ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

Εισαγωγή στη Βιοπληροφορική

Ενότητα 2: Τεχνικές διαχείρισης – Αλγόριθμοι εύρεσης προτύπων

Μακρής Χρήστος, Τσακαλίδης Αθανάσιος, Περδικούρη Αικατερίνη

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Σκοποί ενότητας

- Η εκμάθηση των τεχνικών διαχείρισης και ανάλυσης συμβολοσειρών βιολογικών δεδομένων
- Η εκμάθηση αλγορίθμων ακριβούς εύρεσης προτύπου

Περιεχόμενα ενότητας

- Βασικοί ορισμοί
- Το πρόβλημα της ακριβούς εύρεσης προτύπου
- Το πρόβλημα της προσεγγιστικής εύρεσης προτύπου
- Αλγόριθμος Boyer-Moore
- Αλγόριθμος Knuth-Morris-Pratt
- Αλγόριθμος Shift-Or
- Αυτόματο Aho-Corasick
- Εφαρμογές

Βασική Βιβλιογραφική Πηγή στην οποία βασίζονται οι διαφάνειες

- Dan Gusfield , Algorithms on Strings, Trees and Sequences,, Cambridge University Press, 10th edition 2007

Τεχνικές διαχείρισης – Αλγόριθμοι εύρεσης προτύπων

Διάγραμμα Ύλης

Α' Μέρος

- Κεφάλαιο 1ο: Εισαγωγή στη χρήση αλγορίθμων για αποτελεσματική διαχείριση και αποθήκευση συμβολοσειρών (strings)και ακολουθιών βιολογικών δεδομένων.

- Κεφάλαιο 2ο: Αλγόριθμοι ακριβούς ταιριάσματος προτύπου (Boyer-Moore,Knuth-Morris-Pratt,Shift-Or, Πολλαπλών Προτύπων).

- Κεφάλαιο 3ο:Εισαγωγή στο δέντρο επιθεμάτων (suffix tree) και στις εφαρμογές του.

- Κεφάλαιο 4ο:Αλγόριθμοι προσεγγιστικού ταιριάσματος προτύπου και στοίχισης συμβολοσειρών/ ακολουθιών (Sequence Alignment).

- Κεφάλαιο 5ο: Αλγόριθμοι εύρεσης σε Βάσεις Δεδομένων ακολουθιών (FASTA, BLAST, PROSITE)

Β' Μέρος

Κεφάλαιο 1ο: Η Θεωρητική Βάση του Μοριακού Σχεδιασμού

Κεφάλαιο 2ο: Μοριακά Μοντέλα και Βιοχημική Πληροφορία

Κεφάλαιο 3ο: Η Βασιζόμενη στη Δομή Σχεδίαση Φαρμάκων

Κεφάλαιο 4ο: Ανοικτά Προβλήματα

Γ' Μέρος

Τεχνικές κατηγοριοποίησης βιολογικών δεδομένων (Clustering Techniques) με σκοπό την πρόβλεψη της συμπεριφοράς βιολογικών μορίων.

Τεχνικές Ανάλυσης και Σύγκρισης Ακολουθιών Βιολογικών Δεδομένων

- Παραδείγματα Βάσεων Δεδομένων Βιολογικών Ακολουθιών
- Βασικοί Ορισμοί
- Το πρόβλημα του ακριβούς ταιριάσματος προτύπου
  - Απλοϊκή Μέθοδος
  - Αλγόριθμος Boyer-Moore
  - Αλγόριθμος Knuth-Morris-Pratt
  - Αλγόριθμος Shift-Or/Shift And
  - Το Αυτόματο Aho-Corasick
- Εφαρμογές σε Προβλήματα Μοριακής Βιολογίας

Βιολογικές Βάσεις Δεδομένων
(πηγή: Wikipedia)

- Γενικευμένες (Generalised) ή Αρχειακές (Archival) βιολογικές βάσεις δεδομένων. Διακρίνονται σε:

  - Πρωτογενείς βάσεις δεδομένων ακολουθιών (Primary Sequence Databases). Περιέχουν νουκλεοτιδικές και αμινοξικές ακολουθίες από γονιδιώματα οργανισμών που είτε έχουν αποκρυπτογραφηθεί πλήρως είτε

  - βάσεις δεδομένων που περιέχουν τρισδιάστατες δομές νουκλεϊνικών οξέων και πρωτεϊνών

  (GENBANK, EMBL-Bank, DDJB, Swiss-Prot, PIR-PSD)

- Δευτερεύουσες (Secondary) βιολογικές βάσεις δεδομένων που προκύπτουν από ανάλυση των δεδομένων που είναι αποθηκευμένα στις αρχειακές βιολογικές βάσεις δεδομένων και διακρίνονται σε:

✓ Δευτερεύουσες ΒΔ ακολουθιών DNA και πρωτεϊνών που προκύπτουν από τις βασικές ΒΔ ακολουθιών και περιλαμβάνουν

(α) ΒΔ ακολουθιών στις οποίες έχουν απομακρυνθεί οι ακολουθίες που έχουν αποθηκευτεί περισσότερες από μία φορές

(β) ΒΔ που καταγράφουν μεταλλαγές ή παραλλαγές στις ακολουθίες DNA και πρωτεινών

(γ) Γονιδιωματικές ΒΔ που είτε ομαδοποιούν συγγενή ή όχι πλήρως αποκρυπτογραφημένα γονιδιώματα είτε ασχολούνται με γονιδιώματα οργανισμών μοντέλων

- ✓ ΒΔ που ασχολούνται με τις ιεραρχήσεις ή/και συσχετίσεις μεταξύ βιομορίων όπως οικογένειες πρωτεϊνών, κοινές δομές πρωτεϊνών κοινά μοτίβα ακολουθιών DNA και πρωτεϊνών.

