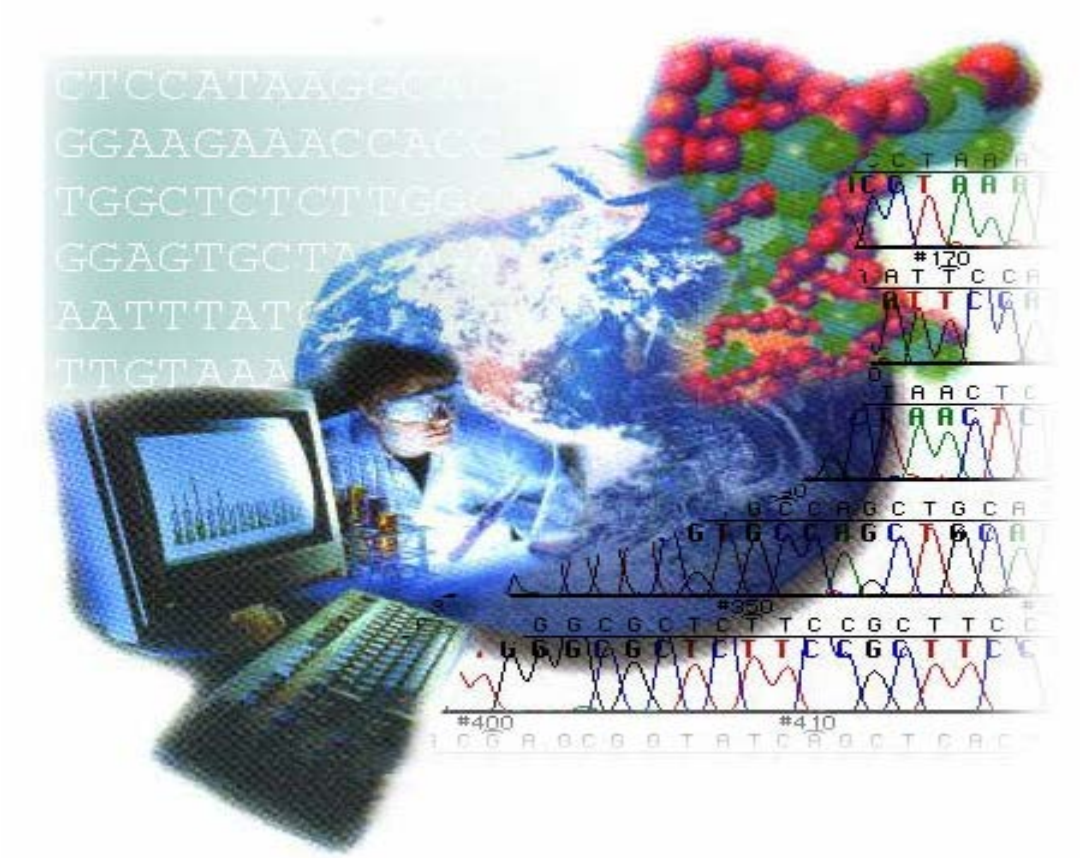
ΤΕΙ ΗΠΕΙΡΟΥ
ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ

ΒΛΑΧΟΥ ΖΑΧΑΡΟΥΛΑ
ΓΙΩΤΙΤΣΑ ΑΝΝΑ

Επιβλέπων καθηγητής:Κωνσταντακόπουλος Φώτης

ΑΡΤΑ 2009-2010

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1.....4
Εισαγωγή.....4
1.2 Ανίχνευση - ρόλος του DNA.....5
1.3 Δομή του DNA……6
1.4 Η διπλή έλικα του DNA.....7
1.4.1Ανίχνευση.....7
1.4.2 Δομή.....7
1.5 Είδη RNA.....8
1.6 Κατάταξη πρωτεϊνών.....9
1.6.1Συνένωση.....10
1.6.2Τύπος δεδομένων.....10
1.7 Σύγχρονη στόχοι της μοριακής βιολογίας.....11
1.8 Τομείς έρευνας στη Βιοπληροφορική.....12
1.9 Ανάλυση πρωτεϊνών.....14

ΚΕΦΑΛΑΙΟ 2.....15
Τεχνικές διαχείρισης και ανάλυσης συμβολοσειρών βιολογικών δεδομένων.....15
2.1 Βασικοί ορισμοί.....15
2.2Το πρόβλημα της ακριβούς εύρεσης προτύπου.....16
2.3Το πρόβλημα της προσεγγιστικής εύρεσης προτύπου.....16

ΚΕΦΑΛΑΙΟ 3.....19
Ανάλυση του αλγορίθμου Boyer-Moore.....19
3.1 Αλγόριθμος Horspool……..19
3.1.1 Ο κώδικας C.....20
3.2 Αλγόριθμος Boyer-Moore…….....20
3.2.1 Ο κώδικας C……23

ΚΕΦΑΛΑΙΟ 4.....34
Ανάλυση του αλγορίθμου Knuth-Morris-Pratt……34
4.1 Ο κωδικός C.....38

ΚΕΦΑΛΑΙΟ 5
Ανάλυση του αλγορίθμου Shift-Or……..48
5.1 Ο κωδικός C.....49
5.2 Ανάλυση των αλγορίθμων και αναφορά σε συμπεράσματα.....50

ΚΕΦΑΛΑΙΟ 6
Αλγόριθμοι σύγκρισης ακολουθιών βιολογικών δεδομένων.....52
6.1 Πίνακες υποκατάστασης και ποινές για τα κενά.....53
6.2 Υπολογισμός της στατιστικής σημαντικότητας.....54
6.3 FASTA.....58
6.3.1Παραλλαγές.....59
6.4 BLAST.....59
6.4.1Παραλλαγές 62
6.5CLUSTALW.....62

ΕΠΙΛΟΓΟΣ.....64

ΒΙΒΛΙΟΓΡΑΦΙΑ.....66

ΚΕΦΑΛΑΙΟ 1^{Ο} : Εισαγωγή

Τα τελευταία χρόνια οι υπολογιστές κατακτούν σημαντική θέση σε κάθε τομέα της ζωής μας αλλά πολύ πιο ενδιαφέρουσα και προκλητική ,σε αρκετούς τομείς διαφόρων επιστημών. Η Βιοπληροφορική αποτελεί ένα σύγχρονο τομέα έρευνας και ανάπτυξης τόσο για τους μοριακούς βιολόγους όσο και για τους επιστήμονες της πληροφορικής.[1]Βιοπληροφορική (bioinformatics) αποκαλείται ένας νέος σχετικά επιστημονικός κλάδος, που προέκυψε από τη συνεργασία των επιστημών της Βιολογίας και της Πληροφορικής . Θεωρώντας τα βιολογικά δεδομένα (DNA, RNA, πρωτεΐνες) ως ψηφιακή πληροφορία, εφαρμόζει αλγορίθμους για την επεξεργασία τους και την παραγωγή χρήσιμων συμπερασμάτων με γρήγορο τρόπο.[1]

Η πρόοδος της τεχνολογίας των υπολογιστών επιτρέπει την προσπάθεια ανάλυσης μέρους των προβλημάτων που προκύπτουν στον τομέα της μοριακής βιολογίας. Λόγω της αύξησης της υπολογιστικής δύναμης και κυρίως της αναπτυγμένης τεχνολογίας των γραφικών, είναι δυνατή η απεικόνιση των διαμορφώσεων της δομής των βιολογικών μορίων στην οθόνη του υπολογιστή. Ακόμα γίνονται προσπάθειες για τη δημιουργία μμεθόδων που θα μπορούν να προβλέπουν τις αλληλεπιδράσεις μμεταξύ των βιολογικών μορίων αλλά και αλγοριθμικών μμεθόδων για την παραγωγή μοριακών δομών με βάση τα ακολουθιακά δεδομένα. Ο μεγάλος αριθμός δεδομένων που μεταφράζονται στην επιστήμη της μοριακής βιολογίας και ειδικότερα στον τομέα της ακολουθιοποίησης του γονιδιώματος (δηλαδή της αλληλουχίας του DNA), αποτελεί μεγάλη πρόκληση για τους επιστήμονες του σχεδιασμού και της ανάλυσης αλγορίθμων.

Συγκεκριμένα η ερμηνεία αυτών των δεδομένων μπορεί να διευκολύνει την αναζήτηση λύσεων αρκετών προβλημάτων όπως είναι η αναγνώριση γονιδίων, ο καθορισμός της δομής των κωδικοποιημένων πρωτεϊνών, η ανακάλυψη των μηχανισμών με τους οποίους οι πρωτεΐνες εκτελούν τη βιολογική λειτουργία τους, η απόκτηση γνώσης για το ρόλο των μη κωδικοποιημένων περιοχών του DNA στη μμορφολογία και έκφραση των γονιδίων.

[2]Για παράδειγμα το DNA μπορεί να θεωρηθεί ως μια ακολουθία χιλιάδων νουκλετιδίων ή βάσεων. Υπάρχουν τέσσερα είδη βάσεων και αντίστοιχα τέσσερα είδη νουκλετιδίων: η αδενίνη, η θυμίνη, η γουανίνη και η κυτοσίνη. Αντιστοιχίζοντας σε καθεμία απ' αυτές ένα σύμβολο, π.χ. "A" για την αδενίνη, "T" για τη θυμίνη, "G" για τη γουανίνη και "C" για την κυτοσίνη, μπορούμε να κατασκευάσουμε τέτοιες συμβολοσειρές (π.χ. "...AAGATCGGTAC..."). Αυτού του είδους η αναπαράσταση είναι ιδιαίτερα βολική για επεξεργασία σε ηλεκτρονικό υπολογιστή και μπορεί να μας δώσει χρήσιμα αποτελέσματα.

Το δε(σ)οξυριβο(ζο)νουκλεϊ(νι)κό οξύ (Deoxyribonucleic acid - DNA) είναι ένα νουκλεικό οξύ που περιέχει τις γενετικές πληροφορίες που καθορίζουν τη βιολογική ανάπτυξη όλων των κυτταρικών μορφών ζωής και των περισσοτέρων ιών. Το DNA συνήθως έχει τη μορφή διπλής έλικας.

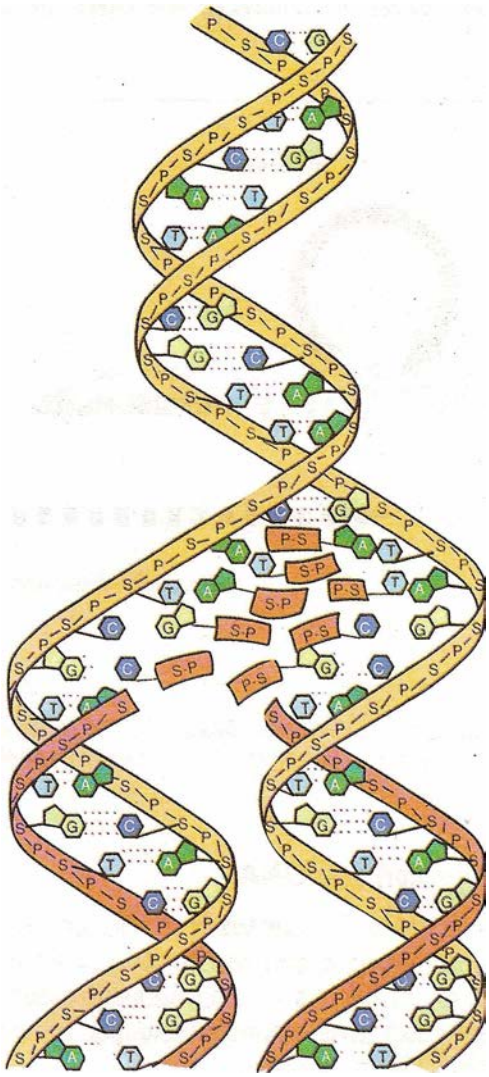
Η αποκωδικοποίηση του DNA, η αποσαφήνιση δηλαδή του τρόπου με τον οποίο η δομή του DNA καθορίζει συγκεκριμένες γενετικές επιλογές, επέτρεψε στους

επιστήμονες να κατανοήσουν καλύτερα την γενετική της ζωής και την κληρονόμηση ορισμένων χαρακτηριστικών και νόσων.

