Ricerca di motivi nei grafi

Applicazione a reti metaboliche

Giuseppe Luca Tomaino e Ilaria Clara Urciuoli

Università di Pisa
http://www.unipi.it

4 giugno 2009

Summary

Il metabolismo

Il metabolismo

Insieme delle reazioni chimiche impiegate nella sintesi e degradazione di piccole molecole (le sostanze di cui un organismo è composto) dalle quali ottiene l'energia necessaria per la crescita

Primario: rappresenta il motore energetico principale; reazioni riscontrabili in tutti gli organismi

Secondario: non essenziale per la semplice crescita, sviluppo o riproduzione dell'organismo: di natura ecologica

Il metabolismo

Elementi essenziali del metabolismo:

- Metaboliti
- Reazioni biochimiche
- Enzimi

Metabolita

Composto coinvolto in una reazione biochimica del metabolismo: puo' costituire il substrato della reazione o un prodotto (intermedio o finale)

Il metabolismo

Reazioni biochimiche:

Anabolismo: composti semplici + energia → macromolecole complesse

Catabolismo: macromolecole complesse → composti semplici + energia

Vie metaboliche o Pathway

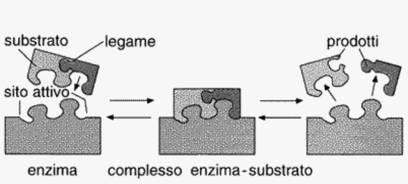
Insieme di reazioni chimiche che avvengono in modo sequenziale, e portano alla produzione di un prodotto finale e di prodotti intermedi (esempio: biosintesi della valina).

L'insieme dei pathway costituisce una rete metabolica

Enzimi

Gli enzimi

Struttura molecolare di natura proteica che catalizza una reazione chimica grazie all'interazione tra il suo *sito attivo* e il substrato che da origine al *complesso enzima-substrato*

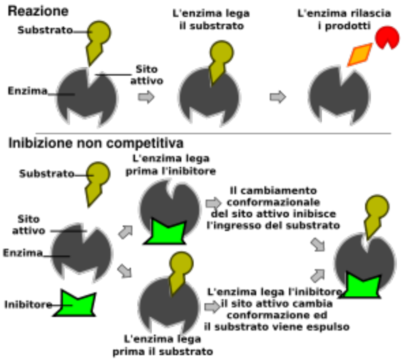


L'enzima non si consuma durante la reazione

Il prodotto viene allontanato dall'enzima che è disponibile per una nuova reazione

Enzimi

Enzimi allosterici: sono provvisti anche di un *sito allosterico* cui si lega una molecola che modifica la forma dell'enzima inibendo o facilitando la creazione di un complesso enzima-substrato



Il prodotto finale della via metabolica opera un meccanismo di regolazione a retroazione o feedback

Omeostasi: capacità di mantenere costanti le condizioni chimico-fisiche interne anche al variare delle condizioni ambientali esterne

Gli enzimi

L' International Union of Biochemistry and Molecular Biology ha creato una commissione (**Enzyme Commission**, EC) per rinominare e numerare gli enzimi

Principi

1. ogni nome deve identificare singoli enzimi e non possono essere applicati a sistemi contenenti più di un enzima
2. numero e nome sono assegnati in base alla reazione catalizzata dall'enzima;
3. gli enzimi sono divisi in gruppi sulla base dei tipi di reazione catalizzata e insieme ai nomi dei substrati sono le basi della nomenclatura e della numerazione

La numerazione EC

Il numero EC indica uno specifico enzima in relazione alla **reazione** che è in grado di catalizzare.

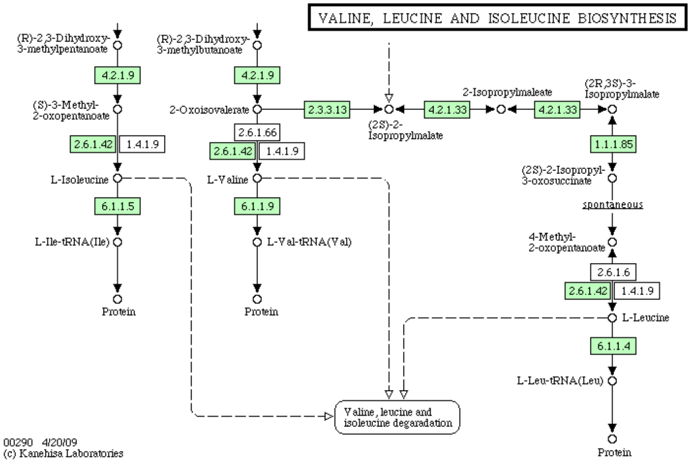
Numerazione gerarchica:

$$\text{EC } x.y.z.t$$

- x tipo di reazione catalizzata (es. ossidoriduttasi, trasferasi, idrolasi)
- y sottoclasse di reazione
- z sotto-sottoclasse di reazione
- t numero seriale per l'enzima nella sotto-sottoclasse

Un enzima puo' catalizzare piu' di una reazione e una reazione puo' essere catalizzata da piu' di un enzima

Esempio di via metabolica



Fonte: KEGG Pathway

KEGG Pathway

KEGG PATHWAY e' una collazione di vie metaboliche disegnate a mano che rappresentano la conoscenza attuale.

Contiene informazioni sul metabolismo di 209 organismi

Sono rappresentate solo le reazioni tra metaboliti primari

Reti metaboliche come grafi

Grafo

Definito come coppie (V, E) dove V è l'insieme di *vertici* e $E \subseteq VxV$ è l'insieme degli *archi*

Tipi di grafo:

- grafo bipartito: $V =$ composti $\cup$ reazioni
- grafo dei composti $V =$ composti (reazioni: etichette di E)
- grafo delle reazioni $V =$ reazioni (composti: etichette di E)

Grafo G

Grafo delle reazioni *non direzionato* e con etichette per i vertici

Ogni vertice di G sara' etichetto con 1 o piu' elementi dell'insieme C (chiamati colori)

C è l'insieme degli EC number

Direzionalita'

Il verso di una reazione dipende dalla sua reversibilità
Informazione non sempre conosciuta e spesso incongruente all'interno dello stesso database

Funzione di similarità tra reazioni: S

Per calcolare la similarita' tra le reazioni si considera la similarita' tra gli enzimi che le catalizzano:

- allineamento delle sequenze proteiche che compongono l'enzima
- confronto degli EC number degli enzimi

Funzione di similarità

Assegna come score il valore corrispondente al livello più profondo al quale due EC number sono uguali.
Es. $S(1.1.1.2,\ 1.1.1.3) = 3$

Valore di cut-off s *in* $[0 \dots 4]$

Definizione di Motivo

Motivo

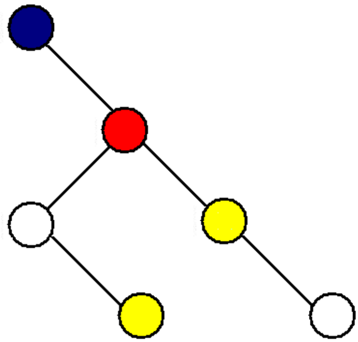
Un motivo M e' un insieme di elementi da un insieme C di colori

Nessun limite topologico: non viene specificato l'ordine in cui devono comparire i colori

Vantaggio: possiamo aggiungere successivamente restrizioni di tipo topologico e misurarne l'impatto

Definizione di Motivo: esempio

$$M = \{blu,\ rosso,\ giallo,\ bianco\}$$



Limite di colore: 3 match
Limite topologico: 1 match

Definizione di Occorrenza

Insieme connesso di vertici etichettati con i colori del motivo

Formalmente

R sottinsieme dei nodi di G

M motivo con $|M| = |R|$

$H(R, M)$ denota grafo bipartito in cui

- $R \cup M$ insieme dei vertici di H
- c'è un arco tra un vertice v di R e un vertice c di M sse v ha c tra i suoi colori