❑Εξειδικευμένες Β.Δ., κατηγορία στην οποία ανήκουν:

- ✓ Β.Δ. μικροσυστοιχιών που περιλαμβάνουν πληροφορίες για την έκφραση γονιδίων και πρωτεϊνών

- ✓ Β.Δ. Μεταβολικών μονοπατιών που περιέχουν πληροφορίες για τις χημικές αντιδράσεις που πραγματοποιούνται στο κύτταρο

❑Βιβλιογραφικές βιολογικές βάσεις δεδομένων

❑Βιολογικές βάσεις δεδομένων ιστοσελίδων που περιλαμβάνουν:

- ✓ Β.Δ. που περιλαμβάνουν ως εγγραφές βιολογικές βάσεις

- ✓ Συνδέσμους μεταξύ βιολογικών βάσεων δεδομένων.

Παραδείγματα Βάσεων Βιολογικών Δεδομένων

- GenBank: NCBI (http://www.ncbi.nlm.nih.gov)
- PIR: Protein Information Resource (http://pir.georgetown.edu)
- Swiss-Prot + TrEMBL: Swiss-Prot.htm (http://tw.expasy.org/sprot/)
- PROSITE: Prosite (http://tw.expasy.org/prosite/)
- PDB-Protein Data Bank: PDB (http://www.rcsb.org/pdb/
- SCOP: (http://scop.berkeley.edu/) Structural Classification of Proteins
- PRINTS: (http://umber.sbs.man.ac.uk/dbbrowser/PRINTS)

Βασικοί Ορισμοί (1)

- *Συμβολοσειρά-string*: x=x[1]x[2]…..x[n], x[i]$\in\Sigma$ & |x|=n
  x= acgttaaaca, |x|=10 & Σ={A,C,G,T}

(Αδενίνη, Θυμίνη, Κυτοσίνη, Γουανίνη)

- Σ^+*:* το σύνολο των συμβολοσειρών που ορίζονται στο αλφάβητο Σ
  ? Πόσες συμβολοσειρές μήκους «λ» ορίζονται στο Σ={a,c,g,t}

- *Κενή συμβολοσειρά:* ε
- *Υπο-συμβολοσειρά-substring w*: *x=uwv*
- *Πρόθεμα –Prefix w: x=wu*
- *Επίθεμα-Suffix w: x=uw*

Βασικοί Ορισμοί (2)

- Border συμβολοσειράς x: συμβολοσειρά w που είναι πρόθεμα και επίθεμα του *x*.
- $x^k = x \ldots x$, κ-οστή δύναμη του x

  $\underbrace{\qquad}$

  κ φορές
- $y = x^k$, $k>1$ → το y περιοδικό με περίοδο x
- Περίοδος y= η μικρότερη τέτοια συμβολοσειρά
- Primitive (πρωταρχική) συμβολοσειρά
- Κάλυμμα (Cover) συμβολοσειράς
- Φύτρο (Seed) συμβολοσειράς

Προβλήματα Ταιριάσματος Προτύπου (1)

- Ακριβές Ταίριασμα: ενδιαφερόμαστε να εντοπίσουμε όλες τις εμφανίσεις ενός δοσμένου προτύπου (μοτίβου) *P* (“δομημένου” ή “μη-δομημένου”) σε μια συμβολοσειρά (βιολογική αλληλουχία) *T* .
- Προσεγγιστικό Ταίριασμα: *Για ένα κείμενο T , ένα μοτίβο P , μια παράμετρο k και μια συνάρτηση ομοιότητας d( ) , εντόπισε τις θέσεις i, j στο κείμενο , έτσι ώστε*

  $d(P, T_{i..j}) >= k$.

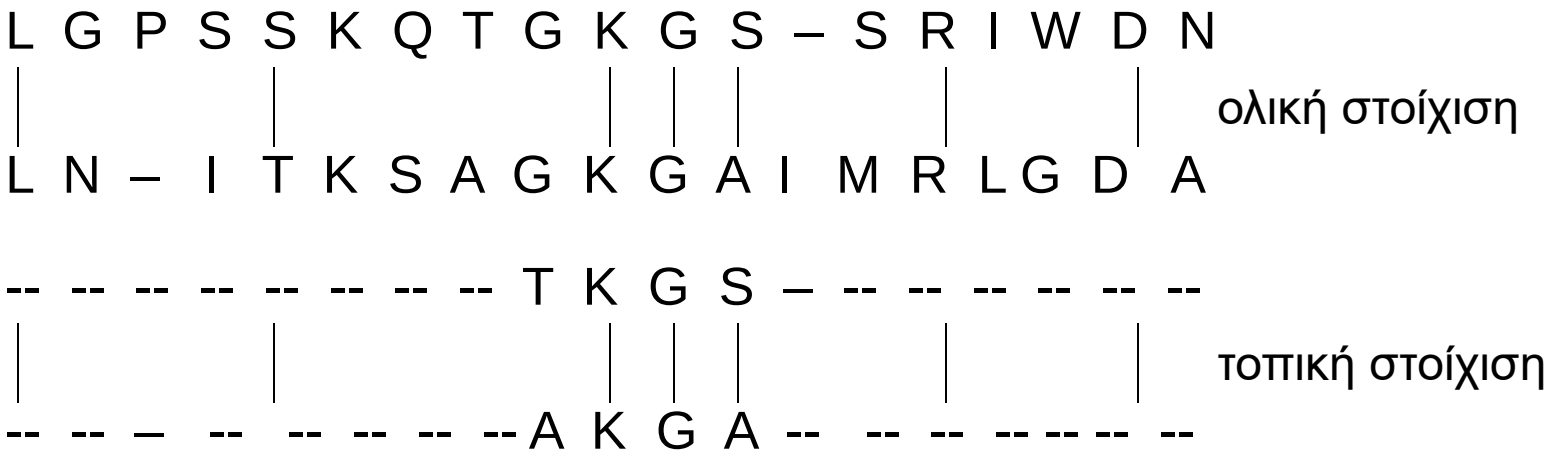
Προβλήματα Ταιριάσματος Προτύπου (2)

- Η διαδικασία σύγκρισης της ομοιότητας δυο ακολουθιών στηρίζεται σε πίνακες που βαθμολογούν τις ομοιότητες (matches) και διαφορές (mismatches) μεταξύ διαδοχικών συμβόλων. Τέτοιου τύπου πίνακες είναι οι: Dayhoff Mutation Data Matrix, BLOSUM κτλ.