1.2 Ανίχνευση - ρόλος του DNA

Πρόκειται για μια μεγαλομοριακή ένωση που συγκροτείται από αζωτούχες-πρωτεϊνικές βάσεις, φωσφορικές ρίζες και ένα σάκχαρο με πέντε άτομα άνθρακα (πεντόζη), την δε(σ)οξυριβόζη. Στα ευκαριωτικά κύτταρα ανιχνεύεται κυρίως μέσα στον πυρήνα του κυττάρου αλλά και σε μερικά άλλα οργανίδια, όπως τα μιτοχόνδρια και τα πλαστίδια, επιτρέποντάς τους να αναπαράγονται αυτόνομα (ημιαυτόνομα οργανίδια).

Το σύνολο των μορίων DNA που υπάρχουν σε ένα κύτταρο αποτελούν το γενετικό υλικό του. Το DNA είναι ο φορέας των γενετικών πληροφοριών του κυττάρου, όχι μόνον με την έννοια της μεταβίβασης χαρακτηριστικών, αναλλοίωτων από γενεά σε γενεά, αλλά και της ρύθμισης της φυσιογνωμίας εξειδίκευσης κάθε κυττάρου για την επιτέλεση των ιδιαίτερων λειτουργιών του. Τέλος, το DNA επιτρέπει τη δημιουργία γενετικής ποικιλότητας, υφιστάμενο μεταλλάξεις.

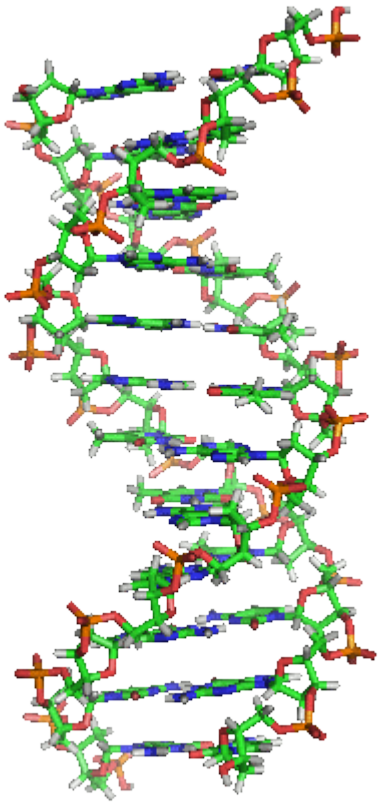


1.3 Δομή του DNA

Η διαμόρφωση των μεγάλων μορίων του DNA στο χώρο έχει τη μορφή δύο επιμηκών αλύσεων, οι οποίες συστρέφονται ελικοειδώς μεταξύ τους. Οι αζωτούχες βάσεις στο DNA είναι τέσσερις:

- κυτοσίνη C
- γουανίνη G
- θυμίνη T
- αδενίνη A

Οι αζωτούχες βάσεις, ανάλογα με την σειρά αλληλουχίας τους σε τριάδες, κωδικοποιούν το μήνυμα για τη σύνθεση των αμινοξέων του κυττάρου στα ριβοσώματα. Εκεί τα αμινοξέα συνδυάζονται, με τη σειρά κατά την οποία μεταφέρθηκαν στο ριβόσωμα και συντίθενται έτσι οι διαφορετικές πρωτείνες.



Τρισδιάστατη απεικόνιση της δομής ενός τμήματος DNA

1.4 Η διπλή έλικα του DNA

Το 1953 οι Τζέιμς Γουάτσον (J. Watson), και Φράνσις Κρίκ, (F. Crick), δύο Βρετανοί ερευνητές που εργάζονταν στο Πανεπιστήμιο του Κέμπριτζ παρουσίασαν ένα "μοντέλο" της δομής του DNA, που ονομάσθηκε "μοντέλο της διπλής έλικας". Σύμφωνα με το μοντέλο αυτό το μόριο του DNA παρουσιάζεται με τα ακόλουθα τρία βασικά χαρακτηριστικά:

Αποτελείται από δύο πολυνουκλεοτιδικές σε μορφή δύο αντιτακτών κλώνων που σχηματίζουν δεξιόστροφη διπλή έλικα. Οι αζωτούχες βάσεις (ή πρωτεϊνικές) κάθε κλώνου είναι κάθετες ως προς τον άξονα του μορίου και προεξέχουν προς το εσωτερικό της συστροφής. Οι δύο δημιουργούμενοι κλώνοι συγκρατούνται μεταξύ τους με δεσμούς υδρογόνου. Τα δε ζευγάρια των αζωτούχων βάσεων όπου αναπτύσσονται μεταξύ τους δεσμοί υδρογόνου είναι καθορισμένα: η αδενίνη με τη θυμίνη και η γουανίνη με την κυτοσίνη.

Το ριβονουκλεοτιδικό οξύ, ή ορθότερα ριβοζονουκλεικό οξύ, και συντομογραφικά **RNA**, (προφέρεται «αρ-εν-έι»), είναι μία τις δύο κατηγορίες των πολυμερών νουκλεικών οξέων στο κύτταρο. Αποτελείται από μονομερή νουκλεοτίδια που παίζουν σημαντικό ρόλο στη διαδικασία της μετάφρασης του γενετικού κώδικα από την έτερη κατηγορία νουκλεϊκού οξέος, το δεοξυριβονουκλεικό οξύ (συντομογραφικά DNA), σε πρωτεϊνικά προϊόντα. Το RNA χαρακτηρίζεται ως ο «αγγελιοφόρος» μεταξύ του DNA και των πρωτεϊνικών συμπλεγμάτων που είναι γνωστά σαν ριβοσώματα στο κυτταρόπλασμα του κυττάρου (αγγελιαφόρο RNA, mRNA). Έτσι το RNA μαζί με το DNA αποτελούν το γενετικό υλικό των οργανισμών.

1.4.1 Ανίχνευση

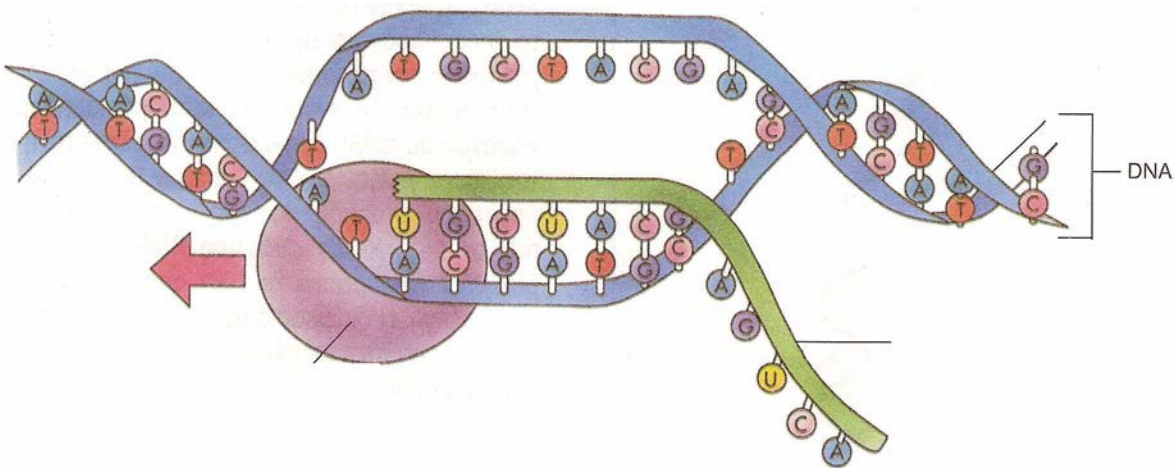
Στα βακτηριακά κύτταρα το μεγαλύτερο μέρος του απαντώμενου RNA εντοπίζεται στο κυτταρόπλασμα, ενώ μια ακόμη ποσότητα (κατά το στάδιο της βιοσύνθεσης) εντοπίζεται να συνδέεται με μη ομοιοπολικούς χημικούς δεσμούς με το DNA.

Επίσης το RNA εντοπίζεται σε όλα τα είδη των ευκαρυωτικών κυττάρων. Για παράδειγμα στα ηπατικά κύτταρα περίπου το 11% της συνολικής ποσότητας RNA απαντάται στον πυρήνα, το 15% στα μιτοχόνδρια, ένα 24% στο κυτοσόλυο και το υπόλοιπο 50% στα ριβοσώματα.

1.4.2 Δομή

Από χημικής άποψης το RNA είναι όμοιο με το DNA. Και οι δύο αυτές κατηγορίες νουκλεϊκών οξέων είναι μακρομοριακές ενώσεις μεγάλου μοριακού βάρους. Το μακρομόριο του RNA αποτελείται από επαναλαμβανόμενες δομικές μονάδες τα νουκλεοτίδια. Το μόριο του RNA περιλαμβάνει (όπως και του DNA), τέσσερις τύπους νουκλεοτιδίων που συνδέονται μεταξύ τους με 3'-5' φωσφοδιεστερικούς δεσμούς. Ωστόσο κύρια διαφορά του RNA από το DNA είναι ότι το μόριό του είναι μονόκλωνο έναντι του δίκλωνου του DNA, αποτελείται δηλαδή από μια μόνο αλυσίδα, ανάλογη της μιας εκ των δύο εκείνων της διπλής έλικας του DNA. Βασική επίσης διαφορά είναι ότι το σάκχαρο στα νουκλεοτίδια του είναι η ριβόζη, εξ ου και η ονομασία τους ριβονουκλεοτίδια, αντί της δεοξυριβόζης στο DNA, και ότι περιέχει την πυριμιδίνη ουρακίλη αντί της θυμίνης (που υπάρχει στο μόριο του DNA), χωρίς να είναι γνωστός ο λόγος της τελευταίας αυτής διαφοράς. Η μακρομοριακή πολυνουκλεοτιδική αλυσίδα του RNA εμφανίζεται από ελικοειδής μέχρι ευθύγραμμη.

Ένας άλλο τύπος RNA πέραν του mRNA, μεταφορικό RNA (tRNA), αποτελεί δομικό συστατικό των ριβοσωμάτων και επιτελεί σημαντικές μεταφορές αμινοξέων που προορίζονται για την πρωτεινοσύνθεση. Το mRNA μεταγράφει τον γενετικό κώδικα από το DNA με τη βοήθεια κυρίως ενός ενζύμου που ονομάζεται RNA |πολυμεράση, ενώ στη συνέχεια υφίσταται επεξεργασία από έναν αριθμό άλλων, δευτερευόντων ενζύμων. Ακολούθως το τελικό mRNA χρησιμοποιείται ως βάση για τη μετάφραση των γονιδίων σε πρωτεΐνες, με τη βοήθεια του tRNA το οποίο μεταφέρει αμινοξέα στα ριβοσώματα για να δημιουργηθούν πρωτεΐνες (πάντα χάρη στην αρχή της συμπληρωματικότητας των βάσεων).[2]



1.5 Είδη RNA

[3]

- Αγγελιοφόρο RNA mRNA (Messenger RNA)
- Μεταφορικό RNA tRNA (Transfer RNA)
- Ριβοσωμικό RNA rRNA (Ribosomal RNA)
- Μη κωδικοποιητικό RNA (Non-coding RNA)
- Καταλυτικό RNA (Catalytic RNA)

Οι πρωτεΐνες αποτελούν τα πιο διαδεδομένα και πολυδιάστατα τόσο στη μορφή όσο και στη λειτουργία τους μακρομόρια. Ακόμη και σ΄ ένα απλό κύτταρο των βακτηρίων εντοπίζονται εκατοντάδες διαφορετικές πρωτεΐνες που κάθε μια εξ αυτών έχει ιδιαίτερο ρόλο. Οι πρωτεΐνες αποτελούν είτε το δομικό συστατικό του κυττάρου είτε συνεργούν σε κάποια συγκεκριμένη λειτουργία.