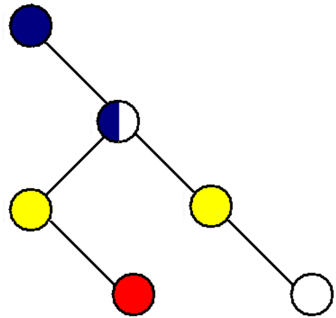
Definizione di Occorrenza esatta

Occorrenza esatta

Un'occorrenza esatta del motivo M e' un insieme R di vertici di G tale che $H(R, M)$ è un match perfetto e R induce un sottografo connesso in G

Definizione di Occorrenza esatta: esempio

$$M = \{blu,\ blu,\ giallo,\ rosso\}$$



1 match

Definizione di Occorrenza approssimata

lb numero massimo di gap locali

gb numero massimo di gap globali

s_c valore di cut-off, $\forall c \in M$

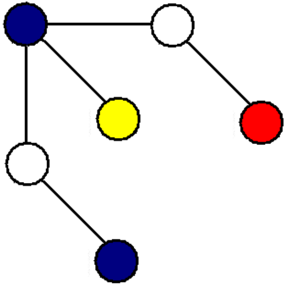
Occorrenza approssimata

Un'occorrenza approssimata di M (che rispetti lb, gb, e la soglia s_c) e' un insieme R di vertici contenuto in un insieme R' di vertici di G tale che:

1. per il grafo bipartito $H(M \cup R, E_H)$ con $E_H = \{\{c, v\} \in MxR|$ esiste un colore c' per il quale $S(c', c) \geq s_c\}$, R e' un perfetto match;
2. per ogni sottoinsieme B di R tale che $B \neq 0$ e R tolto $B \neq 0$ la lunghezza del percorso più breve in $G_{R'}$ tra B e R tolto B e' al piu' lb;
3. $|R'| - |R| \leq gb$.

Definizione di Occorrenza approssimata: esempio

$$M = \{blu,\ blu,\ giallo,\ rosso\}$$

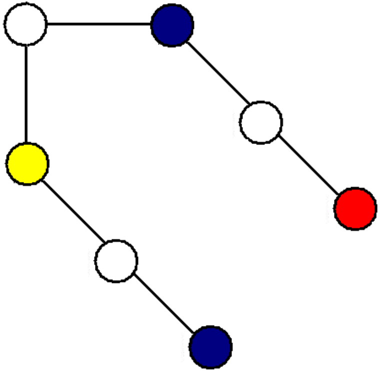


$lb = 1$
$gb = 2$
$s = 4$

1 match

Definizione di Occorrenza approssimata: esempio

$$M = \{blu,\ blu,\ giallo,\ rosso\}$$



$lb = 1$
$gb = 2$
$s = 4$

non match

$gb = 3 \longrightarrow$ match

Introduzione al problema

Problema di ricerca

Dato un motivo M e un grafo G etichettato e non direzionato, trovare tutte le occorrenze di M in G.

- Nessuna restrizione topologica in $M \Longrightarrow$ problema $\neq$ isomorfismo tra sottografi

Problema di esistenza

Dato un motivo M e un grafo G etichettato e non direzionato, stabilire se M occorre in G.

- Complessità NP-C

Introduzione al problema

Problema di ricerca

Dato un motivo M e un grafo G etichettato e non direzionato, trovare tutte le occorrenze di M in G.

- Nessuna restrizione topologica in $M \Longrightarrow$ problema $\neq$ isomorfismo tra sottografi

Problema di esistenza

Dato un motivo M e un grafo G etichettato e non direzionato, stabilire se M occorre in G.

- Complessità NP-C

Introduzione al problema

Problema di ricerca

Dato un motivo M e un grafo G etichettato e non direzionato, trovare tutte le occorrenze di M in G.

- Nessuna restrizione topologica in $M \Longrightarrow$ problema $\neq$ isomorfismo tra sottografi

Problema di esistenza

Dato un motivo M e un grafo G etichettato e non direzionato, stabilire se M occorre in G.

- Complessità NP-C

Introduzione al problema

Problema di ricerca

Dato un motivo M e un grafo G etichettato e non direzionato, trovare tutte le occorrenze di M in G.

- Nessuna restrizione topologica in $M \Longrightarrow$ problema $\neq$ isomorfismo tra sottografi

Problema di esistenza

Dato un motivo M e un grafo G etichettato e non direzionato, stabilire se M occorre in G.

- Complessità NP-C

Dimostrazione

Proposizione

Il "Problema di esistenza" è NP-C anche se G è un albero

Dimostrazione

Utilizzata una semplificazione dell'EXACT COVER BY 3-SETS (X3C) $\Longrightarrow$ Complessità NP-C

Istanza:

- un insieme $X = \{1, 2, \ldots, 3q\}$ con $|X| = 3q$
- una collezione di sottoinsiemi $C = \{C_1, C_2, \ldots, C_n\}$ dove tutti gli elementi di C_i appartengono ad X e $|C_i|$=3 , $\forall i = 1, \ldots, n$
- un motivo $M = \{Y, B, \ldots, B, 1, \ldots, 3q\}$
- un albero T

Dimostrazione

Proposizione

Il "Problema di esistenza" è NP-C anche se G è un albero

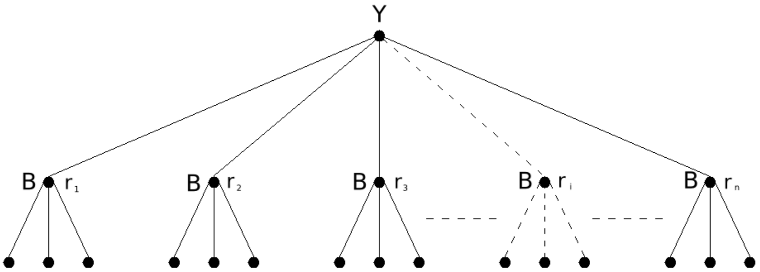
Dimostrazione

Utilizzata una semplificazione dell'EXACT COVER BY 3-SETS (X3C) $\Longrightarrow$ Complessità NP-C

Istanza:

- un insieme $X = \{1, 2, \ldots, 3q\}$ con $|X| = 3q$
- una collezione di sottoinsiemi $C = \{C_1, C_2, \ldots, C_n\}$ dove tutti gli elementi di C_i appartengono ad X e $|C_i|$=3 , $\forall i = 1, \ldots, n$
- un motivo $M = \{Y, B, \ldots, B, 1, \ldots, 3q\}$
- un albero T

Dimostrazione



Dimostrazione

Definiamo una sotto-collezione $C' \subseteq C$ avente la proprietà che ogni elemento di X occorre esattamente in un solo elemento di C'

$$\exists C' \iff M \text{ occorre in } T$$

$\exists C' \implies M$ *occorre in* T

$|C'| = q$.

Sia R l'insieme dei nodi di T che consiste in un nodo marcato Y e 4 nodi per ogni elemento di C'.

In R saranno presenti anche q nodi marcati B ed uno marcato da ogni elemento in X.

$\implies R$ è un'occorrenza di M in T

$\exists C' \Longleftarrow M$ *occorre in* T

Esiste un insieme R (con $|R| = 1 + 4q$) che induce un sottografo di T connesso ed ha un nodo marcato da ognuno dei colori in M.

In R è inoltre presente il nodo marcato Y

Una foglia da un elemento di C' è in R se e solo se il corrispondente r_i e' anche in R.

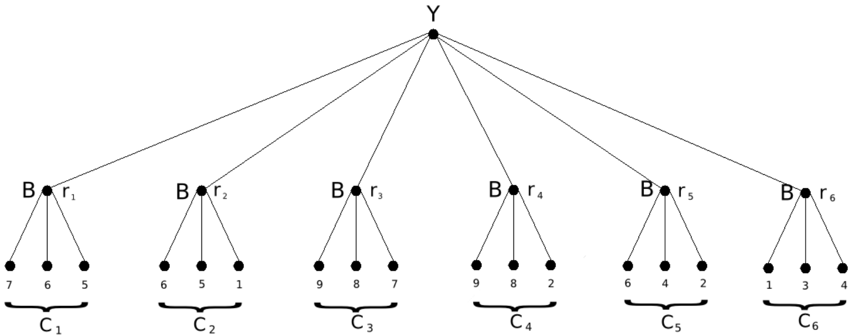
$|C'| = q$ (R deve contenere esattamente q nodi marcati B).