- Επίσης η σύγκριση ακολουθιών μπορεί να κατηγοριοποιηθεί σε: α) **τοπική ευθυγράμμιση -local alignment** και β) **ολική ευθυγράμμιση - global alignment**. Στην τοπική ευθυγράμμιση αναζητούμε περιοχές τοπικής ομοιότητας. Γνωστοί τέτοιοι αλγόριθμοι είναι των Smith-Waterman (τοπικοί), Needleman & Wunsch (ολικοί). Και στις δυο περιπτώσεις υπάρχουν παραπάνω από μια δυνατές ευθυγραμμίσεις. Η βέλτιστη λύση πρέπει να ελαχιστοποιεί τις διαφορές ανάμεσα στις δυο ακολουθίες ή διαφορετικά να μεγιστοποιεί τη συνάρτηση ομοιότητας.

Στοίχιση Ακολουθιών

- Συνέκρινε δύο ή περισσότερες ακολουθίες ελέγχοντας για μία ακολουθία ατομικών χαρακτήρων που είναι με την ίδια σειρά στις ακολουθίες.
- Ανακάλυψε λειτουργική, δομική και εξελεκτική πληροφορία.



Εύρεση Επαναλήψεων σε Βιολογικές Ακολουθίες

- Οι επαναλήψεις σε βιολογικές ακολουθίες κατηγοριοποιούνται στις εξής 3 βασικές κατηγορίες:
  - επαναλήψεις περιορισμένου μήκους που εμφανίζονται σε τοπικό επίπεδο, και των οποίων η λειτουργία είναι γνωστή,
  - επαναλήψεις περιορισμένου μήκους που εμφανίζονται σε όλο το μήκος της ακολουθίας, και των οποίων η λειτουργία δεν είναι απόλυτα γνωστή,
  - δομημένες επαναλήψεις μεγάλου μήκους των οποίων η λειτουργία δεν έχει προσδιοριστεί.

Παραδείγματα Επαναλήψεων

- 1η κατηγορία:
  - τα **συμπληρωματικά παλίνδρομα** σε ακολουθίες DNA & RNA, που ρυθμίζουν τη μετεγγραφή του DNA,
  - τα **εμφωλευμένα συμπληρωματικά παλίνδρομα** σε ακολουθίες RNA
- 2η κατηγορία:
  - **συνεχόμενες επαναλήψεις- tandem repeats**,
  - **δορυφορικά τμήματα DNA**- **satellite DNA**, (**micro** & **mini satellite DNA**)
- 3η κατηγορία:
  - **SINE-Short Interspersed Nuclear Sequences (π.χ.:** *Alu family)*
  - **LINE-Long Interspersed Nuclear Sequences**.

Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Πρότυπα

- **Μοτίβα DNA**

  TRANSFAC, JASPAR, SCPD, DBTBS, RegulonDB

- **Μοτίβα πρωτεϊνών**

  PROSITE, Pfam, ProDom, BLOCKS, TIGRFAM, Interpro

Ακριβές Ταίριασμα (εφαρμογές)

- Επεξεργαστές κειμένου
- Utilities (grep στο Unix)
- Textual Information Retrieval (Medline, Lexis, Nexis)
- Internet News Readers
- On-line dictionaries και θησαυρούς
- Molecular Biology Databases

Ακριβής Εύρεση Προτύπου
The Exact Pattern Matching Problem

Ορισμός: *«έστω μια ακολουθία χαρακτήρων Τ. Αναζητούμε τις θέσεις εμφάνισης* του προτύπου/ λέξης *P* μέσα *στην ακολουθία».*

P= acgttaaaca

T= tcgacgttaaacatttaaatttacgttaaacaggggaattcgacgttaaaca

1η εμφάνιση **2η εμφάνιση** **3η εμφάνιση**

Η απλοϊκή μέθοδος επίλυσης – Naive Method

1ο βήμα: Στοιχίζουμε την ακολουθία και το πρότυπο & συγκρίνουμε τους χαρακτήρες

g	c	a	t	g	c	a	g	a	g	a	g	t	a	t	a	c	a	g	t	a	c	g
1	2	3	4	5	6	7	8															
g	c	a	g	a	g	a	g															

2ο βήμα: Στο πρώτο mismatch – 4η θέση μετατοπίζουμε το πρότυπο κατά 1 θέση

g	c	a	t	g	c	a	g	a	g	a	g	t	a	t	a	c	a	g	t	a	c	g
	1	2	3	4	5	6	7	8														
	g	c	a	g	a	g	a	g														

Η απλοϊκή μέθοδος επίλυσης – Naive Method

3ο βήμα: Σε κάθε mismatch μετατοπίζουμε το πρότυπο κατά 1 θέση

g	c	a	t	g	c	a	g	a	g	a	g	t	a	t	a	c	a	g	t	a	c	g
		1	2	3	4	5	6	7	8													
		g	c	a	g	a	g	a	g													

4ο βήμα:

g	c	a	t	g	c	a	g	a	g	a	g	t	a	t	a	c	a	g	t	a	c	g
			1	2	3	4	5	6	7	8												
			g	c	a	g	a	g	a	g												

Η απλοϊκή μέθοδος επίλυσης – Naive Method

5ο βήμα: 1η εύρεση του προτύπου $L_{\{X\}}=\{5,\ldots\}$

g	c	a	t	g	c	a	g	a	g	a	g	t	a	t	a	c	a	g	t	a	c	g
				1	2	3	4	5	6	7	8											
				g	c	a	g	a	g	a	g											