Οι πρωτεΐνες είναι μεγάλα σύνθετα βιομόρια, με μοριακό βάρος από 10.000 μέχρι πάνω από 1 εκατομμύριο), αποτελούμενα από αμινοξέα, τα οποία ενώνονται μεταξύ τους με πεπτιδικούς δεσμούς σχηματίζοντας μια γραμμική αλυσίδα, καλούμενη αλυσίδα πολυπεπτιδίων. Όλες οι πρωτεΐνες περιέχουν άνθρακα, οξυγόνο και άζωτο και οι περισσότερες εξ αυτών και θείο.

Η ακολουθία αμινοξέων σε μια πρωτεΐνη καθορίζεται από ένα γονίδιο και κωδικοποιείται κατά τον γενετικό κώδικα DNA. Παρόλο που ο γενετικός κώδικας κωδικοποιεί 20 αμινοξέα, τα αμινοξέα που συνιστούν την πρωτεΐνη συχνά υφίστανται χημικές αλλαγές κατά τη μεταγραφική κωδικοποίηση: είτε προτού να μπορέσει η πρωτεΐνη να λειτουργήσει στο κύτταρο, είτε ως τμήμα των μηχανισμών ελέγχου.

Περισσότερες από μια πρωτεΐνες συχνά λειτουργούν μαζί για να επιτύχουν κάποια συγκεκριμένη λειτουργία, ή μπορεί ακόμα και να συσσωματωθούν για να διαμορφώσουν τα στα **Σχηματισμός – οργάνωση**

Οι πρωτεΐνες παράγονται στο κυτόπλασμα και συγκεκριμένα στα ριβοσώματα όπου ξεκινούν ως απλές μη διακλαδωμένες αλληλουχίες αμινοξέων, δηλαδή πεπτιδίων ή πολυπεπτιδίων, σχηματίζοντας την "πρωτοταγή δομή", επί της οποίας και για την οποία καθοριστικοί παράγοντες είναι τα νουκλεικά οξέα, τα οποία και φέρονται να ελέγχουν όλες τις λειτουργίες αλλά και τα κληρονομικά γνωρίσματα των οργανισμών.

Στη συνέχεια όλα τα πρωτεϊνικά μόρια υφίστανται μια φυσική αναδιάταξη προκειμένου να δώσουν μια "δευτεροταγή δομή" η οποία προκαλείται από δεσμούς υδρογόνου μεταξύ των καρβοξυλομάδων και των αμινομάδων των αμινοξέων. Κατά τη δευτεροταγή δομή δε λαμβάνονται υπ' όψιν οι αλληλεπιδράσεις μεταξύ των πλευρικών ομάδων των αμινοξέων. Ο πλέον διαδεδομένος τύπος τέτοιας μορφής είναι η λεγόμενη "α-έλικα", δεξιόστροφη, όπου οι σπείρες διατηρούνται στη θέση τους με δεσμούς υδρογόνου μεταξύ των καρβοξυ- και αμινομάδων. Μια άλλη δευτεροταγής δομή είναι η λεγόμενη "β-πτυχωτή επιφάνεια" όπου στη περίπτωση αυτή διασταυρώνονται παράλληλες αλυσίδες πολυπεπτιδίων που ενώνονται στις διασταυρώσεις με δεσμούς υδρογόνου σχηματίζοντας έτσι μια εξαιρετικά σφιχτή δομή, όπως στο μετάξι. Οι πρωτεΐνες με τέτοιες σχετικά απλές δισδιάστατες δευτερογενείς δομές ονομάζονται γενικά ινώδεις πρωτεΐνες. Παραταύτα οι πρωτεΐνες υφίστανται ακόμα ποιο περίπλοκο δίπλωμα (πτύχωση) το οποίο καλείται "τριτοταγής δομή". Με τον όρο τριτοταγή δομή, εννοούμε το τελικό και λειτουργικό σχήμα που αποκτά η πρωτεΐνη μετά κι από την αλληλεπίδραση των πλευρικών ομάδων των αμινοξέων (π.χ. σχηματισμός δισουλφιδικών δεσμών μεταξύ δύο κυστεϊνικών καταλοίπων). Τέλος, υπάρχουν και πρωτεΐνες που αποτελούνται από πολλές πολυπεπτιδικές αλυσίδες που είναι χαλαρά ενωμένες και αυτό αποτελεί τη λεγόμενη "τεταρτοταγή δομή". Παράδειγμα είναι η αιμοσφαιρίνη.[3]

1.6 Κατάταξη πρωτεϊνών

[4]Γενικά οι πρωτεΐνες ανάλογα της μορφής τους διακρίνονται σε ινώδεις πρωτεΐνες και σε σφαιρικές πρωτεΐνες.Με κριτήριο τη σύνθεσή τους διακρίνονται σε απλές (όταν αποτελούνται μόνο από αμινοξέα) και σε σύνθετες (όταν στο μόριό τους περιλαμβάνονται και μη πρωτεϊνικά τμήματα όπως μέταλλα, σάκχαρα, λίπη κ.λπ.). Επίσης με κριτήριο ακόμη τη λειτουργία τους διακρίνονται σε δομικές (όταν αποτελούν τα δομικά υλικά του κυττάρου), και λειτουργικές (όταν συμβάλλουν σε κάποιες λειτουργίες).
Βιολογικός ρόλος
Οι διάφορες λειτουργίες που παρατηρούνται στους οργανισμούς γίνονται χάρη στις πρωτεΐνες. Ο δε βιολογικός τους ρόλος καθορίζεται κάθε φορά από την τρισδιάστατη δομή τους που είναι συνέπεια της αλληλουχίας των αμινοξέων, η οποία και ξεκινά από την πρωτοταγή δομή.

Όπως άλλα βιολογικά μακρομόρια (π.χ. οι πολυσακχαρίτες, τα λιπίδια, και νουκλεικά οξέα) έτσι και οι πρωτεΐνες είναι απαραίτητες για όλους τους ζωντανούς οργανισμούς και συμμετέχουν σε κάθε διαδικασία μέσα στα κύτταρα. Πολλές πρωτεΐνες δρουν ως ένζυμα που καταλύουν τις βιοχημικές αντιδράσεις, και είναι ζωτικής σημασίας στο μεταβολισμό. Άλλες πρωτεΐνες έχουν δομικές ή μηχανικές λειτουργίες, όπως οι πρωτεΐνες του κυτταρικού σκελετού, οι οποίες συμβάλλουν στη διατήρηση της μορφής των κυττάρων. Οι πρωτεΐνες είναι επίσης σημαντικές στη διακυτταρική επικοινωνία, τη δράση του ανοσοποιητικού συστήματος, τον

σχηματισμό κυτταρικών ιστών και τον κυτταρικό κύκλο.Οι πρωτεΐνες είναι απαραίτητα συστατικά στη διατροφή μας, δεδομένου ότι τα ζώα δεν μπορούν να συνθέσουν όλα τα αμινοξέα, αλλά πρέπει να τα λάβουν από τα τρόφιμα. Μέσω της διαδικασίας της πέψης, τα ζώα αποικοδομούν την πρωτεΐνη στα ελεύθερα αμινοξέα που μπορούν να χρησιμοποιηθούν για την πρωτεϊνική σύνθεση.

Συμβολοσειρά, (*string*), στην επιστήμη των υπολογιστών καλούμε μια σειρά διαδοχικών συμβόλων τα οποία σύμβολα είναι στοιχεία ενός πεπερασμένου συνόλου ή αλλιώς αλφάβητου. Ανάλογα με τα στοιχεία που διαθέτει το αλφάβητο, διαφοροποιείται και το περιεχόμενο της συμβολοσειράς.Στην περίπτωση που το αλφάβητο περιορίζεται σε γράμματα και αριθμούς, η συμβολοσειρά λέγεται και αλφαριθμητικό.Στον προγραμματισμό υπολογιστών , συνήθως με τον όρο συμβολοσειρά αναφερόμαστε σε έναν τύπο δεδομένων με τον οποίο ορίζονται ακολουθίες χαρακτήρων. Οι χαρακτήρες παριστάνονται με μια συγκεκριμένη κωδικοποίηση χαρακτήρων.

1.6.1 Συνένωση (ή παράθεση)

Έστω μια συμβολοσειρά u και μια συμβολοσειρά v ενός αλφαβήτου Σ. Η συνένωση (concatenation) της u με τη v συμβολίζεται uv και είναι μια καινούργια συμβολοσειρά που προκύπτει από τη u ακολουθούμενη από τη v. Για παράδειγμα:
Για κάθε συμβολοσειρά w ισχύει wε = εw = w, όπου ε η κενή συμβολοσειρά

1.6.2 Τύπος δεδομένων

Οι γλώσσες προγραμματισμού επιτρέπουν τον ορισμό και την διαχείριση σειρών χαρακτήρων.Οι χαρακτήρες είναι γράμματα (όπως a b c α β γ ...), αριθμοί (όπως 0 1 2 3 ...), τελεστές πράξεων (όπως * + / - ...), σημεία στίξης (όπως κόμμα, δίστιγμο, παρένθεση, ...), σύμβολα σχεδίασης (όπως ╚ ╤ ╧ ...), σύμβολα ελέγχου (όπως ESCape, LineFeed, ACKnowledgement, CarriageReturn, ...), οι οποίοι ανήκουν σε κάποιον προκαθορισμένο κώδικα (όπως ASCII, Ebcdic, Fieldata, Unicode, ISO 8859, UTF-16, κλπ).Ανάλογα με την γλώσσα προγραμματισμού κάποιοι χαρακτήρες μπορεί να έχουν ειδικό ρόλο, οπότε περιορίζεται η χρήση τους. Μερικοί χαρακτήρες είναι εκτυπώσιμοι και μερικοί δεν είναι.Για παράδειγμα, αν υποθέσουμε ότι το αλφάβητο της σειράς χαρακτήρων είναι το ελληνικό, τότε τα παρακάτω είναι αποδεκτές σειρές χαρακτήρων:

αβγ
ααββγγ
αυτό είναι σειρά χαρακτήρων

Οι σειρές χαρακτήρων στις γλώσσες προγραμματισμού συνήθως περικλείονται με διπλά ή μονά εισαγωγικά (αγγλ. quote) .[4]

1.7 Σύγχρονοι Στόχοι της Μοριακής Βιολογίας

Οι σύγχρονοι στόχοι της επιστημής της Μοριακής Βιολογίας επικεντρώνονται στις ακόλουθες περιοχές:
Η Ακολουθωποίηση και Σύγκριση τουΓουιδιωμάτωυ διαφορετικών Οργανισμών. Οργανισμοί οι οποίοι είναι συγγενείς έχουν κοινές πρωτεΐνες, οπότε η σύγκριση διαφορετικών γονιδιωμάτων μπορεί να δώσει σημαντικές πληροφορίες για την εξελικτική πορεία των οργανισμών. Επίσης η σύγκριση των ακολουθιών DNA μεταξύ διαφορετικών ειδών βοηθά σημαντικά στη μελέτη και την κατανθηση των σχέσεων μεταξύ τους .