Le tre foglie di ogni elemento di C' devono essere in R ($|R| = 1 + 4q$).

R deve contenere un nodo marcato per ogni elemento di X.

$\Longrightarrow$ esiste l'insieme C' contenente ogni elemento di X.

Esempio



Albero T e le sue etichette per $X = \{1, \dots, 9\}$
$C = \{\{7,6,5\}, \{6,5,1\}, \{9,8,7\}, \{9,8,2\}, \{6,4,2\}, \{1,3,4\}\}$
In questo esempio $M = \{Y, B, B, B, 1, \dots, 9\}$

Trattabilita'

Dato che "Problema di esistenza" e' NP-C anche se G e' un albero, il "Problema di ricerca" e' NP-Hard

Trattabilita' a parametro fisso

- Il "Problema di ricerca" pur essendo NP-Hard e' trattabile a parametro fisso se G e' un albero con parametro k la lunghezza del motivo.
- Per un generico grafo G invece la complessita' e' sempre NP-C.

Trattabilita'

Dato che "Problema di esistenza" e' NP-C anche se G e' un albero, il "Problema di ricerca" e' NP-Hard

Trattabilita' a parametro fisso

- Il "Problema di ricerca" pur essendo NP-Hard e' trattabile a parametro fisso se G e' un albero con parametro k la lunghezza del motivo.
- Per un generico grafo G invece la complessita' e' sempre NP-C.

Trattabilita'

Dato che "Problema di esistenza" e' NP-C anche se G e' un albero, il "Problema di ricerca" e' NP-Hard

Trattabilita' a parametro fisso

- Il "Problema di ricerca" pur essendo NP-Hard e' trattabile a parametro fisso se G e' un albero con parametro k la lunghezza del motivo.
- Per un generico grafo G invece la complessita' e' sempre NP-C.

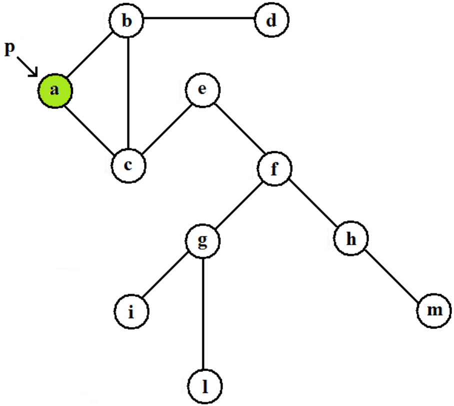
Risultati di complessita'

TABLE 1
Complexity Results for the Motif Search Problem

Input Graph / Motif		Tree	Arbitrary
Topological motifs		polynomial	NP-complete
Colored topological motifs	General case	polynomial	NP-complete
	Fixed colors and no repetition	polynomial	polynomial (conjecture)
Colored motifs (this paper)		NP-complete, FPT in k	NP-complete

Nella pratica le reti metaboliche sono grafi e non alberi, ma essendo relativamente piccole (3184 nodi e 17642 archi per una rete costruita a partire dal database di KEGG) è concepibile tentare di risolverlo applicando opportune potature nonostante il problema sia NP-C.

Senza gap:

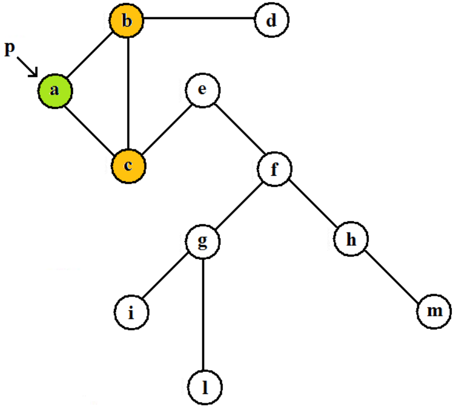


INPUT : $|M| = 3$

Inizializzazione:

$Q = \{a\}$
$p = a$

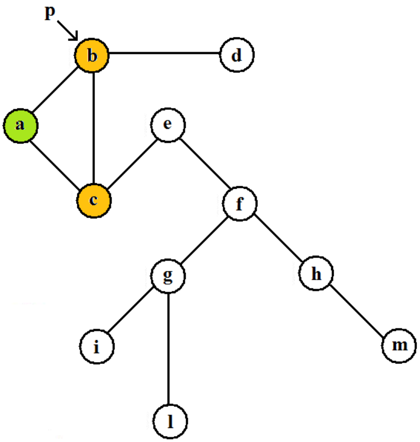
Senza gap:



Iterazione 1:

R = {a}
V[a] = {b, c}
Q = {a, b, c}

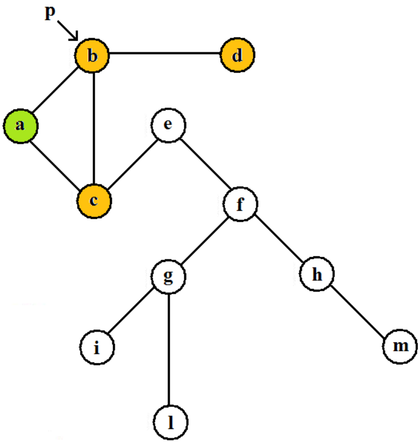
Senza gap:



Iterazione 1:

R = {a}
V[a] = {b, c}
Q = {a, b, c}
p = b

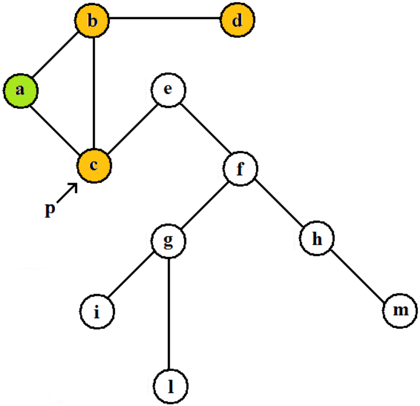
Senza gap:



Iterazione 2:

$R = \{a, b\}$
$V[b] = \{d\}$
$Q = \{a, b, c, d\}$

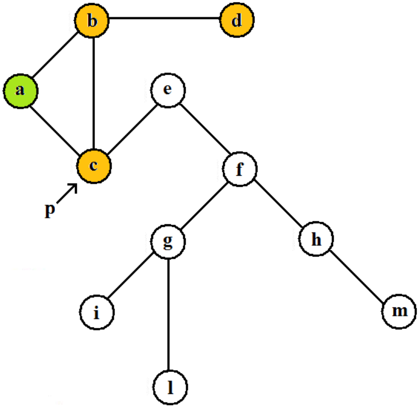
Senza gap:



Iterazione 2:

$R = \{a, b\}$
$V[b] = \{d\}$
$Q = \{a, b, c, d\}$
$p = c$

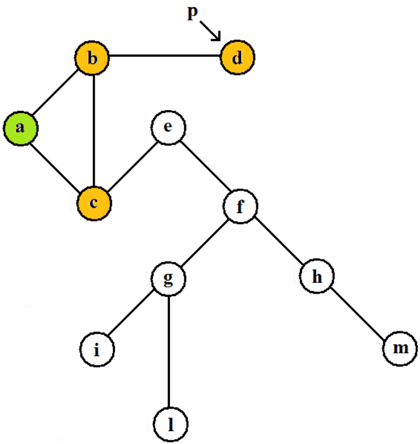
Senza gap:



Iterazione 3:

R = {a, b, c}
OCC = {{a, b, c}}
R = {a, b}

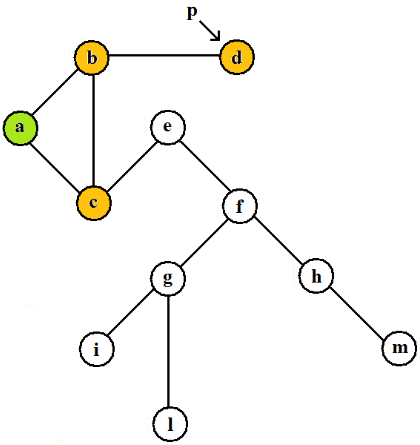
Senza gap:



Iterazione 3:

R = {a, b, c}
OCC = {{a, b, c}}
R = {a, b}
p = d

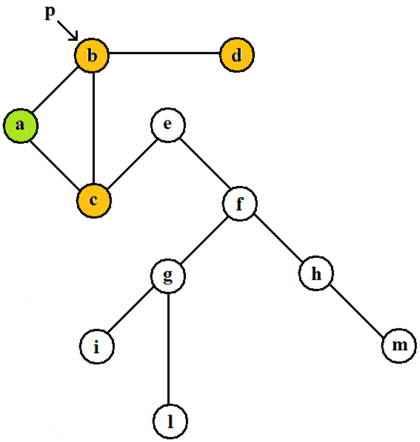
Senza gap:



Iterazione 4:

R = {a, b, d}
OCC = {{a, b, c}}
R = {a, b}

Senza gap:

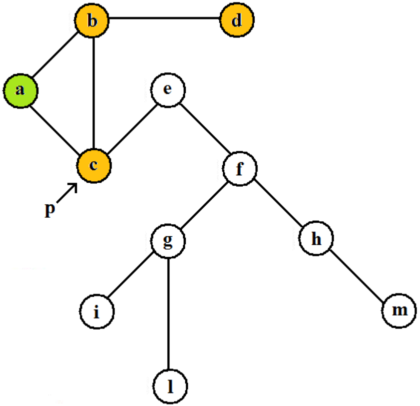


Iterazione 4:

R = {a, b, d}
OCC = {{a, b, c}}
R = {a, b}

Backtracking
p = b
R = {a}
Q = {a, b, c}

Senza gap:

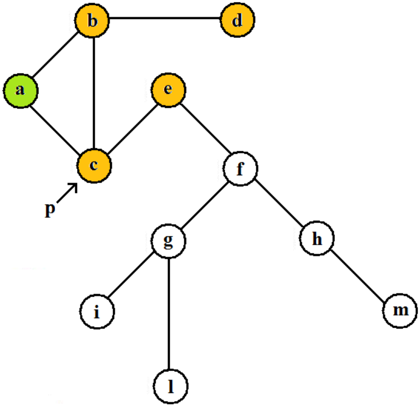


Iterazione 4:

R = {a, b, d}
OCC = {{a, b, c}}
R = {a, b}

Backtracking
p = b
R = {a}
Q = {a, b, c}
p = c

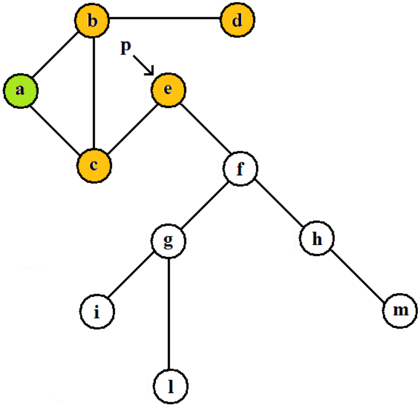
Senza gap:



Iterazione 5:

R = {a, c}
V[c] = {e}
Q = {a, b, c, e}

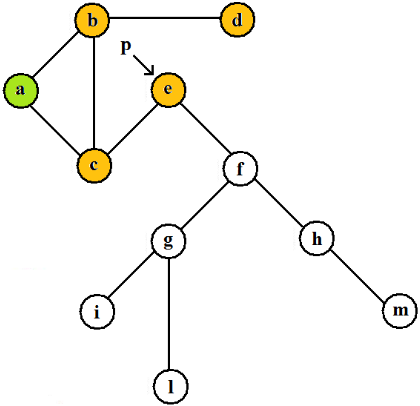
Senza gap:



Iterazione 5:

R = {a, c}
V[c] = {e}
Q = {a, b, c, e}
p = e

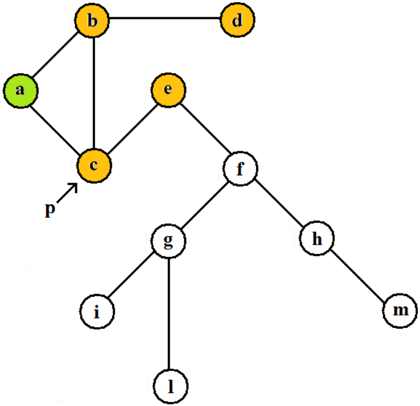
Senza gap:



Iterazione 6:

R = {a, c, e}
OCC = {{a, b, c}}
R = {a, c}

Senza gap:

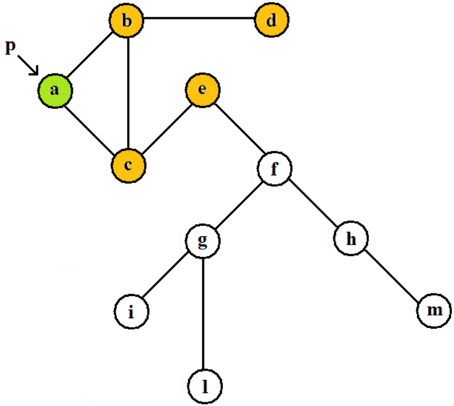


Iterazione 6:

R = {a, c, e}
OCC = {{a, b, c}}
R = {a, c}

Backtracking
p = c
R = {a}
Q = {a, b, c}

Senza gap:

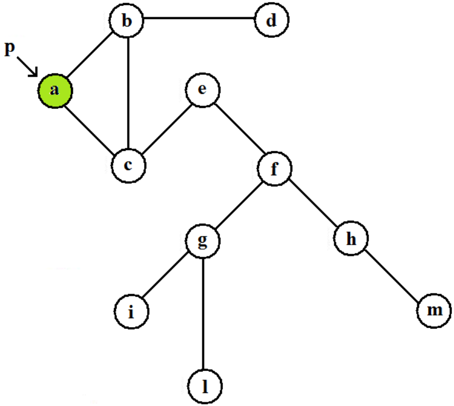


Iterazione 6:

R = {a, c, e}
OCC = {{a, b, c}}
R = {a, c}

Backtracking
p = c
R = {a}
Q = {a, b, c}
p = a
R = {}

Senza gap:



Iterazione 6:

R = {a, c, e}
OCC = {{a, b, c}}
R = {a, c}

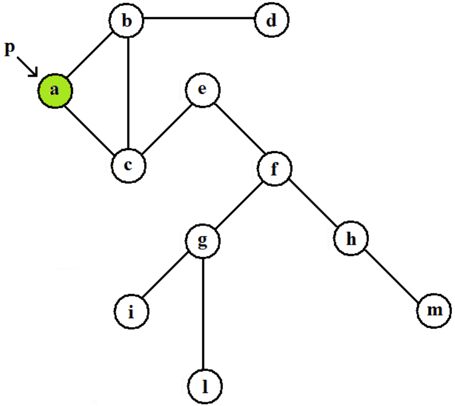
Backtracking
p = c
R = {a}
Q = {a, b, c}
p = a
R = {}
Q = {a} STOP