6ο βήμα:

g	c	a	t	g	c	a	g	a	g	a	g	t	a	t	a	c	a	g	t	a	c	g
					1	2	3	4	5	6	7	8										
					g	c	a	g	a	g	a	g										

Κώδικας Απλοϊκής Μεθόδου

```
void Naïve-Method (char *P, int m, char *T, int n)
{
    int i,j;
    for (j=0; j<=n-m; ++j)   {
        for (i=0; i<m && P[i]==T[i+j]; ++i);
        if (i>=m) output(j);
                                }
}
```

Ανάλυση της απλοϊκής μεθόδου σε χρόνο

- Πολυπλοκότητα μεθόδου: O(n*m), όπου |T|=n & |P|=m

1. Πόσες μετατοπίσεις θα χρειαστεί να γίνουν?
   |T|- |P|+1= n –m +1

2. Πόσες συγκρίσεις πραγματοποιούνται το πολύ κάθε φορά?
   |P|=m

3. Συνολικός χρόνος επεξεργασίας: (n –m +1)*m
4. Πώς μπορώ να βελτιώσω το χρόνο?

Ας θυμηθούμε τα μεγέθη των δεδομένων

Πηγή Δεδομένων	**Μέγεθος Δεδομένων**	**Πολυπλοκότητα για το Exact Pattern Matching Problem**
Ακολουθίες DNA	11.5 εκατ. Ακολουθίες (12.5 δις. Βάσεις)	- |n|= 12.5 δις. Βάσεις - |m|= 100 - n*m= 12.500.000.000*100= ..**απαγορευτικό**
Γονιδιώματα	300 πλήρη γονιδιώματα (1.6 εκατ-3 δις βάσεις το καθένα)	- |n|= 1.6 εκ. Βάσεις - |m|= 100 - n*m= 1.600.000*100=.....**απαγορευτικό**

Θα παρουσιάσουμε 3 αποδοτικούς αλγορίθμους γραμμικού χρόνου ως προς το μέγεθος της ακολουθίας εισόδου O(|T|)

Βασική Προεπεξεργασία (1)
(D. Gusfield)

- $Z_i(S)$= το μήκος της μεγαλύτερης υποσυμβολοσειράς του S, που αρχίζει στο i και ταιριάζει πρόθεμα του S.
- Z-box at i= το σύνολο χαρακτήρων που αρχίζουν από i και τελειώνουν στη θέση $i+Z_i(S)-1$.
- Για κάθε i, r_i συμβολίζει το δεξιότερο άκρο των Z-boxes που ξεκινά από ή πριν τη θέση i. Διαφορετικά, r_i είναι η μεγαλύτερη τιμή του $j+Z_j-1$ για κάθε $2<j<=i$. Το αριστερό άκρο (j) το συμβολίζουμε ως l_i.

Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Βασική Προεπεξεργασία (2)

Δοθέντων Z_i για i<=k-1 και r, l για Z_{k-1}:

1. If k>r, find Z_k explicitly. If Z_k>0, r=k+Z_k-1, l=k.
2. If k<=r, S[k..r] matches S[k'..Z_l] and the substring at k matches a prefix of S of length >=min($Z_{k'}$,r-k+1) (k'=k-l+1)

   2.a. If $Z_{k'}$ < |S[k..r]| then Z_k=$Z_{k'}$, r,l remain unchanged

   2.b. Compare the characters starting at r+1 of S with the characters starting at |S[k..r]| until a mismatch. Say the mismatch occurs at q>=r+1. Then Z_k is q-k, r is set to q-1 and l is set to k.

Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Βασική Προεπεξεργασία (3)

- Εφάρμοσε τον αλγόριθμο στην ακολουθία P$T
- Κάθε τιμή Z_i=m, i>m σηματοδοτεί match στη θέση i-m-1.
- Η μέθοδος μπορεί να υλοποιηθεί έτσι ώστε να απαιτεί επιπλέον (του χώρου αποθήκευσης P, T), O(m) χώρο.
- Είναι μέθοδος που οι πολυπλοκότητές της είναι ανεξάρτητες από το μέγεθος του αλφαβήτου (ίδια ιδιότητα έχει ο αλγόριθμος Boyer Moore, Knuth Morris Pratt, όχι όμως ο Shift Or και ο αλγόριθμος Aho-Corasick)

Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Ο αλγόριθμος Boyer-Moore

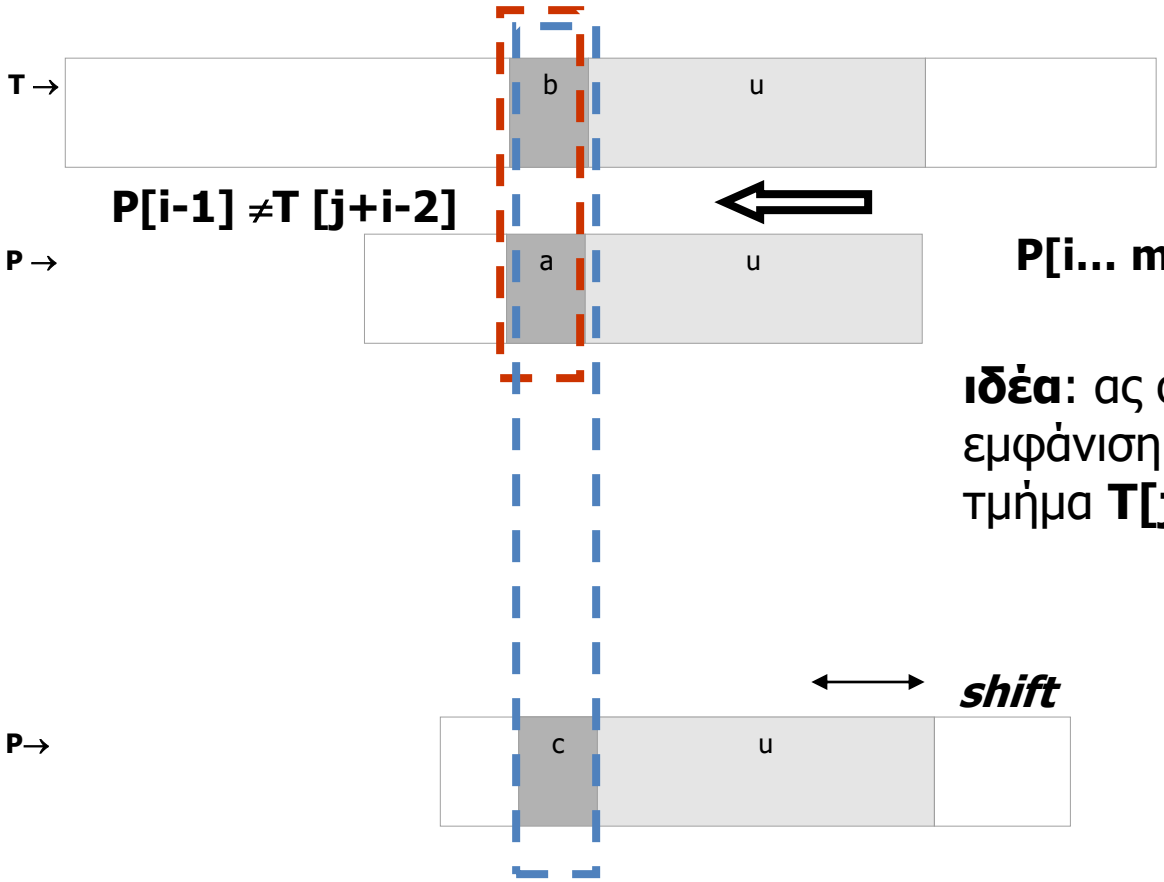
1η ιδέα: Στοιχίζουμε την ακολουθία και το πρότυπο & συγκρίνουμε τους χαρακτήρες από δεξιά προς τα αριστερά

g	c	a	t	c	g	c	a	g	a	g	a	g	t	a	t	a	c	a	g	t	a	c	g
1	2	3	4	5	6	7	8																
g	c	a	g	a	g	a	g																

2η ιδέα: Σε κάθε mismatch μετατοπίζω το πρότυπο περισσότερες από 1 θέσεις βάσει 2 κανόνων

g	c	a	t	c	g	c	a	g	a	g	a	g	t	a	t	a	c	a	g	t	a	c	g
	1	2	3	4	5	6	7	8															
	g	c	a	g	a	g	a	g															

α' κανόνας: *"good suffix shift"*

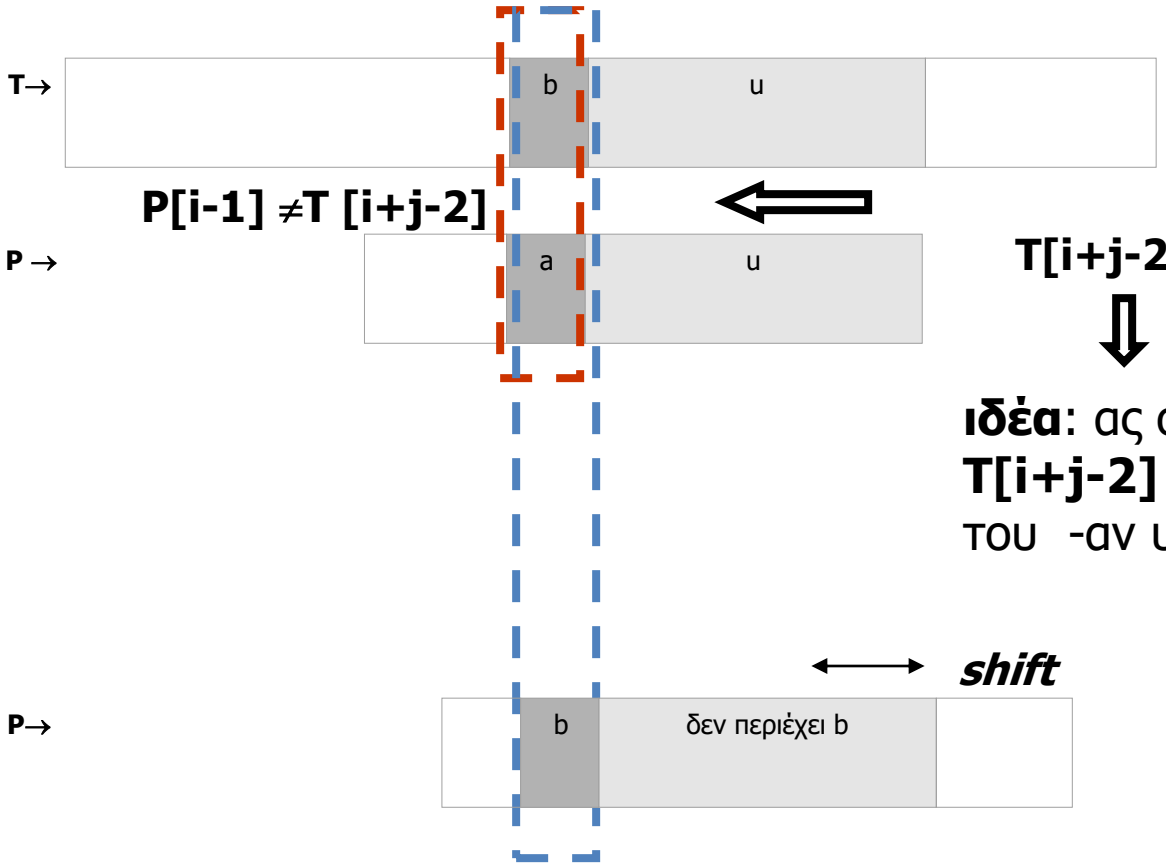


P[i… m] = T[j+i-1…j+m]