Η Αναγνώριση Γονιδίων και καθορισμός των λειτουργιών που ρυθμίζουν.Η ακολουθία του DNA παρουσιάζει ορισμένα δομικα Χαρακτηριστικά κρίσιμα για τη λειτουργία του πως τα σημεία πρθσδεσης στα οποία συνδέονται πρωτεΐνες η συμπλέγματα πρωτεϊνών. Η μελέτη αυτών των περιοχών βοηθά στην αναγνώριση των γονιδίων και στον καθορισμό των ιδιοτήτων τους.
η Κατανόηση της Γονιδιακής έκφρασης. Κάθε γονίδιο δραστηριοποιείται μέσα στο κύτταρο μετα την παραγωγή της αντίστοιχης πρωτεΐνης. Η έκφραση των γονιδίων ποικίλλει στα διάφορα κύτταρα και σε διαφορετικές χρονικές στιγμές. Παράγοντες που επηρεάζουν τη γονιδιακή έκφραση είναι το ίδιο το περιβάλλον του κυττάρου, η λήψη βιοχημικών σημάτων και η παρουσία πρωτεϊνών. Η κατανθηση της γονιδιακής έκφρασης βοηθά στην κατανθηση βασικών κυτταρικών λειτουργιών όπως ο πολλαπλασιασμός, η αναπνοή και ο μεταβολισμός καθώς και η Κατανόηση Γενικών Ασθενειών. Η μμετάλλαξη των γονιδίων ευθύνεται για ένα σύνολο ασθενειών όπως ο διαβήτης, ο καρκίνος κ.α. Η κατανόηση του τρόπου με τον οποίο τα γονίδια επηρεάζουν τις ασθένειες και η κατανόηση των λειτουργιών των πρωτεϊνών, που τα γονίδια κωδικοποιούν, μπορεί να βοηθήσει στην ανάπτυξη θεραπείας που στοχεύει στον περιορισμό και τη βελτίωση ελαττωματικών γονιδίων. Λαμβάνοντας υπόψιν ότι η προδιάθεση των παιδιών για ορισμένες ασθένειες είναι κωδικοποιημένη στο γενετικό υλικό με την κατάλληλη προληπτική ιατρική θεραπεία μπορουν έγκαιρα να αποφευχθούν οι παράγοντες που μπορουν να εκδηλώσουν τη συγκεκριμένη ασθένεια .

Παρατηρούμε λοιπόν ότι η Μοριακή Βιολογία ασχολείται σε μμεγάλο βαθμό με τη σύγκριση και την κατηγοριοποίηση μοριακών δομών όπως τα γονίδια, οι ακολουθίες του DNA, και οι πρωτεΐνες. Τα δεδομένα αυτά προέρχονται από πειραματικά δεδομένα και έρευνες ή από υπάρχουσες βάσεις μοριακών δεδομένων. Η χρήση των εργαλείων της Πληροφορικής μπορεί να επιλύσει αρκετά υπολογιστικά προβλήματα που προκύπτουν όπως:

1) Διασύνδεση της Γονιδιακής ακολουθίας. Οι σύγχρονες μμέθοδοι ανάγνωσης της ακολουθίας του DNA βασίζονται στη σταδιακή ανάγνωση τμημάτων (fragments) από το υπό μελέτη μόριο, που μπορεί να φθάνει και τις χιλιάδες βάσεις αμινοξέων. Η διαδικασία επανασύνδεσης υπόκειται σε σφάλματα και αποτελεί μια πολύτιμη αλλά ταυτόχρονα πολύπλοκη διαδικασία.

2) Σύγκριση ακολουθιών. Υπάρχει μια βασική αρχή η οποία θέλει τις ακολουθίες του DNA και των πρωτεϊνών που μμοιάζουν να εμφανίζουν παρόμοια λειτουργία. Αυτό ισχύει και στην περίπτωση που οι ακολουθίες αυτές προέρχονται από διαφορετικά είδη. Για αυτό το λόγο το πρώτο βήμα στην αναγνώριση της δράσης μιας ακολουθίας είναι η σύγκριση της με άλλες

για να ερευνήσουμε πιθανές ομοιότητες στη δομή. Σε αυτό το σημείο εμπλέκονται αλγόριθμοι ανάκτησης πληροφορίας βάσει σχηματικών ομοιοτήτων.

3) Κατηγοριοποίηση του πρωτεϊυώυ. Οι πρωτεΐνες κατηγοριοποιούνται σε οικογένειες με παρόμοια δομή και λειτουργία. Με αυτό τον τρόπο μπορούμε να γνωρίζουμε τη συμπεριφορά κα ι την τρισδιάστατη δομή τους.

4) Εξαγωγή πληροφοριών από γονιδιακές ακολουθίες. Η μελέτη γονιδιακών ακολουθιών μπορεί να βοηθήσει στην εξαγωγή χρήσιμων αποτελεσμάτων γύρω από τη συμπεριφορά και τη βιολογική δράση των γονιδίων (εμπλοκή σε συγκεκριμένες ανωμαλίες, όμοια συμπεριφορά
σε θεραπευτικές αγωγές κ.ά). Η πολύπλοκη φύση των γονιδίων κάνει πολύ δύσκολη την όλη διαδικασία. Η Αναπαράσταση των Κυττάρων ως μμεταγραφικών διτύων. Ένα ζωντανό κύτταρο μπορεί να χαρακτηριστεί ως μια αλληλεπίδραση διαφορετικών κυτταρικών διαδικασιών. Αυτό μπορεί να μμοντελοποιηθεί ως ένα δυναμικό συ στη μα με συγκεκριμένες εισόδους (π.χ.: φάρμακα, λαμβανόμενα σήματα από γειτονικά κύτταρα η τον ανθρώπινο οργανισμό) και πιθανές καταστάσεις.

1.8 Τομείς Έρευνας στη Βιοπληροφορική

Οι στόχοι της Βιοπληροφορικής μπορουν να ταξινομηθούν σε 3 ομάδες. Σε ένα πρωτο επίπεδο η Βιοπληροφορικη επιτρέπει την αποδοτική οργάνωση των δεδομένων ώστε να είναι δυνατή η αποθήκευση, ανάκτηση και ενημέρωση τους. Σημαντικό παράδειγμα αποτελεί η βάση δεδομένων της δομης τρισδιάστατων μορίων Protein Data Bank.

Σε ένα δεύτερο επίπεδο η Βιοπληροφορική περιλαμβάνει τα εργαλεία που επιτρέπουν την ανάλυση των βιολογικών δεδομένων. Για παράδειγμα έχοντας ακολουθιοποιήσει μια πρωτεΐνη, οι επιστήμονες ενδιαφέρονται να τη συγκρίνουν με ήδη γνωστές και ταυτοποιημένες ακολουθίες. Αυτή η διαδικασία απαιτεί τη χρήση πολύπλοκων εργαλείων όπως τα προγράμματα FASTA και PSI-BLAST, που επιτρέπουν την ανακάλυψη και αναζήτηση κοινών τμημάτων σε βιολογικές ακολουθίες. Τέλος σε ένα τρίτο επίπεδο η Βιοπληροφορικη θέτει ως στόχο την ανάπτυξη εργαλείων που επιτρέπουν την ερμηνεία των αποτελεσμάτων βιολογικής σημασίας.

Οι ακολουθίες ΩΝΑ αποτελούν συμβολοσειρές (strings) πάνω σε ένα αλφάβητο 4 γραμμάτων- βάσεων. Κάθε ακολουθία μπορεί να έχει μήκος έως και 1000 βάσεις. Η βάση δεδομένων Gen Bank, περιέχει μέχρι στιγμής 11.5 εκατομμύρια εγγραφές. Αντίστοιχα οι ακολουθίες πρωτεϊνών αποτελούν συμβολοσειρές (strings) πάνω σε ένα αλφάβητο 20 γραμμάτων- αμινοξέων. Μέχρι στιγμης υπάρχουν 400.000 γνωστές ακολουθίες πρωτεϊνών και μια τυπική πρωτεΐνη βακτηρίου περιέχει περίπου 300 αμινοξέα. Από την άλλη πλευρά οι βάσεις δεδομένων των δομων των μακρομορίων αποτελούν μια πολύπλοκη δομη πληροφοριών. Στην Protein Data Bank περιλαμβάνονται 15.000 εγγραφές που περιλαμβάνουν τις ατομικές δομές πρωτεϊνών, ΩΝΑ και RNA που έχουν καθοριστεί από x-ray κρυσταλλογραφία και NMR. Μια τυπική εγγραφή στην ΡΩΒ για μια μμεσαίου μμεγέθους πρωτει\ιη περιλαμβάνει την Μοριακή Μοντελοποίηση .

Η Μοριακή Μοντελοποίηση, αποτελεί ένα νέο και ταυτόχρονα γοργά αναπτυσσόμενο επιστημονικό κλάδο που συνδυάζει σε μμεγάλο βαθμό τις επιστήμες της Βιολογίας και της Πληροφορικής. Η Μοριακή Μοντελοποίηση προσπαθεί να μιμηθεί τη συμπεριφορα των μοριακων συστημάτων, βασιζόμενη σε μμεγάλο βαθμό στη σχεδίαση μμοντέλων μορίων με τη βοήθεια ηλεκτρονικού υπολογιστή.

Τα σύγχρονα πακέτα λογισμικού μοριακης σχεδίασης, αποτελούν χρήσιμα εργαλεία στα χέρια των ερευνητών, οι οποίοι έχουν τη δυνατότητα να συνδυάσουν τη θεωρία και το πείραμα. Το μόνο ίσως μμειονέκτημα είναι ότι δεν υπάρχει ένα γενικότερο και διευρυμένο εργαλείο μοριακης σχεδίασης. Το σύνολο των βιολογικών μορίων που μελεταμε στη μοριακη σχεδίαση ποικίλλει από μεμονωμένα μόρια και απλές ατομικές δομές σε πολυμερή και βιολογικά μακρομόρια όπως οι πρωτεΐνες και το DNA. Έτσι το επιλεγόμενο κάθε φορά μμοντέλο πρέπει σαφώς να καλύπτει τις ιδιαιτερότητες και τα χαρακτηριστικά του προς σχεδίαση συστήματος.

Η Μοριακή Μοντελοποίηση ακολουθεί τα εξής 3 βασικά βήματα: α) επιλογή του κατάλληλου μμοντέλου που περιγράφει ικανοποιητικά τις ενδομοριακές και εσωμοριακές συσχετίσεις του μορίου, β) υπολογισμός της ενεργειακής κατάστασης του συστήματος και ελαχιστοποίηση της και γ) ανάλυση των παραπάνω υπολογισμών και έλεγχος της τελικής διαμόρφωσης ώστε να ικανοποιούνται όλες οι συνθήκες και περιορισμοί που ο σχεδιαστής έχει θέσει.

Αν και στη Μοριακή Σχεδίαση το σύνολο των συντεταγμένων παρέχει μια χρήσιμη απεικόνιση του μορίου σε δισδιάστατο και τρισδιάστατο επίπεδο, είναι αναγκαία και η χρήση συμπληρωματικών πληροφοριών που έχουμε στη διάθεση μας προκειμένου να αναπαραστήσουμε και τις βιοχημικές ιδιότητες του μορίου. Σημαντική πρόκληση σε αυτή την κατεύθυνση αποτελεί η ικανοποιητική απεικόνιση των επιφανειών των μορίων οι οποίες είναι υπεύθυνες για τη συμπεριφορα των μορίων και τις φυσικοχημικές τους ιδιότητες.

Οι σύγχρονες τεχνικές των μοριακών γραφικών (molecu1ar graphics) επιτρέπουν την απεικόνιση της τρισδιάστατης αρχιτεκτονικής των μορίων στην ενεργειακά ευνοϊκότερη διαμόρφωση. Στη δομή αυτή είναι δυνατόν να επέμβουμε απομονώνοντας τμήματα των μορίων, αλαζονείας τον προσανατολισμό ορισμένων ομάδων ή ψάχνοντας για άλλες δυνατές διαμορφώσεις. Τα μοριακά αυτά μοντέλα επιτρέπουν επίσης την απεικόνιση φυσικοχημικών Χαρακτηριστικών που επηρεάζουν τις αλληλεπιδράσεις ενός μορίου με άλλα μόρια. Είναι π.χ. δυνατόν να παρασταθούν οι ακτίνες Van der Waa1s των ατόμων, η μοριακή επιφάνεια η ο μοριακός όγκος, το μοριακό ηλεκτροστατικό δυναμικό, η ηλεκτρονιακή πυκνότητα κ.α. Η κατασκευή μοριακών μοντέλων επιτρέπει εξάλλου τη σύγκριση ανάμεσα σε διαφορετικά μόρια προσδιορίζοντας περιοχές ομοιοτήτων και διαφορών. Παράλληλα είναι δυνατή η απεικόνιση της τρισδιάστατης προσαρμογής ενός μικρομορίου (φαρμάκου) σε ένα μακρομόριο (υποδοχέα). Σε αυτή την κατεύθυνση είναι δυνατό να σχεδιαστούν στην οθόνη του ηλεκτρονικού υπολογιστή νέα μόρια τα οποία μιμούμενα το σχήμα μιας δραστικής ένωσης η του φυσιολογικού υποστρώματος έχουν θεωρητικά τη δυνατότητα να καταλάβουν την ενεργό θέση ενός υποδοχέα .

