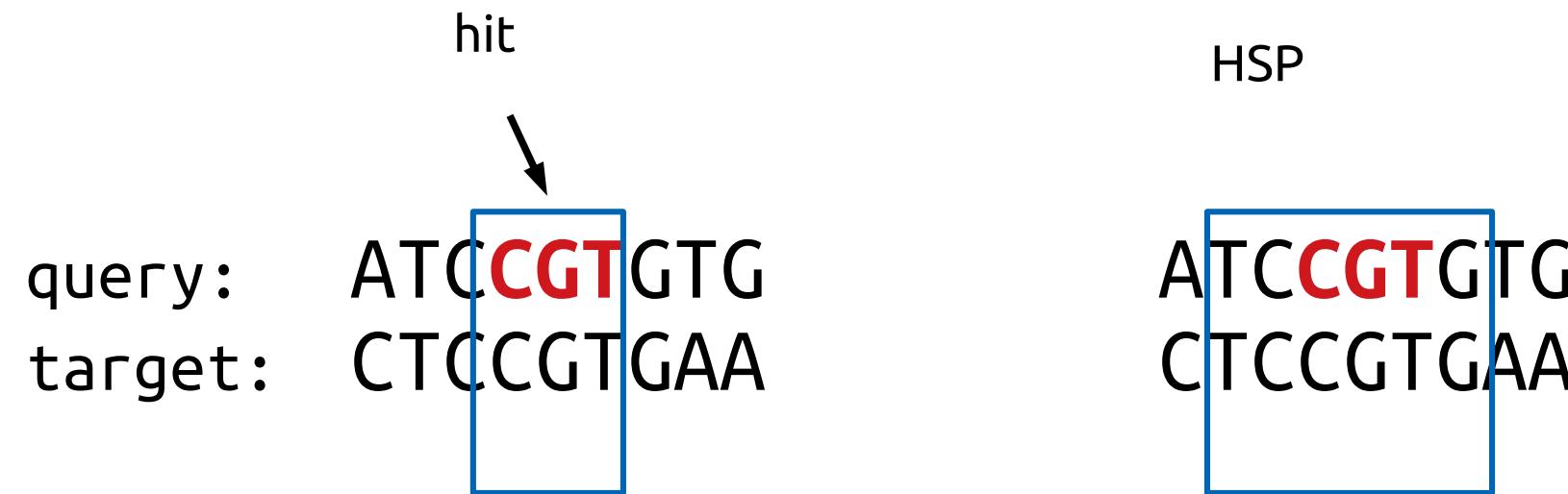
PASSO 3. cercare, in tutte le sequenze del database, tutte le word di lunghezza N che hanno un buon match con le word della sequenza query

- ⇒ buon match = score di allineamento sopra una certa soglia
- ⇒ utilizzo della “substitution matrix” per calcolare lo score
- ⇒ lo score considera l’allineamento senza gap
- ⇒ ogni word trovata si chiama “hit” (o “hotspot”)
- ⇒ allineamento senza gap è molto veloce: possibilità di memorizzare una volta per tutte le posizioni delle word in tutto il database

Un esempio biologico: BLAST

PASSO 4. utilizzare ogni “hit” come “seme” per allargare la regione di similarità

- ⇒ cercare di estendere la coppia di similarità a dx e a sx fino a quando lo score di similarità non diminuisce
- ⇒ il risultato si chiama HSP (High Scoring segment pair)



Lunghezza word = 3

Un esempio biologico: BLAST

PASSO 5. visualizzare tutti gli HSP con uno score sufficientemente alto

⇒ vengono listati in ordine di score

PASSO 6. fornire un'analisi statistica degli score risultanti: l'E-value

⇒ misura il numero di hit che si potrebbero vedere “per caso”, in un database di sequenze casuali

⇒ dipende dalla dimensionalità del database e dalla lunghezza della sequenza di query

⇒ la significatività statistica è proporzionale al valore di tale indice (valori attorno allo zero supportano fortemente i risultati)

Un esempio biologico: BLAST

Note finali:

- ⇒ Eventualmente si può gestire anche la presenza di più HSP in una stessa sequenza del database
- ⇒ si può utilizzare on line:
<http://blast.ncbi.nlm.nih.gov/Blast.cgi>
- ⇒ utilizzatissimo per il buon compromesso tra accuratezza e velocità (negli anni sono state presentate molte varianti)
 - ⇒ l'articolo dove viene presentato è il più citato degli anni 90

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990). "Basic local alignment search tool". *J Mol Biol* **215** (3): 403–410