⇩

ιδέα: ας στοιχίσουμε την επόμενη εμφάνιση του **u** με το αντίστοιχο τμήμα **T[j+1-1, …, j+m]**

β' κανόνας: *"bad character shift"*



T[i+j-2] = b

⇩

ιδέα: ας στοιχίσουμε το χαρακτήρα **T[i+j-2] =b** με τη δεξιότερη εμφάνισή του -αν υπάρχει- στο πρότυπο **P**

Υλοποίηση “bad character shift”

- (Simple Shift) Χρησιμοποίησε ένα πίνακα μεγέθους m x |Σ| και σάρωσε το πρότυπο από αριστερά προς τα δεξιά.

- (Extended Shift) Σάρωσε το πρότυπο από δεξιά προς τα αριστερά και για κάθε χαρακτήρα δημιούργησε μία λίστα από εμφανίσεις. Σάρωσε την κατάλληλη λίστα όταν εντοπιστεί.

Υλοποίηση “goof suffix shift” (1)

- Έστω L(i) η μεγαλύτερη θέση στο P, έτσι ώστε η P[i..m] να ταιριάζει ένα επίθεμα του P[1..L[i]] με τον επιπλέον περιορισμό ότι ο χαρακτήρας που προηγείται του επιθέματος θα πρέπει να είναι διαφορετικός του P[i-1].
- Ορίζουμε ως l’(i) το μήκος του μεγαλύτερου επιθέματος του P[i..m] που είναι επίσης πρόθεμα του P.

Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Υλοποίηση “goof suffix shift” (2)

- Έστω $N_j(P)$ το μήκος του μεγαλύτερου επιθέματος της συμβολοσειράς P[1..j] που είναι επίθεμα του P .

  $N_j(P)= Z_{m-j+1}(P^r)$

  Από τον ορισμό αυτό προκύπτεί ότι L(i) είναι η μεγαλύτερη τιμή j<m έτσι ώστε

  $$N_j(P)=|P[i..m]|$$

  Τέλος l’[i] είναι το μεγαλύτερο j<=|P[i..m]|, έτσι ώστε $N_j(P)=j$

Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Υλοποίηση “goof suffix shift” (3)

for i=1 to m do L’(i)=0;

for j=1 to m-1 do

  begin

  i=m-N_j(P)+1;

  L(i):= j;

  end

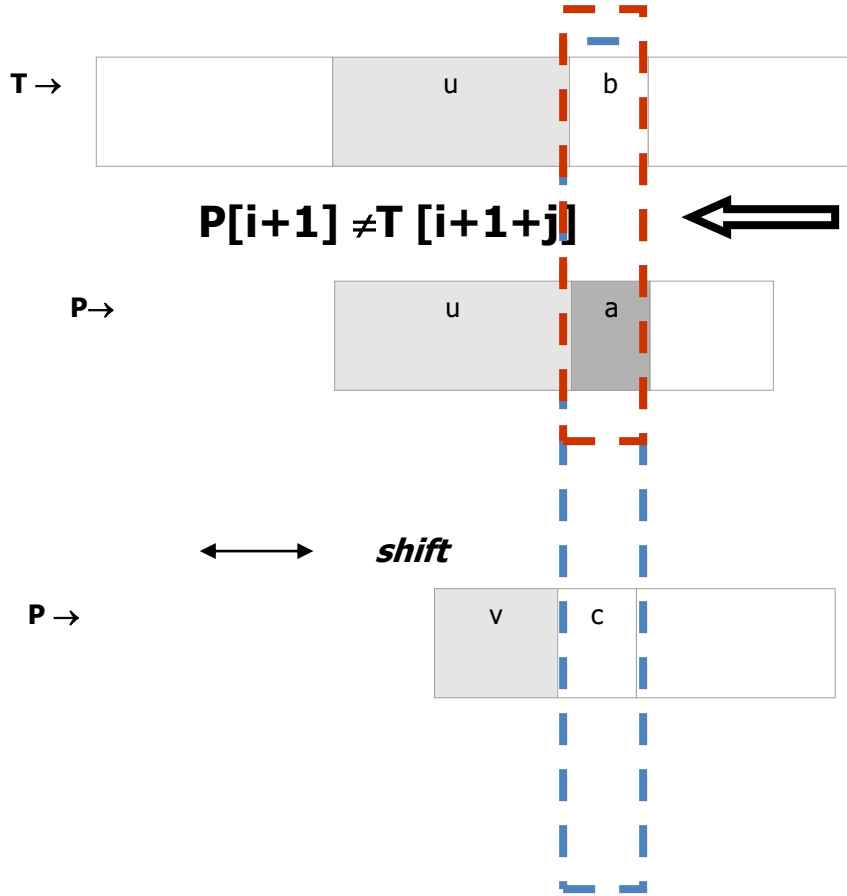
Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Ανάλυση του αλγορίθμου Boyer-Moore σε χρόνο

- Πολυπλοκότητα μεθόδου: O(n*m), όπου |T|=n & |P|=m

1. Οι απαιτούμενοι πίνακες υπολογίζονται σε O(m+σ) χρόνο.
2. Σε πραγματικές εφαρμογές απαιτούνται “3n” συγκρίσεις.
3. Για μεγάλο αλφάβητο |Σ|= ($\approx|\text{pattern}|$), απαιτούνται O(n/m) συγκρίσεις.
4. Χωρίς match ο χρόνος είναι O(n).
5. Υπάρχουν παραλλαγές με worst-case O(n+m) χρόνο