1.9 Ανάλυση Πρωτεϊνών

Όπως έχουμε ήδη αναφέρει οι πρωτεΐνες περιγράφονται πλήρως από την αμινοξεϊκη τους ακολουθία, όμως διακρίνονται και για ειδικές λειτουργίες λόγω της τρισδιάστατης δομής τους. Αν και η δομή μιας πρωτεΐνης αποτελεί το κλειδί για τη βιολογική της λειτουργία, για πολλές πρωτεΐνες η επίλυση της δομής τους δεν είναι αρκετή για να καθοριστεί η λειτουργία τους. Πολλα ένζυμα εντείνουν την καταλυτική τους λειτουργία με βάση μια μικρή περιοχή στην πρωτεϊνική επιφάνεια που ονομάζεται ενεργός περιοχή (active site) η ενεργό κέντρο του ενζύμου. Αυτή η περιοχή χαρακτηρίζεται από γεωμετρικά και φυσικοχημικά. Χαρακτηριστικά που είναι σχεδόν συμπληρωματικά ενός άλλου μορίου, του υποστρώματος. Έτσι το ενεργό κέντρο μιας πρωτεΐνης ενεργεί σαν υποδοχέας. Αυτή η διαδικασία πρόσδεσης υποδοχέα και υποστρώματος καλείται προσάραξη (docking). Η προσπάθεια εντοπισμού του ενεργού κέντρου μιας πρωτεΐνης και της κατανόησης με ακρίβεια της διαδικασίας προσάραξης αποτελεί ένα πολύ σημαντικό βήμα στην προσπάθεια αποκρυπτογράφησης των περισσότερων μεταβολικών αντιδράσεων. Με την κατανόηση της πρωτεϊνικής λειτουργίας ο σχεδιασμός φαρμάκων μπορεί να αναπτυχθεί σημαντικά. Εδώ πρέπει να συμπληρώσουμε ότι προκειμένου μια πρωτεΐνη να βρεθεί σε μια ενεργειακή ισορροπία (ιδανική για την προσάραξη της) περνά από ένα σύνολο στεροδιαμορφώσεων. Υπάρχουν εκατομμύρια διαμορφώσεις οι οποίες μπορούν να διαφέρουν σημαντικά.

ΚΕΦΑΛΑΙΟ 2^{Ο}:Τεχνικές Διαχείρισης και Ανάλυσης Συμβολοσειρών Βιολογικών Δεδομένων

Βασικός στόχος των αλγορίθμων και τεχνικών διαχείρισης και ανάλυσης συμβολοσειρών βιολογικών δεδομένων είναι η σύγκριση ακολουθιών προκειμένου να ανιχνευθούν κοινά μοτίβα που καθορίζουν τη δομική και λειτουργική ιδιότητα των βιολογικών μορίων, καθώς και η αναγνώριση επαναλαμβανόμενων μοτίβων ή αλλιώς περιοδικοτήτων (regua1arities) που καθορίζουν τη βιολογική σημασία κάθε μακρομορίου.

Η βασική υπόθεση για την ανάπτυξη και εφαρμογή τεχνικών διαχείρισης συμβολοσειρών βιολογικών δεδομένων είναι ότι κάθε βιολογικό μόριο μπορεί να περιγραφεί ως μια ακολουθία συμβόλων από ένα ορισμένο αλφάβητο Σ. Συγκεκριμένα κάθε μόριο του DNA, μπορεί να θεωρηθεί ως μια ακολουθία συμβόλων (συμβολοσειρά), από ένα αλφάβητο τεσσάρων χαρακτήρων /γραμμάτων: A,C,G,T, ενώ κάθε μόριο πρωτεΐνης μπορεί να θεωρηθεί ως μια ακολουθία συμβόλων (συμβολοσειρά), από ένα αλφάβητο είκοσι χαρακτήρων/ γραμμάτων, των 20 αμινοξέων.

Στις παραγράφους που ακολουθούν δίνουμε τους βασικούς ορισμούς που θα χρησιμοποιήσουμε σε επόμενα κεφάλαια.

2.1 Βασικοί Ορισμοί

Μια συμβολοσειρά. είναι μια ακολουθία 0 ή περισσότερων συμβόλων από ένα ορισμένο αλφάβητο Σ . Το σύνολο όλων των συμβολοσειρών που ορίζονται από το συγκεκριμένο αλφάβητο, συμβολίζεται με Σ+. Μια συμβολοσειρά *χ* μμήκους *n* αναπαριστάται ως η ακολουθία: x[l .. n] = x[1]x[2] ... x[n], όπου χ[i] Ε Σ για 1<=ι<=v και n =1 χ l είναι το μμήκος της συμβολοσειράς Χ.

Η κενή συμβολοσειρά. είναι η ακολουθία από μηδέν σύμβολα και συμβολίζεται ως ε οπότε Σ * = Σ +υ{ε}. Η συμβολοσειρά xy, είναι η συνένωση των συμβολοσειρών / λέξεων x και y. Η συνένωση k αντιγράφων του χ ορίζεται ως X^k και ονομάζεται *k* δύναμη του Χ.

Μια συμβολοσειρά w αποτελεί υπο-συμβολοσειρά (substring) της λέξης *χ* εάν x=uwv u,v Σ * Μια υπο-συμβολοσειρά w είναι πρόθεμα του x εάν x=wv για v Ε Σ* , ενώ ένα κανονικό πρόθεμα εάν Vε Σ* .Όμοια w είναι επίθεμα του x εάν x=vw για v Ε Σ*. Μια συμβολοσειρά v, που είναι και πρόθεμα και επίθεμα του x ονομάζεται border του x.

Εάν το Χ, έχει μη κενό border, ονομάζεται περιοδικό.Διαφορετικά, το χ ονομάζεται πρωτεύον- pnmitive. Αν συμβολίσουμε με v ένα border του Χ, μήκους l, όπου 1<=1<=n-1 ,p=n- l, καλείται περίοδος του *X*. Το p είναι περίοδος του x εάν xi=xi+p όπου 1<=i,i+p<=n. Για παράδειγμα η συμβολοσειρά Χ = abababab έχει δυο borders u_j = ab και U_2 =abab; άρα και δυο περιοδικές επαναλήψεις μήκους 4 και 2, όπου 2 είναι η περίοδος του Χ.

Μια υπο-συμβολοσειρά ονομάζεται cover μιας λέξης Χ εάν κάθε θέση του *X* βρίσκεται μέσα σε κάθε εμφάνιση του *v.* Επιπρόσθετα εάν l v 1<1 X l,

ονομάζουμε το v καυουικό cover του X. Για παράδειγμα, το *X* είναι πάντα ένα cover του X και v = aba είναι ένα κανονικό cover του X = abaababa. Οι περιοδικότητες των συμβολοσειρων παρουσιάζονται ως περιοδικές επαναλήψεις μοτίβων. Οι πιο σημαντικές περιοδικότητες που έχουν μελετηθεί στη σχετική βιβλιογραφία περιλαμβάνουν:

Την περίοδο P μιας συμβολοσειράς χ

Το cover v μιας συμβολοσειράς X, που δημιουργείται

από συνενώσεις και επικαλύψεις του χ

Το seed s μιας συμβολοσειράς X

και οι επαναλήψεις μέσα σε μια συμβολοσειρά X.

2.2 Το πρόβλημα της Ακριβούς Εύρεσης Προτύπου

Στο πρόβλημα της ακριβούς εύρεσης/ αναζήτησης ενός προτύπου σε μια μοριακή ακολουθία (συμβολοσειρά) ενδιαφερόμαστε να εντοπίσουμε όλες τις εμφανίσεις ενός δοσμένου μοτίβου X ("δομημένου" ή "μη-δομημένου") σε μια βιολογική αλληλουχία BS .

Ένα "μη-δομημένο" μοτίβο είναι μια συμβολοσειρά s μήκους, l s 1= n, όπου s[i]εΣDNA={a,c,g,t}, για κάθε 1<=i<=n, ενώ ένα "δομημένο" μοτίβο μπορεί να οριστεί ως η διατεταγμένη λίστα k "λέξεων" Bi (κάθε Bi αποτελεί μια λέξη που ορίζεται στο αλφάβητο Σ DNA) και k-1 διαστημάτων ή αλλιώς κενών (καθένα ανάμεσα σε δυο διαδοχικές λέξεις).Κάθε κενό gi,μπορεί να έχει μια ελάχιστη min και μια μέγιστη τιμή max , ή ένα καθορισμένο μήκος.

2.3 Το πρόβλημα της Προσεγγιστικής Εύρεσης Προτύπου

Στην προσεγγιστική εύρεση ενός προτύπου δεν αναζητούμε τις ακριβείς εμφανίσεις ενός προτύπου αλλά αυτές που ικανοποιούν κάποια κριτήρια ομοιότητας. Η πιο συνηθισμένη εκδοχή αυτού του προβλήματος είναι η αναζήτηση υπο-συμβολοσειρών που προσεγγίζουν το προς αναζήτηση μοτίβο, περιέχοντας το πολύ *k-δταφορές {k-differences}.* Η παράμετρος *k* ορίζει το βαθμό ομοιότητας- προσέγγισης με το μοτίβο ή με άλλα λόγια τα λάθη (αντικαταστάσεις ,διαγραφές ,ή ενθέσεις) που μπορεί να έχουν συμβεί ως αποτέλεσμα της εξελικτικής διαδικασίας. Το πρόβλημα της προσεγγιστικής εύρεσης μοτίβου έχει μελετηθεί αρκετά τα τελευταία χρόνια λόγω των ποικίλων εφαρμογών του σε ακολουθίες βιολογικών μορίων, όπως για παράδειγμα το γνωστό *k-difference global alignment problem,* στο οποίο οι βιολόγοι περιορίζουν τον αριθμό των επιτρεπόμενων διαφορών για να εντοπίσουν διαδικασίες μετάλλαξης (π.χ.: να εντοπίσουν γονίδια των οποίων

η μετάλλαξη ευθύνεται για γενετικές ασθένειες, η έχοντας ακολουθιοποιήσει μια νέα πρωτεΐνη να τη συγκρίνουν με τις ακολουθίες ήδη γνωστών).

Το πρόβλημα της προσεγγιστικής εύρεσης ενός μοτίβου μπορεί να οριστεί ως:

Ορισμός -1: Για ένα κείμενο *T,* ένα μοτίβο P, μια παράμετρο k και μια συνάρτηση ομοιότητας d(), εντοπίστε τις θέσεις στο κείμενο j, έτσι ώστε υπάρχει i ώστε d(P,Ti...j)<=k.

Το παραπάνω πρόβλημα παρουσιάζεται σε διαφορετικές εκδοχές, ανάλογα με την μετρική απόστασης ή συνάρτηση ομοιότητας (π.χ. Hamming distance, κτλ) που υιοθετούμε. Μια ενδιαφέρουσα εναλλαγή του προβλήματος εμφανίζεται όταν επιτρέπουμε την παρουσία κενών.

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. Στο ακόλουθο σχήμα παρουσιάζεται η σύγκριση δύο ακολουθιών με διαφορετικούς πίνακες μετρικών ομοιότητας.

(A)

Identities = 36/52 (69%), Positives =47/52 (90%)
query: 214 KMGPGFTKALGHGVDLGHIYGDNLERQYQLRLFKDGKLKYQVLDGEMYPPSV 265
GP+FTK+ HGVDL+HIYG++LERQ +LRLFKDGK+KYQ+++GEMYPP+V
seq: 97
ERGPAFTKGKNHGVDLSHIYGESLERQHKLRLFKDGKMKYQMINGEMYPPTV 148

(B)

Identities = 36/53 (68%), Positives =47/53 (89%)
query: 214 KMGPGFTKALGHGVDLGHIYGDNLERQYQLRLFKDGKLKYQVLDGEMYPPSV 266
+ GP ΠK HGVDL HIYG++LERQ++LRLFKDGK+KYQ++GEMYPP+V+
seq: 97
ERGPAFTKGKNHGVDLSHIYGESLERQHKLRLFKDGKMKYQMINGEMYPPTV 148

Σχήμα 2. Σύγκριση ακλουθιών α) με τον πίνακα PAM 250 και β) με τον πίνακα BLOSUM 62. Οι αντικαταστάσεις συμβόλων δηλώνονται με "+".