Senza gap: pseudocodice

```
procedure noGaps  (   INPUT: grafo G = <N,A> , motivo M
                      OUTPUT: tutte le occorrenze di M in G    )

begin {
    OCC = {} //Occorrenze di M in G
    Q = {} //coda
    V[] = {} //insieme dei vicini di un nodo
    R = {} //insieme candidato come occorrenza
    p = null //puntatore a nodo in N
    foreach nodo n appartenente N  {
            Q = {}
            Q = Q unito {n}
            p = n
            do {
                R = R unito {p}
                if (|R| == |M|) {
                    if (R è un match con M)
                        OCC unito R
                    R = R tolto {p}
                }
                else {
                    V[p] = {nodi vicini a P non presenti in Q}
                    Q = Q unito V[p]
                }
                while (p == ultimo elemento di Q && p !=n) {
                    p = ultimo elemento in R
                    R = R tolto {p}
                    Q = Q tolto V[p]
                }
                p = successivo elemento in Q;
            } while (Q non contiene solo n)
    }
    return OCC
}
```

Con gap:

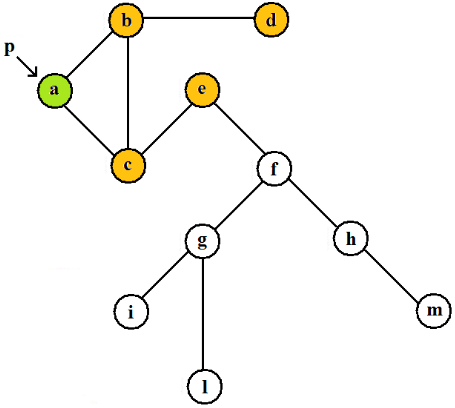


$$\text{INPUT} : |M| = 3$$
$$lb = 1$$
$$gb = 2$$

Inizializzazione:

$Q = \{a\}$
$p = a$

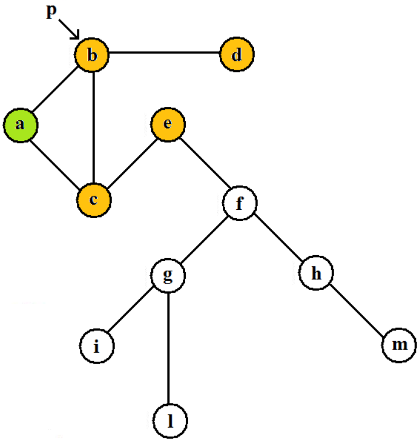
Con gap:



Iterazione 1:

R = {a}
S = {}
d = 0
dtot = 0
DIST[a] = 0
V[a] ={b,c,e,d}
Q = {a,b,c,e,d}

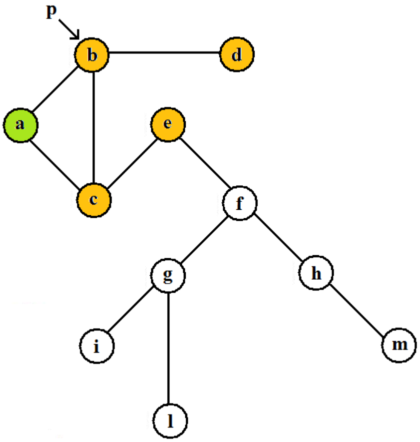
Con gap:



Iterazione 1:

R = {a}
S = {}
d = 0
dtot = 0
DIST[a] = 0
V[a] ={b,c,e,d}
Q = {a,b,c,e,d}
p = b

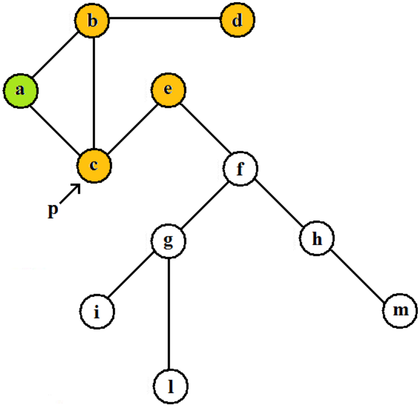
Con gap:



Iterazione 2:

R = {a, b}
S = {a}
d = 0
dtot = 0
DIST[b] = 0
V[b] ={}
Q = {a,b,c,e,d}

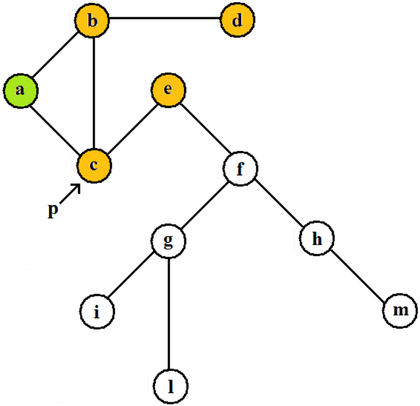
Con gap:



Iterazione 2:

R = {a, b}
S = {a}
d = 0
dtot = 0
DIST[b] = 0
V[b] ={}
Q = {a,b,c,e,d}
p = c

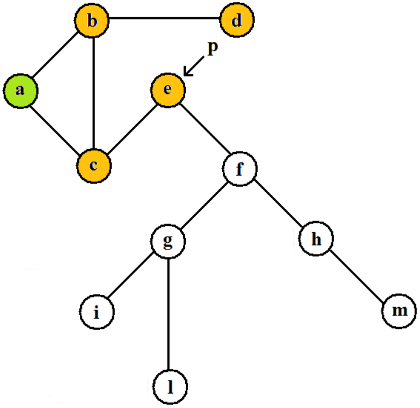
Con gap:



Iterazione 3:

R = {a, b, c}
OCC = {}
R = {a, b}

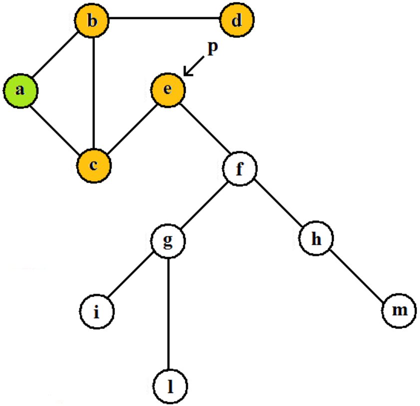
Con gap:



Iterazione 3:

R = {a, b, c}
OCC = {}
R = {a, b}
p = e

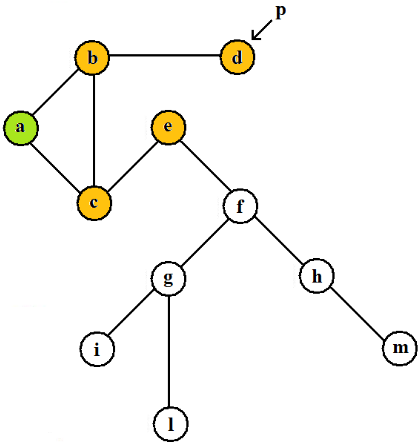
Con gap:



Iterazione 4:

R = {a, b, e}
OCC = {}
R = {a, b}

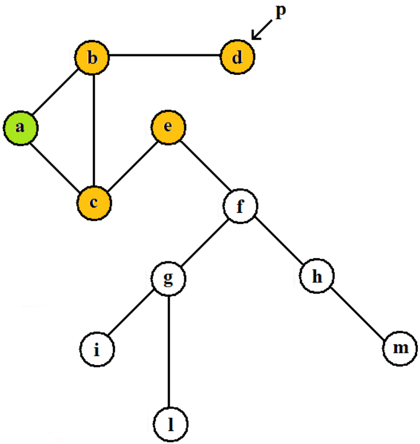
Con gap:



Iterazione 4:

R = {a, b, e}
OCC = {}
R = {a, b}
p = d

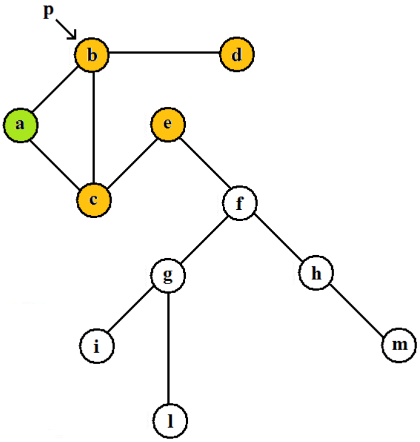
Con gap:



Iterazione 5:

R = {a, b, d}
OCC = {}
R = {a, b}

Con gap:

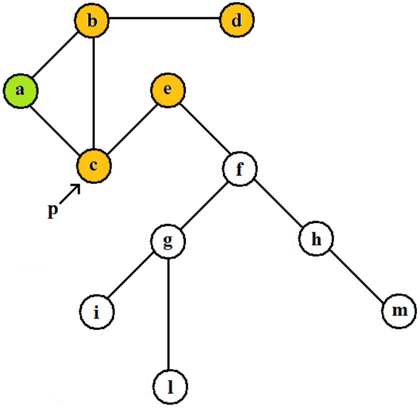


Iterazione 5:

R = {a, b, d}
OCC = {}
R = {a, b}

Backtracking
p = b
R = {a}
dtot = 0
Q = {a,b,c,e,d}

Con gap:

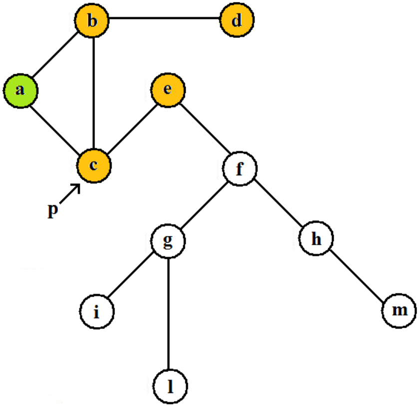


Iterazione 5:

R = {a, b, d}
OCC = {}
R = {a, b}

Backtracking
p = b
R = {a}
dtot = 0
Q = {a,b,c,e,d}
p = c

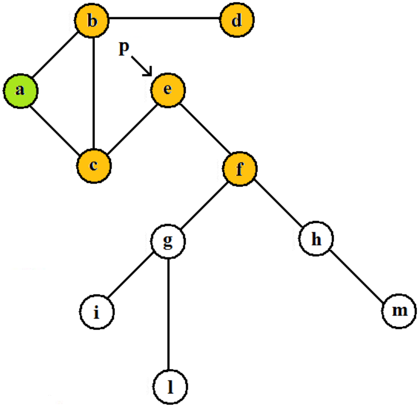
Con gap:



Iterazione 6:

R = {a, c}
S = {a}
d = 0
DIST[c]=0
V[c] = {f}

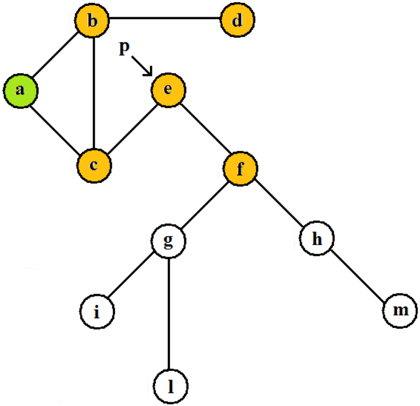
Con gap:



Iterazione 6:

R = {a, c}
S = {a}
d = 0
DIST[c]=0
V[c] = {f}
Q = {a,b,c,e,d,f}
p = e

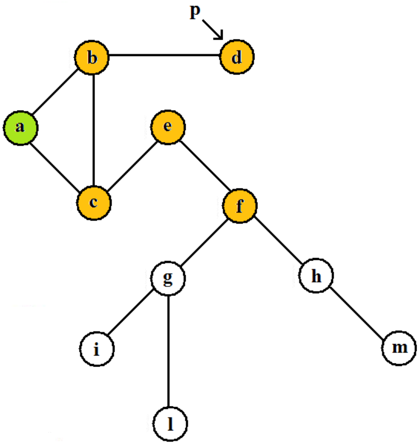
Con gap:



Iterazione 7:

R = {a, c, e}
OCC = {}
R = {a, c}

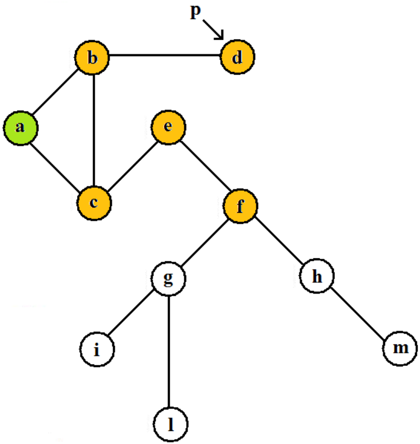
Con gap:



Iterazione 7:

R = {a, c, e}
OCC = {}
R = {a, c}
p = d

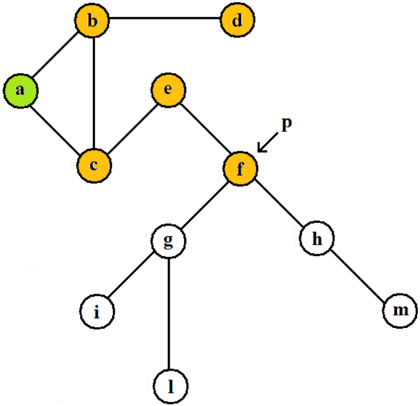
Con gap:



Iterazione 8:

R = {a, c, d}
OCC = {}
R = {a, c}

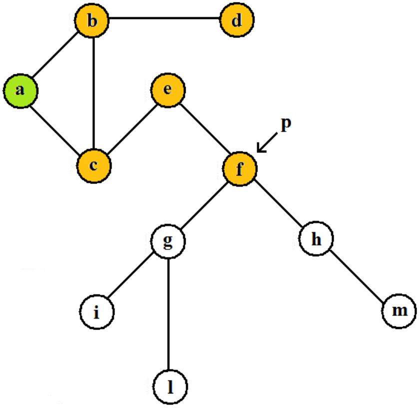
Con gap:



Iterazione 8:

R = {a, c, d}
OCC = {}
R = {a, c}
p = f

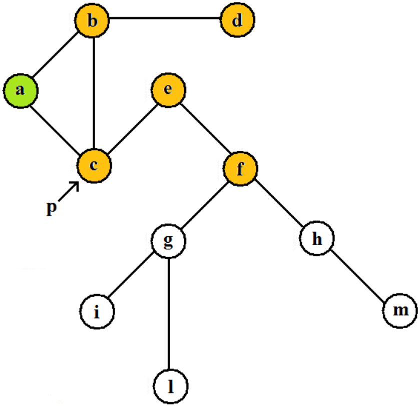
Con gap:



Iterazione 9:

R = {a, c, f}
OCC = {}
R = {a, c}

Con gap:

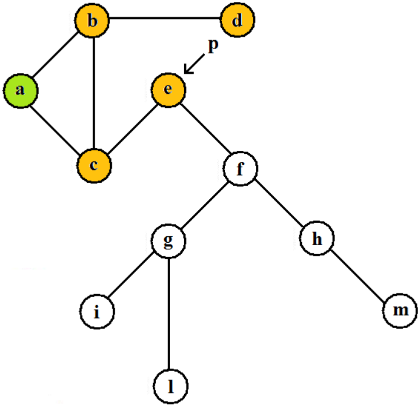


Iterazione 9:

R = {a, c, f}
OCC = {}
R = {a, c}

Backtracking
p = c
R = {a}
dtot = 0

Con gap:

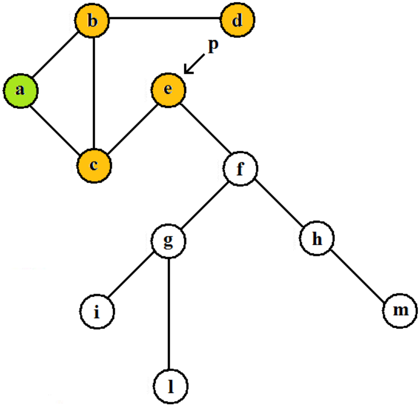


Iterazione 9:

R = {a, c, f}
OCC = {}
R = {a, c}

Backtracking
p = c
R = {a}
dtot = 0
Q = {a, b, c, e, d}
p = e

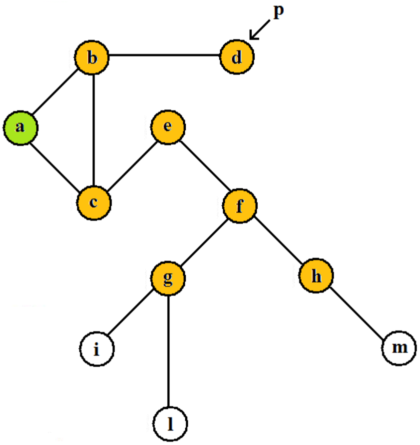
Con gap:



Iterazione 10:

R = {a, e}
S = {a}
d = 1
dtot = 1
DIST[e] = 1
V[e]=f,g,h

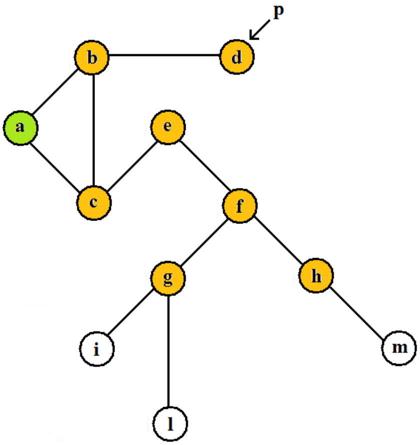
Con gap:



Iterazione 10:

R = {a, e}
S = {a}
d = 1
dtot = 1
DIST[e] = 1
V[e]=f,g,h
Q = {a,b,c,e,d,f,g,h}
p = d

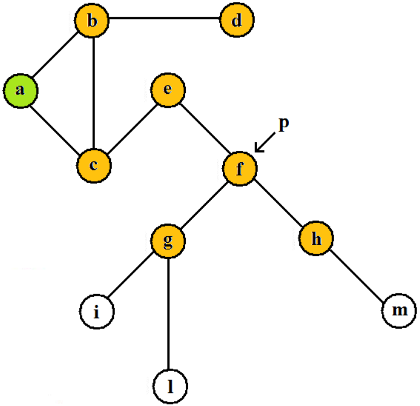
Con gap:



Iterazione 11:

R = {a, e, d}
OCC = {}
R = {a, e}

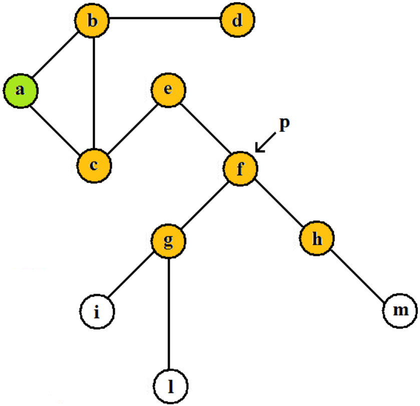
Con gap:



Iterazione 11:

R = {a, e, d}
OCC = {}
R = {a, e}
p = f

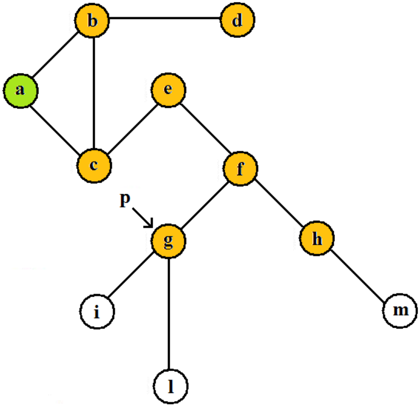
Con gap:



Iterazione 12:

R = {a, e, f}
OCC={{a, e, f}}
R = {a, e}

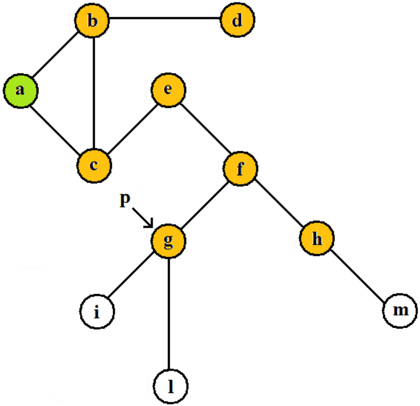
Con gap:



Iterazione 12:

R = {a, e, f}
OCC={{a, e, f}}
R = {a, e}
p = g

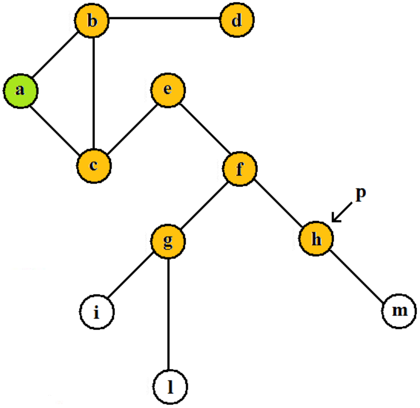
Con gap:



Iterazione 13:

R = {a, e, g}
OCC = {{a, e, f}}
R = {a, e}

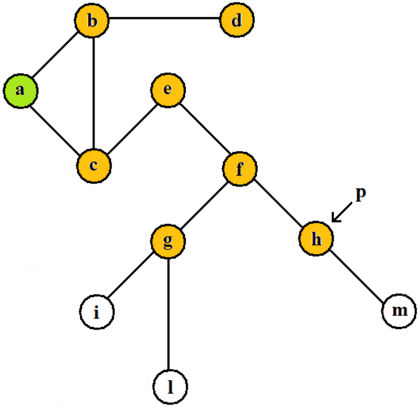
Con gap:



Iterazione 13:

R = {a, e, g}
OCC = {{a, e, f}}
R = {a, e}
p = h

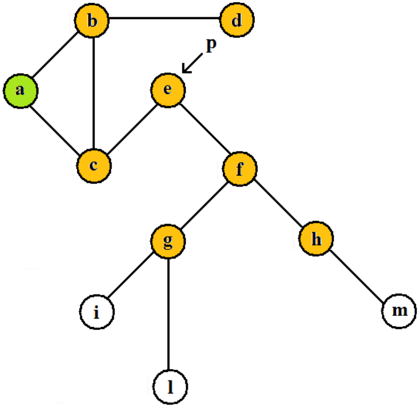
Con gap:



Iterazione 14:

R = {a, e, h}
OCC={{a, e, f},
{a, e, h}}
R = {a, e}

Con gap:

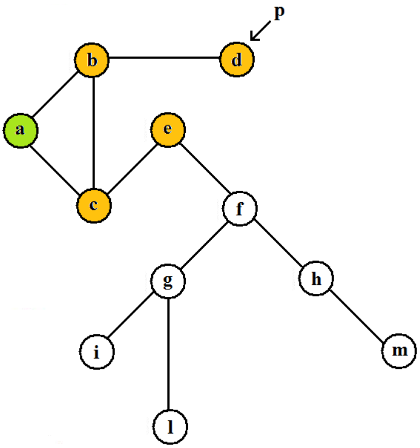


Iterazione 14:

R = {a, e, h}
OCC={{a, e, f},
{a, e, h}}
R = {a, e}

Backtracking
p = e
R = {a}
dtot = 0

Con gap:

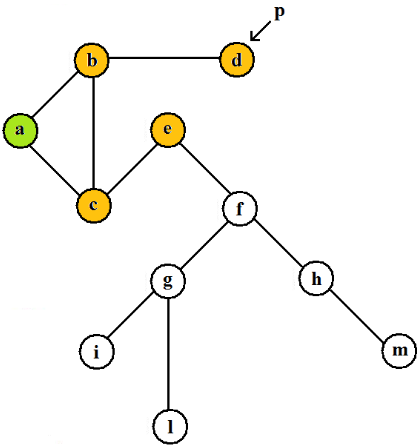


Iterazione 14:

R = {a, e, h}
OCC={{a ,e ,f},
{a ,e ,h}}
R = {a, e}

Backtracking
p = e
R = {a}
dtot = 0
Q = {a,b,c,e,d}
p = d

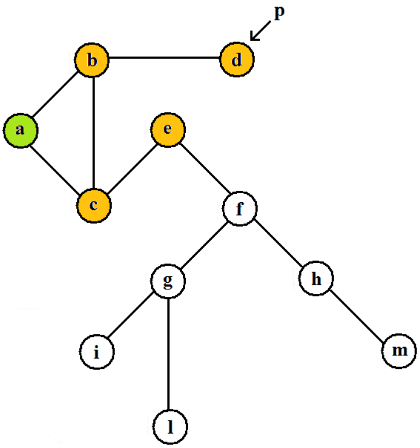
Con gap:



Iterazione 15:

R = {a,d}
S = {a}
d = 1
dtot = 1
V[d] = {}
Q = {a,b,c,e,d}

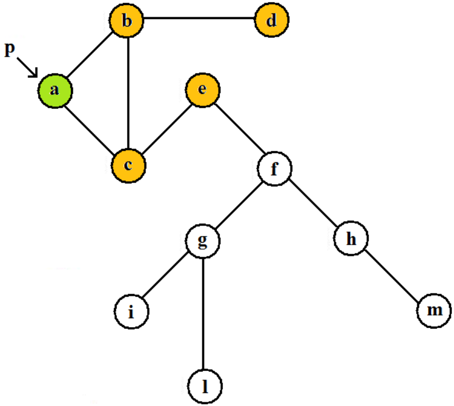
Con gap:



Iterazione 15:

R = {a,d}
S = {a}
d = 1
dtot = 1
V[d] = {}
Q = {a,b,c,e,d}
Backtracking
p = d
R = {a}
dtot = 0
Q = {a,b,c,e,d}

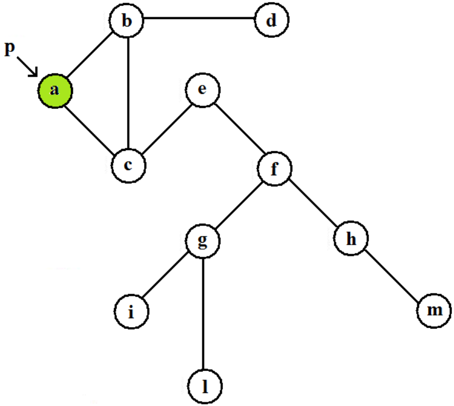
Con gap:



Iterazione 15:

R = {a,d}
S = {a}
d = 1
dtot = 1
V[p] = {}
Q = {a,b,c,e,d}
Backtracking
p = d
R = {a}
dtot = 0
Q = {a,b,c,e,d}
p = a
R={}
dtot=null

Con gap:



Iterazione 15:

R = {a,d}
S = {a}
d = 1
dtot = 1
V[p] = {}
Q = {a,b,c,e,d}
Backtracking
p = d
R = {a}
dtot = 0
Q = {a,b,c,e,d}
p = a
R={}
dtot=null
Q={a} STOP

Con gap: pseudocodice 1

```
procedura withGaps (  INPUT:  grafo G = <N,A> , motivo M, numero massimo di salti locali lb, numero
                              massimo di salti globali gb
                      OUTPUT: tutte le occorrenze di M in G   )

begin {
    OCC = {} //Occorrenze di M in G
    Q = {} //coda
    V[] = {} //insieme dei vicini di un nodo
    R = {} //insieme candidato come occorrenza
    p = null //puntatore a nodo in N
    S //sottografo formato dai nodi candidati a distanza da p al massimo pari a gb
    d //distanza minima tra p e ed il sottografo S
    dtot //il numero di salti totali compiuti per un determinato percorso
    DIST[ ] //numero di salti da un nodo totali tra un nodo ed il nodo origine
```

Con gap: pseudocodice 2

```
    foreach nodo n appartenente N  {
        Q = { }
        Q = Q unito {n}
        p = n
        do {
            R = R unito {p}
            if (|R| == |M|) {
                if (R è un match con M)
                    OCC unito R
                R = R tolto {p}
            }
            else {
                S = nodi presenti in R, escluso p, a distanza da p al massimo pari a gb
                d=distanza minima per arrivare da p a uno qualsiasi dei nodi di S oppure zero se p=n
                dtot=dtot+d
                DIST[p]=dtot
                V[p] = {nodi a distanza massima da p pari a min{lb,gb-dtot} non presenti in Q}
                Q = Q unito V[p]
            }
            while (p == ultimo elemento di Q && p!=n) {
                p = ultimo elemento in R
                R = R tolto {p}
                dtot=DIST[ultimo elemento in R]
                Q = Q tolto V[p]
            }
            p = successivo elemento in Q;
        } while (Q non contiene solo n)
    }
    return OCC
}
```

Risultati in tempo

Grafo utilizzato: 3184 nodi e 17642 archi (rete costruita a partire dal database di KEGG)

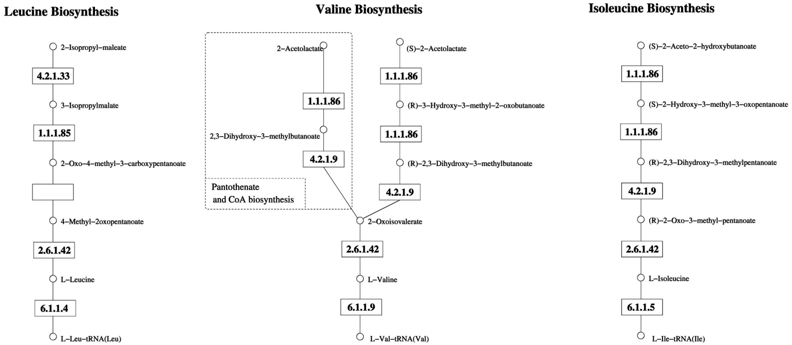
Per s=3:

- numero di colori: 171
- frequenza minima di un colore: 0,006
- frequenza più alta di un colore: 0,089

Per un motivo M con $|M| = 4$ ed $lb = gb = 0$, ricercare tutte le occorrenze ha impiegato 8ms (in media) su un Pentium IV (CPU da 1,70GHz) con 512Mb di memoria ram

Ipotesi evolutive

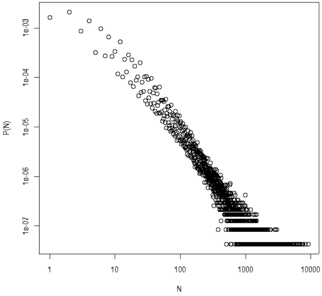
Tentativo di spiegare la storia evolutiva a partire dalla somiglianza dei pathway



$M = \{1.1.1.86, 1.1.1.86, 4.2.1.9, 2.6.1.42, 6.1.1.9\}$, $s = 3$, $lb = gb = 1$

Caratteristiche della rete: power law

- Distribuzione dei gradi dei vertici del grafo (pochi hub, molti nodi poco connessi)
- Distribuzione del numero di occorrenze dei motivi (con $|M| = 4$ e $s = 3$, $|M| = 4$ e $s = 2$, $|M| = 3$ e $s = 3$ e $|M| = 3$ e $s = 2$)



Inoltre il 95% dei motivi di lunghezza 3 (il 98% nel caso di motivi lunghi 4) non occorrono mai

Caratteristiche della rete: occorrenze interpathway

Occorrenze interpathway

Occorrenze di un motivo che superano i confini dei pathway così come sono stabiliti ora

$|M| = 3$

$s = 3$

74% delle occorrenze sono interpathway

$|M| = 4$

$s = 3$

92% delle occorrenze sono interpathway

Bibliografia

- Vincent Lacroix, Cristina G. Fernandes, Marie-France Sagot - *Motif search in graph: application to metabolic network*
- Vincent Lacroix, Ludovic Cottret, Patricia Thébault, Marie-France Sagot - *An introduction to metabolic networks and their structural analysis*
- Giuseppe, Domenico Arrabito - *Le reti metaboliche*
- http://www.genome.jp/kegg/
- http://www.wikipedia.org/