Ο αλγόριθμος Knuth-Morris-Pratt



P[1… i] = T [j…i+j-1] = u

⇩

ιδέα: ας στοιχίσουμε το μέγιστο πρόθεμα **prefix(P)=v** με το αντίστοιχο τμήμα από το επίθεμα **u** της ακολουθίας

Ας δούμε τον αλγόριθμο στην πράξη

1ο βήμα: Στοιχίζουμε την ακολουθία και το πρότυπο & συγκρίνουμε τους χαρακτήρες από αριστερά προς τα δεξιά

x	y	a	b	c	x	a	b	c	x	a	d	c	d	q	f	e	g	a	g	t	a	c	g
		1	2	3	4	5	6	7	8	9													
		a	b	c	x	a	b	c	d	e													

2ο βήμα: Μετατοπίζω το πρότυπο κατά 4 θέσεις

x	y	a	b	c	x	a	b	c	x	a	d	c	d	q	f	e	g	a	g	t	a	c	g
						1	2	3	4	5	6	7	8	9									
						a	b	c	x	a	b	c	d	e									

Υλοποίηση ΚΜΡ (1)

- Ορίζω ως $sh_i(P)$ το μήκος του μεγαλύτερου επιθέματος του P[1..i] που ταιριάζει ένα πρόθεμα του P, με την επιπλέον συνθήκη ότι οι χαρακτήρες P[i+1] και $P[sh_i(P)+1]$ είναι διαφορετικοί.
- Η θέση j>1 αντιστοιχίζεται στο i εάν $i=j+Z_j(P)-1$.
- Για κάθε i>1 $sh_i(P)=i-j+1$;οπου j είναι η μικρότερη θέση που αντιστοιχίζεται στο i.

Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Υλοποίηση KMP (2)

For i=1 to m to $sh_i(P)=0$;

For j=m downto 2 do

begin

$i=j+Z_j(P)-1$;

$sh_i(P)=Z_j(P)$;

end

Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Real Time KMP

- Ορίζω ως $sh_i(P,x)$ το μήκος του μεγαλύτερου επιθέματος του P[1..i] που ταιριάζει ένα πρόθεμα του P, με την επιπλέον συνθήκη ότι ο χαρακτήρας $P[sh_i(P,x)+1]$ είναι ο x.
- Όπως πριν:

For i=1 to m to $sh_i(P)=0$;

For j=m downto 2 do

begin

$i=j+Z_j(P)-1$;

$x=P[Z_j(P)+1]$

$sh_i(P,x)=Z_j(P)$;

end

Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Ανάλυση του αλγορίθμου Knuth-Morris-Pratt σε χρόνο

- Πολυπλοκότητα μεθόδου: O(n+m), όπου |T|=n & |P|=m

1. Σε χρόνο O(m) υπολογίζω, την «περίοδο» του προτύπου με την οποία μετατοπίζω το πρότυπο

Ο αλγόριθμος Shift-Or

- Ο αλγόριθμος χρησιμοποιεί αριθμητικές τεχνικές:

1. Έστω για κάθε char c $\in$ Σ, το διάνυσμα S_c μεγέθους m=|P|, που αποθηκεύει τις εμφανίσεις του c μέσα στο πρότυπο (με 0 p προσδιορίζεται η εμφάνιση),

2. Ο πίνακας R[mxn]: bit-array όπου R[i,j] είναι 0 αν και μόνο αν οι πρώτοι i χαρακτήρες του P ταιριάζουν με τους i χαρακτήρες που τελειώνουν στο j-οστό χαρακτήρα του T.

3. $R_{j+1} = \text{Shift}\ (R_j\ )\ \text{OR}\ S_{T[j+1]}$

Πηγή Dan Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 2010.

Ας δούμε τον αλγόριθμο στην πράξη

1ο βήμα: Υπολογίζω τα διανύσματα S_C για το πρότυπο p=gcagagag

	Sa	S_c	S_g	S_t
g	1	1	0	1
c	1	0	1	1
a	0	1	1	1
g	1	1	0	1
a	0	1	1	1
g	1	1	0	1
a	0	1	1	1
g	1	1	0	1

...Ας δούμε τον αλγόριθμο στην πράξη

2ο βήμα: Υπολογίζω τις τιμές του πίνακα R, σύμφωνα με τον τύπο:

$R_{j+1} = \text{Shift}(R_j)\ \text{Or}\ S_{T[j+1]}$

		0	1	2	3	4	5	6	7	8	9	10	11	12
		g	c	a	t	c	g	c	a	g	a	g	a	g
0	g	0	1	1	1	1	0	1	1	0	1	0	1	0
1	c	1	0	1	1	0	1	0	1	1	1	1	1	1
2	a	1	1	0	1	1	1	1	0	1	1	1	1	1
3	g	1	1	1	1	1	1	1	1	0	1	1	1	1
4	a	1	1	1	1	1	1	1	1	1	0	1	1	1
5	g	1	1	1	1	1	1	1	1	1	1	0	1	1
6	a	1	1	1	1	1	1	1	1	1	1	1	0	1
7	g	1	1	1	1	1	1	1	1	1	1	1	1	0