Επίσης η σύγκριση ακολουθιών μπορεί να κατηγοριοποιηθεί σε:α) - local alignment και β) ολική ευθυγράμμιση- global alignment. Στην τοπική ευθυγράμμιση αναζητούμε περιοχές τοπικής ομοιότητας. Ο πρώτος αλγόριθμος τοπική ευθυγράμμισης δημιουργήθηκε από τους Smith-Waterman και αρκετές σύγχρονες τεχνικές βασίζονται σε αυτόν. Αντίστοιχα ο πρώτος αλγόριθμος ολικής ευθυγράμμισης δημιουργήθηκε από τους Needleman & Wunsch. Και στις δυο περιπτώσεις υπάρχουν παραπάνω από μία δυνατές ευθυγραμμίσεις. Η βέλτιστη λύση πρέπει να ελαχιστοποιεί τις διαφορές ανάμεσα στις δυο στις ακολουθίες ή διαφορετικά να μεγιστοποιεί τη συνάρτηση ομοιότητας .

Το πρόβλημα εύρεσης τοπικών ευθυγραμμίσεων χρησιμοποιείται ευρέως στη σύγκριση μιας δοσμένης ακολουθίας μικρού μήκους (input query sequence) ως προς το σύνολο γνωστών ακολουθιών που αποθηκεύονται σε μια βάση δεδομένων. Τα προγράμματα που χρησιμοποιούνται πιο συχνά είναι τα BLAST και FASTA που αναζητούν περιοχές τοπικής ομοιότητας, που συνεισφέρουν στην ολική ευθυγράμμιση δυο ακολουθιών. Τα προγράμματα BLAST και FASTA θα αναφερθούν ονομαστικά παρακάτω ενώ θα αναλυθούν σε επόμενο κεφάλαιο.

[5]FASTA είναι μια DNA και πρωτεϊνών ευθυγράμμιση ακολουθίας πρώτο πακέτο λογισμικού που περιγράφεται (όπως FASTP) από τον David J. Lipman και William R. Pearson το 1985. Το αρχικό πρόγραμμα FASTP σχεδιάστηκε για ομοιότητα ακολουθία πρωτεϊνών αναζήτηση. FASTA προστίθεται η ικανότητα να κάνει DNA: DNA αναζητήσεων, που μεταφράζεται σε πρωτεΐνες: DNA αναζητήσεις, και προέβλεπε επίσης ένα πιο εξελιγμένο πρόγραμμα ανακατέματος για την αξιολόγηση της στατιστικής σημαντικότητας. Υπάρχουν πολλά προγράμματα σε αυτό το πακέτο που θα επιτρέπουν την ευθυγράμμιση των ακολουθιών πρωτεϊνών και αλληλουχίες του DNA . FASTA προφέρεται "FAST-Aye", και σημαίνει "FAST-All", επειδή λειτουργεί με κάθε αλφάβητο, την επέκταση του "FAST-P" (πρωτεΐνη) και "FAST-N" (νουκλεοτιδίων) ευθυγράμμιση.

[6]BLAST, είναι ένας αλγόριθμος για τη σύγκριση πρωτογενείς βιολογικές πληροφορίες για την αλληλουχία, όπως τα αμινοξέα ακολουθίες διαφορετικές πρωτεΐνες ή τα νουκλεοτίδια των αλληλουχιών του DNA. Μια αναζήτηση BLAST επιτρέπει σε έναν ερευνητή να συγκρίνει ένα ερώτημα ακολουθία με μια βιβλιοθήκη ή βάση δεδομένων των ακολουθιών, και να εντοπίσει ακολουθίες βιβλιοθήκη που μοιάζουν με το ερώτημα ακολουθία πάνω από ένα ορισμένο όριο. Για παράδειγμα, μετά την ανακάλυψη ενός γονιδίου προηγουμένως άγνωστη στο ποντίκι, ένας επιστήμονας θα εκτελεί συνήθως μια αναζήτηση BLAST του ανθρώπινου γονιδιώματος για να δούμε αν οι άνθρωποι φέρουν ένα παρόμοιο γονίδιο? BLAST θα εντοπίσει ακολουθίες του ανθρώπινου γονιδιώματος που μοιάζουν με το γονίδιο του ποντικιού που βασίζονται την ομοιότητα της ακολουθίας. Το πρόγραμμα BLAST σχεδιάστηκε από τον Eugene Myers, Stephen Altschul, Warren Gish, David J. Lipman και Webb Miller στο NIH και δημοσιεύθηκε στην J. Mol. Biol. το 1990.

ΚΕΦΑΛΑΙΟ 3^{Ο}:Ανάλυση του Αλγορίθμου Boyer-Moore

Σε αυτό το κεφάλαιο θα αναφερθούμε σε έναν από τους 3 σημαντικότερους αλγορίθμους ακριβούς εύρεσης προτύπου – exact pattern matching, που χρησιμοποιούνται σε προγράμματα σύγκρισης και ανάλυσης ακολουθιών βιολογικών δεδομένων, στον αλγόριθμο Boyer Moore ο οποίος αλγόριθμος στηρίζεται σε έναν άλλον αλγόριθμο, τον Horspool τον οποίον θα αναλύσουμε εν συντομία.

3.1 ΑΛΓΟΡΙΘΜΟΣ HORSPOOL

[12]Στην επιστήμη των υπολογιστών, η Boyer-Moore-Horspool αλγόριθμος ή αλγόριθμος Horspool είναι ένας αλγόριθμος για την εύρεση substrings σε χορδές. Η δημοσίευση έγινε από τον Nigel Horspool το 1980.

Πρόκειται για μια απλοποίηση των Boyer-Moore αλγόριθμος που σχετίζεται με την Knuth-Morris-Pratt αλγόριθμο. Ο αλγόριθμος χώρος συναλλαγών για το χρόνο, προκειμένου να αποκτήσουν ένα μέσο-πολυπλοκότητα περίπτωση του O (N) σε ένα τυχαίο κείμενο, αν και έχει O (MN), στην χειρότερη περίπτωση. Το μήκος του προτύπου είναι M και το μήκος της συμβολοσειράς αναζήτησης N.

[11]Ο αλγόριθμος με την καλύτερη επίδοση με μεγάλα strings βελόνα, όταν πέφτει με συνέπεια μια μη-που ταιριάζουν στο χαρακτήρα ή κοντά στην τελική byte από την τρέχουσα θέση του άχυρα και το τελικό byte της βελόνας δεν συμβαίνει σε άλλα τελωνεία της βελόνας. Για παράδειγμα, ένα byte 32 βελόνα που λήγει σε "z" αναζήτηση μέσω ενός byte 255 άχυρα που δεν έχει «z», σε byte που θα διαρκέσει έως και 224 byte συγκρίσεις.

Η καλύτερη περίπτωση είναι η ίδια όπως και για την Boyer-Moore αλγόριθμος σε μεγάλο σημειογραφία O, αν και η συνεχής γενικά της προετοιμασίας και για κάθε βρόχου είναι μικρότερη. Η συμπεριφορά χειρότερη περίπτωση συμβαίνει όταν το κακό χαρακτήρα παρακάμψετε είναι συνεχώς χαμηλή (με το χαμηλότερο όριο του 1 byte κίνηση) και ένα μεγάλο μέρος της βελόνας ταιριάζει με άχυρα. Ο κακός χαρακτήρας παρακάμψετε είναι μόνο χαμηλά, σε μια μερική αγώνα, όταν ο τελικός χαρακτήρας της βελόνας εμφανίζεται επίσης σε άλλο σημείο της βελόνας. Με 1 byte κυκλοφορίας συμβαίνει όταν το ίδιο byte είναι και στις δύο από τις δύο τελευταίες θέσεις.

Το κανονικό εκφυλίζονται περίπτωση παρόμοια με την παραπάνω «καλύτερη» περίπτωση είναι μια βελόνα ενός «byte ένα» ακολουθούμενο από z «31» bytes στα άχυρα που αποτελείται από z «255» bytes. Αυτό θα κάνουν επιτυχημένη 31 byte συγκρίσεις, ένα 1 byte σύγκριση που αποτύχει και στη συνέχεια να προχωρήσουμε 1 byteΑυτή η διαδικασία θα επαναληφθεί 223 φορές (255 - 32), ανεβάζοντας το συνολικό byte συγκρίσεις σε 7.168 (32 * 224).

Η χειρότερη περίπτωση είναι σημαντικά υψηλότερο από ό, τι για την Boyer-Moore αλγόριθμος, αν και προφανώς αυτό είναι δύσκολο να επιτευχθεί σε συνήθεις περιπτώσεις χρήσης. Αξίζει επίσης να σημειωθεί ότι αυτή η χειρότερη περίπτωση είναι και η χειρότερη περίπτωση για την αφελή (αλλά συνήθως) memmem () αλγόριθμος, αν και η εφαρμογή της εν λόγω τείνει να βελτιστοποιηθεί σημαντικά (και θα είναι πιο φιλική cache). [11]

3.1.1 Ο ΚΩΔΙΚΑΣ C (horspool)

```
void HORSPOOL(char *x, int m, char *y, int n) {
  int j, bmBc[ASIZE];
  char c;

  /* Preprocessing */
  preBmBc(x, m, bmBc);

  /* Searching */
  j = 0;
  while (j <= n - m) {
    c = y[j + m - 1];
    if (x[m - 1] == c && memcmp(x, y + j, m - 1) == 0)
      OUTPUT(j);
    j += bmBc[c];
  }
}
```

ΠΑΡΑΔΕΙΓΜΑ

a	A	C	G	T
$bmBc[a]$	1	6	2	8

bmBc πίνακας χρησιμοποιείται απο τον Horspool αλγόριθμο[12]

3.2 ΑΛΓΟΡΙΘΜΟΣ BOYER MOORE

[7]Ο αλγόριθμος Boyer-Moore αναπτύχθηκε από τον Bob Boyer και Strother Moore το 1977 και έχει ως κύρια χαρακτηριστικά τα εξής:

εκτελεί τις συγκρίσεις από δεξιά προς τα αριστερά

προεπεξεργασία φάση σε *O (m +σ)* Χρόνο και την πολυπλοκότητα χώρο

αναζήτηση φάση *O (n m)* χρόνο πολυπλοκότητα

3 συγκρίσεις χαρακτήρα *v* κείμενο στη χειρότερη περίπτωση για την αναζήτηση ενός μη περιοδικού προτύπου

O (n / m) καλύτερη απόδοση

Ο αλγόριθμος των Boyer και Moore συγκρίνει το σχέδιο με το κείμενο από δεξιά προς τα αριστερά. Αν το σύμβολο κείμενο που έχει σχέση με την άκρα δεξιά σύμβολο πρότυπο δεν υπάρχει κατά το πρότυπο σε όλα, τότε το μοντέλο μπορεί να μετατοπιστεί από τις θέσεις *μ.* πίσω από αυτό το σύμβολο κείμενο. Το ακόλουθο παράδειγμα δείχνει την κατάσταση αυτή.

ΠΑΡΑΔΕΙΓΜΑ

0	1	2	3	4	5	6	7	8	9	….
ένα	β	β	ένα	δ	ένα	β	ένα	γ	β	ένα
β	ένα	β	ένα	γ						
				β	ένα	β	ένα	γ		

Η πρώτη dc σύγκριση σε θέση 4 παράγει μια αναντιστοιχία. Το σύμβολο δ κείμενο δεν εμφανίζεται στο μοτίβο. Ως εκ τούτου, το σχέδιο δεν μπορεί να συγκριθεί με καμία από τις θέσεις 0, ..., 4, δεδομένου ότι όλα τα αντίστοιχα παράθυρα περιέχουν δ. Το μοντέλο μπορεί να μετατοπιστεί στη θέση 5.