Ανάλυση του αλγορίθμου Shift-Or σε χρόνο

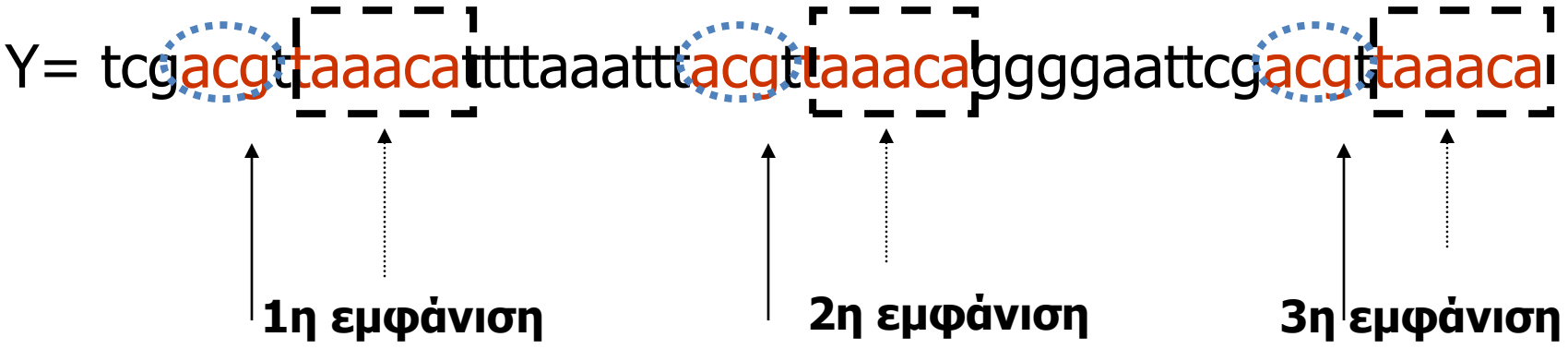
- Πολυπλοκότητα μεθόδου: O(n+m), όπου |T|=n & |P|=m

1. Σε χρόνο O(m*σ) υπολογίζω, τα διανύσματα S_C

Ακριβής Εύρεση για ένα σύνολο προτύπων

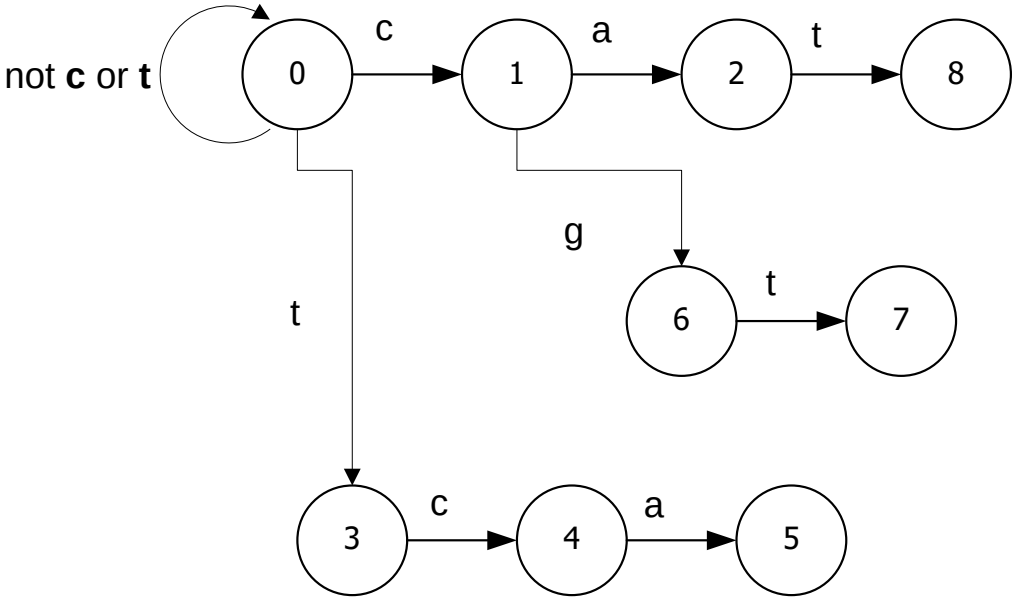
Ορισμός: *«έστω μια ακολουθία χαρακτήρων Υ. Αναζητούμε τις θέσεις εμφάνισης* ενός συνόλου προτύπων P μέσα *στην ακολουθία».*

P= {acg, taaaca}



Το Αυτόματο Aho-Corasick

Έστω P={ca, tca, cgt, cat}



goto function

Η Συνάρτηση “goto”

- g(s, α) = s’: το αυτόματο μεταπηδά στην κατάσταση s’και ο επόμενος χαρακτήρας της ακολουθίας διαβάζεται στην είσοδο,
- g(s, α) = fail, το αυτόματο μεταβαίνει στην κατάσταση s’=f(s) σύμφωνα με τη failure function. Η αναζήτηση συνεχίζεται με τρέχουσα κατάσταση την s’ και σύμβολο εισόδου το χαρακτήρα που ήδη έχει διαβαστεί στην είσοδο → α.

Η Συνάρτηση “failure-function”

- f(s) = 0, για κάθε κατάσταση s, βάθους 1.
- Για να υπολογίσεις το f(s), για κάθε κατάσταση s, βάθους d, θεώρησε όλες τις καταστάσεις r, βάθους d-1 :
  - Αν g(r,α) = fail, για κάθε α, μην κάνεις τίποτα,
  - Διαφορετικά, για κάθε σύμβολο α, έτσι ώστε g(r,α)= s, τότε:
    - Θέσε state = f(r)
    - Εκτέλεσε την εντολή state ← f(state), έως ότου g(state, α) ≠fail
- Θέσε f(s)= g(state, α)

Εφαρμογές εύρεσης προτύπων σε προβλήματα Βιοπληροφορικής

- Αναζήτηση ***Sequence-tagged-site (STS) & Expressed Sequence Tags (ESTs)*** σε ακολουθίες γονιδιωμάτων.
  - ***STS:*** τμήματα του DNA μήκους 200-300 νουκλεοτιδίων
  - ***ESTs:*** τμήματα mRNA & cDNA ακολουθίες που αντιπροσωπεύουν τα τμήματα κωδικοποίησης μιας πρωτεΐνης σε μια ακολουθία γονιδίων.

- Αναζήτηση ***"κανονικών εκφράσεων"*** (regular expressions)
  - [ED]-[EN]-L-[SAN]-x-x-[DE]-x-E-L $\Rightarrow$ **ENLSSEDEEL**

Τέλος Ενότητας

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Σημείωμα Αναφοράς

Σημείωμα Αδειοδότησης

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.