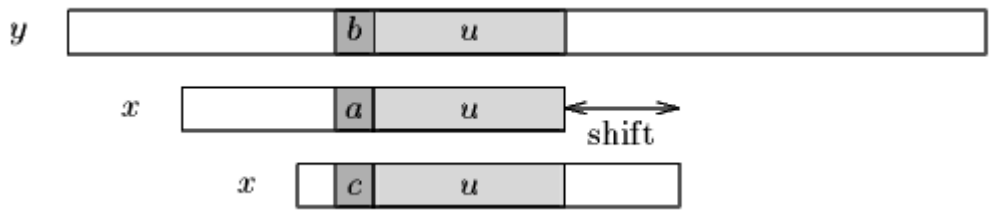
Η καλύτερη περίπτωση για τον Boyer-Moore αλγόριθμο επιτυγχάνεται, εφόσον σε κάθε προσπάθεια σε σύγκριση με το πρώτο σύμβολο κείμενο δεν υπάρχει στο μοντέλο. Συνέχεια ο αλγόριθμος απαιτεί μόνο *O (n* / *m)* συγκρίσεις.

Η Boyer-Moore αλγόριθμος θεωρείται ως η πιο αποτελεσματική string-αντιστοίχισης αλγόριθμος σε συνήθεις εφαρμογές. Μια απλοποιημένη εκδοχή του ή το σύνολο ο αλγόριθμος εφαρμόζεται συχνά σε εφαρμογές επεξεργασίας κειμένου για την "έρευνα και« υποκατάστατο »εντολές.

Ο αλγόριθμος χρησιμοποιεί τις ακόλουθες τεχνικές: α)ελέγχει τους χαρακτήρες από δεξιά προς τα αριστερά, και β) στην περίπτωση μη ταιριάσματος χρησιμοποιεί τους κανόνες *good suffix shift* και *bad character shift* για να μετατοπιστεί το σημείο στοίχισης του προτύπου ως προς την ακολουθία.

Ας υποθέσουμε ότι γίνεται μια αναντιστοιχία μεταξύ των χαρακτήρων *x [i] = α* του προτύπου και του χαρακτήρα *y [i + j] = β* του κειμένου Στη συνέχεια, από την σύγκριση αυτή ισχύει ότι :
x [i +1 .. *m* -1] = *y [i + j* +1 .. *j* + *m* -1] = u ,ενώ *x [i] =/* **y[*i + j*] .**

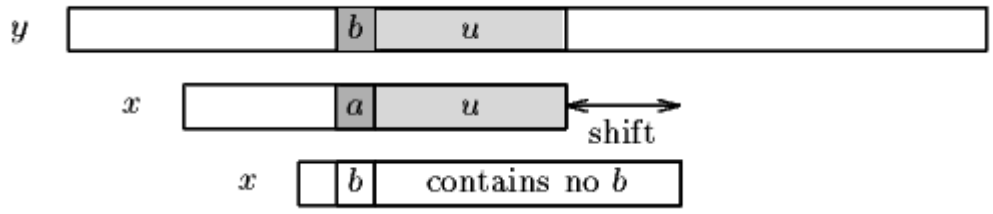
Ο κανόνας good suffix shift στηρίζεται στη μετατόπιση του τμήματος x[i+1...m-1]=y[i+j+1...j+m-1] στην επόμενη προς τα δεξιά εμφάνιση του στο κείμενο όπου ο επόμενος προς τα αριστερά χαρακτήρας είναι διάφορος του x[i].(σχήμα 1)



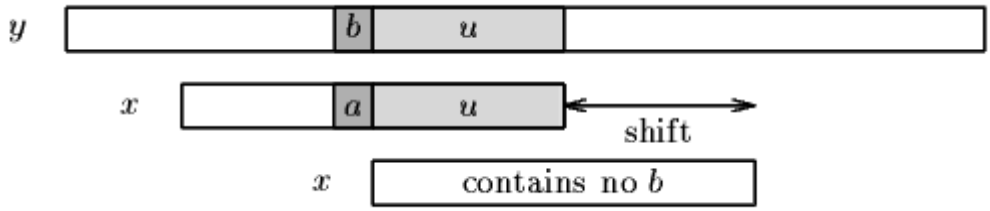
Σχήμα 1

Στην περίπτωση που δεν υπάρχει τέτοια εμφάνιση στοιχίζουμε το μέγιστο πρόθεμα v του x που συμφωνεί με το αντίστοιχο τμήμα του y[i+j+1...j+m-1].

Ο κανόνας bad character shift στηρίζεται στη στοίχιση του χαρακτήρα y[i+j] του κειμένου με τη δεξιότερη εμφάνιση του στο πρότυπο x[0...m-2] (σχήμα 2).Αν ο χαρακτήρας δεν εμφανίζεται μέσα στο πρότυπο, τότε καμία εμφάνιση του x δε θα συμπεριλαμβάνει το χαρακτήρα y[i+j], και το αριστερό τμήμα του προτύπου στοιχίζετε με τον επόμενο χαρακτήρα y[i+j+1] όπως φαίνεται στο σχήμα 3.



Σχήμα 2



Σχήμα 3

Η συνάρτηση good suffix shift, αποθηκεύεται σε έναν πίνακα bmGs μεγέθους m+1.Οι τιμές του πίνακα εκφράζουν την περίοδο του x, ως εξής :

Cs(i,s)= για κάθε k, i<k<m και s>=k, ή x[k-s]=x[k],

και

Co(i,s)=εάν s<i, τότε x[i-s]=/x[i]

Η συνάρτηση bad character shift αποθηκεύεται σε έναν πίνακα bmBc.Η τιμή της συνάρτησης ορίζεται για κάθε χαρακτήρα σ:

BmBc[σ]={ min,i:1<=i<=m-1 & x[m-1-i]=c
m,

3.2.1 Ο Κώδικας C

```
void preBmBc (char * x, int m, int bmBc [])
{
   int i;

   for (i = 0; i < ASIZE; ++i)
      bmBc[i] = v;
   for (i = 0; i < m - 1; ++i)
      bmBc[x[i]] = m - i - 1;
 }



void preBmGs (char * x, int m, int bmGs []) {
   int i, j, suff[XSIZE];

   suffixes(x, m, suff);
   for (i = 0; i < m; ++i)
      bmGs[i] = m;
   j = 0;
   for (i = m - 1; i >= 0; --i)
      if (i==-1||suff[i] == i + 1)
         for (; j < m - 1 - i; ++j)
if (bmGs[j] == m)
    bmGs[j] = m - 1 - i;
for (i = 0; i <= m - 2; ++i)
      bmGs[m - 1 - suff[i]] = m - 1 - i;
}



void suffixes(char *x, int m, int *suff)
{
   int f, g, i;

   suff[m - 1] = m;
   g = m - 1;
   for (i = m - 2; i >= 0; --i) {
      if (i > g && suff[i + m - 1 - f] < i - g)
         suff[i] = suff[i + m - 1 - f];
      else {
         if (i < g)
            g = i;
         f = i;
         while (g >= 0 && x[g] == x[g + m - 1 - f])

            --g;
         suff[i] = f - g;
       }
    }
  }
```

```
void Boyer-Moore (char * x, int m, char * y, int n)
{
  int i, j, bmGs[XSIZE], bmBc[ASIZE];

  /* Preprocessing */
  preBmGs(x, m, bmGs);
  preBmBc(x, m, bmBc);

  /* Searching */
  j = 0;
  while (j <= n - m) {
    for (i = m - 1; i >= 0 && x[i] == y[i + j]; --i);
    if (i < 0) {
      OUTPUT(j);
      j += bmGs[0];
    }
    else
      j += MAX(bmGs[i], bmBc[y[i + j]] - m + 1 + i);
  }
}
```

Οι πίνακες bmBc και bmGs υπολογίζονται σε O(m+σ) χρόνο. Η πολυπλοκότητα αναζήτησης του προτύπου είναι τετραγωνική ως προς τον απαιτούμενο χρόνο, αλλά σε πραγματικές εφαρμογές και όταν το πρότυπο που αναζητάμε δεν είναι περιοδικό απαιτούνται 3n συγκρίσεις. Επίσης για μεγάλα αλφάβητα (~|pattern|) οι απαιτούμενες συγκρίσεις είναι O(n/m), χρόνος βέλτιστος για οποιοδήποτε αλγόριθμο αναζήτησης.

ΠΑΡΑΔΕΙΓΜΑ 1

Έστω το πρότυπο x=gcagagag, ορίζουμε τις 2 βοηθητικές συναρτήσεις και εφαρμόζουμε τον αλγόριθμο Boyer-Moore.

c	A	C	G	T
$bmBc[c]$	1	6	2	8

i	0	1	2	3	4	5	6	7
$x[i]$	G	C	A	G	A	G	A	G
$suff[i]$	1	0	0	2	0	4	0	8
$bmGs[i]$	7	7	7	2	7	4	7	1

```
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
g c a g a g a g
```

ΒΗΜΑ 1: Στο πρώτο βήμα συγκρίνουμε το g με το a και βλέπουμε πως δεν υπάρχει αντιστοιχία άρα μετατοπίζουμε την κωδικολέξη κατά 1 θέση (bmGs[7]=bmBc[a]-7+7)

```
g c a t c g c a g a g a g t a t a c a g t a c g
            3 2 1
  g c a g a g a g
```

ΒΗΜΑ 2: Συγκρίνω το g με το g τα οποία είναι ίδια,στην συνέχεια το a με το a που είναι ίδια και τέλος το c με το c που είναι και αυτά ίδια.Συγκρίνοντας το ακριβώς επόμενο g με το c φαίνεται πως δεν υπάρχει αντιστοιχία άρα θα έχουμε μετατόπιση της κωδικολέξης κατά 4 θέσεις (bmGs[5]=bmBc[c]-7+5) .

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

ΒΗΜΑ 3:Στο επόμενο βήμα μετά την μετατόπιση θα συκρίνουμε το g με το g που είναι ίδια, το a με το a που είναι ίδια, το g με το g, το a με το a, το g με το g, το a με το a, το c με το c, και τέλος το g με το g τα οποία όλα είναι ίδια με αποτέλεσμα να έχουμε ακριβής εύρεση του προτύπου και μετατόπιση του κατά 7 θέσεις.

```
g c a t c g c a g a g a g t a t a c a g t a c g
                                  3 2 1
                          g c a g a g a g
```

ΒΗΜΑ 4:Έπειτα μετά την μετατόπιση κατά 7 θέσεων της κωδικολέξης συγκρίνουμε το g με το g που είναι ίδια και το a με το a τα οποία και αυτά είναι ίδια αλλα κατά την σύγκριση του c με το g βλέπουμ ότι δεν υπάρχει αντιστοιχία με αποτέλεσμα να έχουμε μετατόπιση κατά 1 θέση (bmGs[5]=bmBc[c]-7+5).

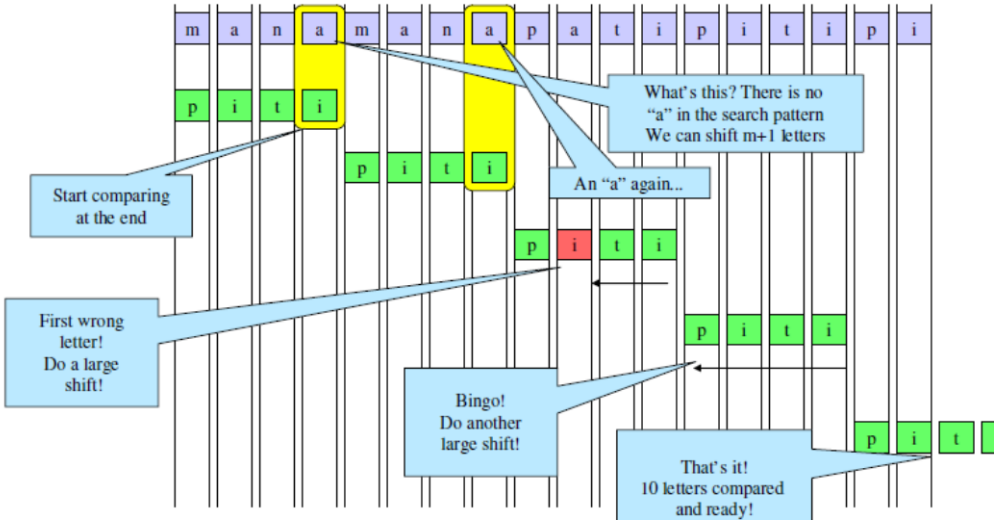
```
g c a t c g c a g a g a g t a t a c a g t a c g
                                          2 1
                                  g c a g a g a g
```

ΒΗΜΑ 5:Και τέλος μετατόπιση της κωδικολέξης κατά 7 θέσεις (bmGs[5]=bmBc[c]-7+5)

Συνολικά ο αλγόριθμος εκτελεί 17 συγκρίσεις χαρακτήρων στο συγκεκριμένο παράδειγμα.[7]

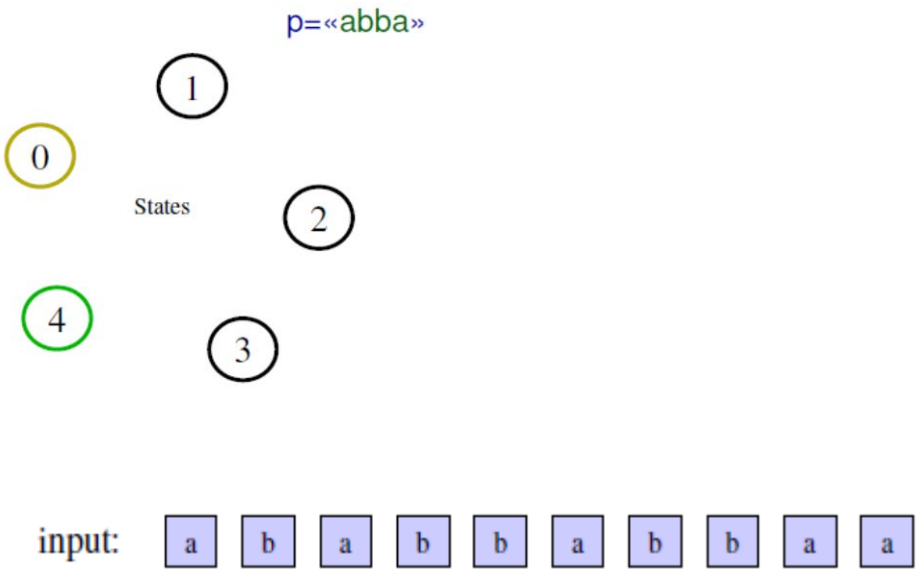
ΠΑΡΑΔΕΙΓΜΑ 2

[10]Μια γενική ιδέα για τον αλγόριθμο Boyer-Moore με ένα παράδειγμα



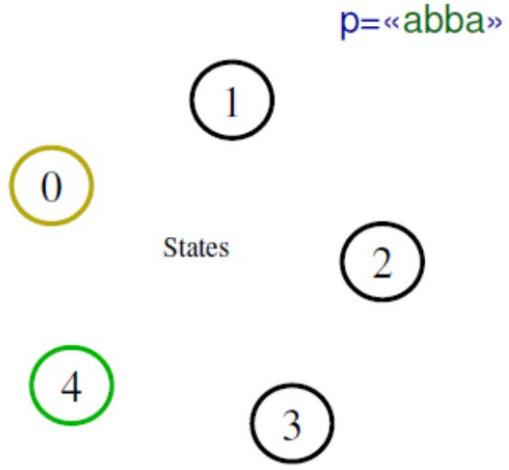
ΒΗΜΑ 1

Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας



ΒΗΜΑ 2

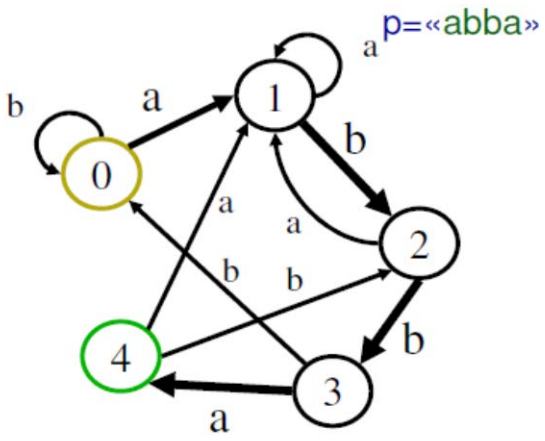
Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας



input / state	a	b
0	1	0
1	1	2
2	1	3
3	4	0
4	1	2

ΒΗΜΑ 3

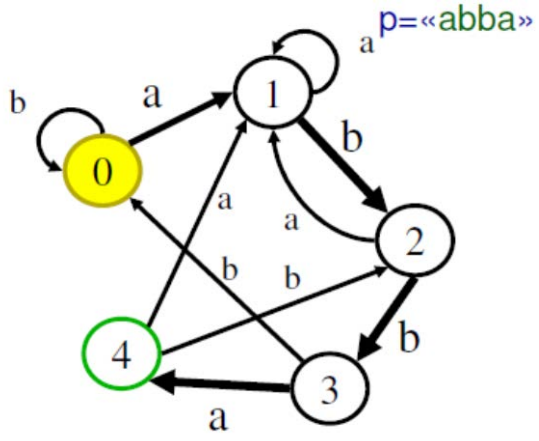
Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας



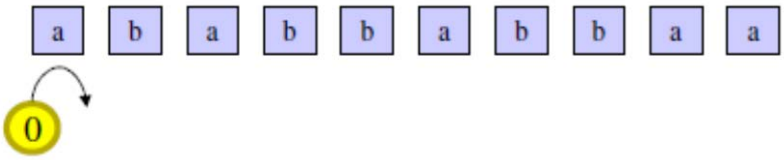
input	a	b
state 0	1	0
1	1	2
2	1	3
3	4	0
4	1	2

ΒΗΜΑ 4

Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας

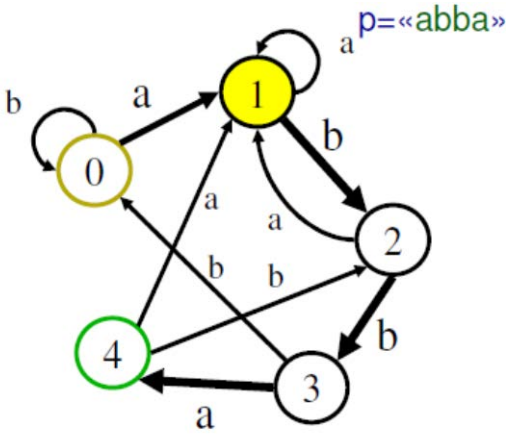


input	a	b
state 0	1	0
1	1	2
2	1	3
3	4	0
4	1	2

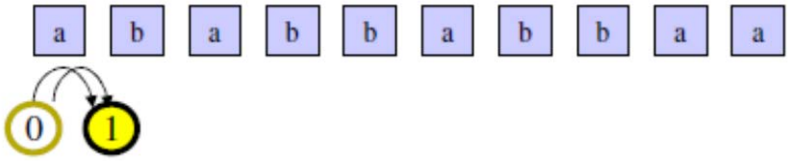


ΒΗΜΑ 5

Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας

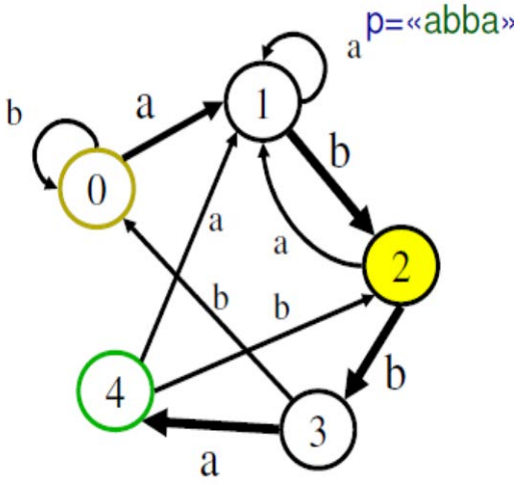


input	a	b
state 0	1	0
1	1	2
2	1	3
3	4	0
4	1	2

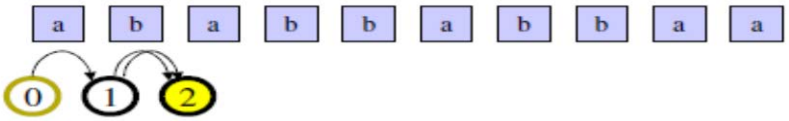


ΒΗΜΑ 6

Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας

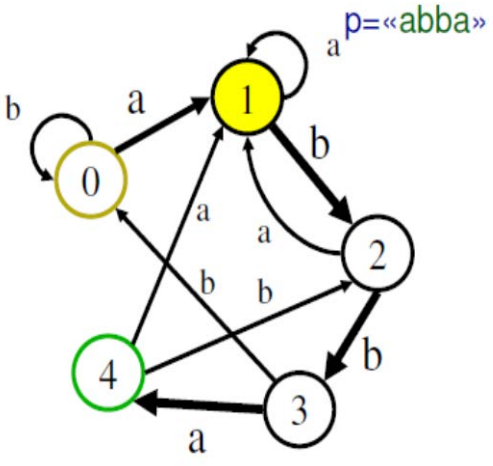


input	a	b
state 0	1	0
1	1	2
2	1	3
3	4	0
4	1	2

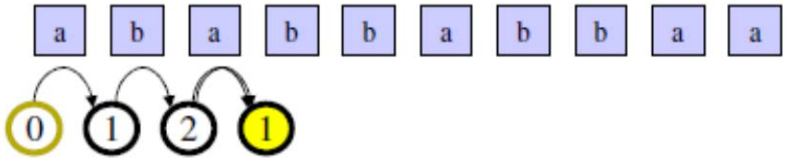


ΒΗΜΑ 7

Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας

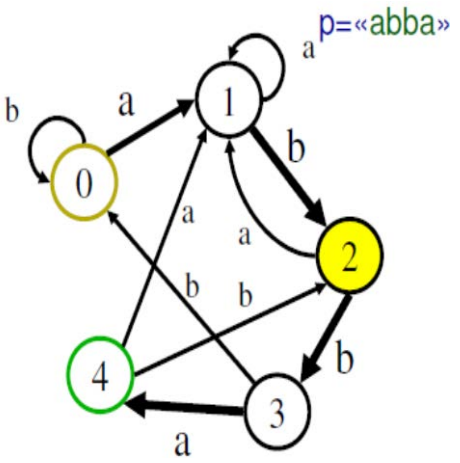


input	a	b
state 0	1	0
1	1	2
2	1	3
3	4	0
4	1	2

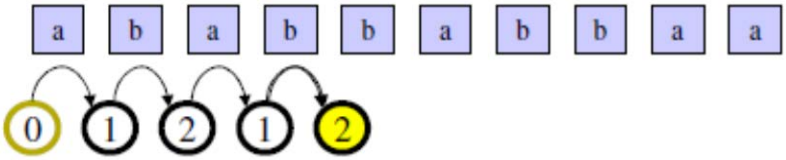


ΒΗΜΑ 8

Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας

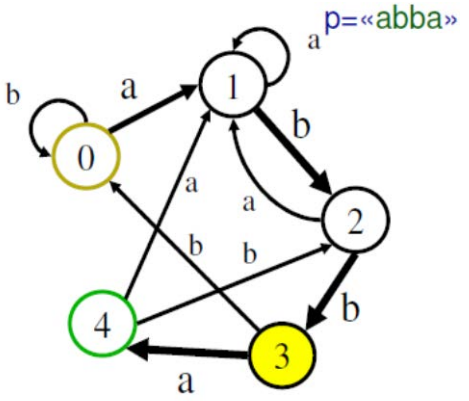


input / state	a	b
0	1	0
1	1	2
2	1	3
3	4	0
4	1	2

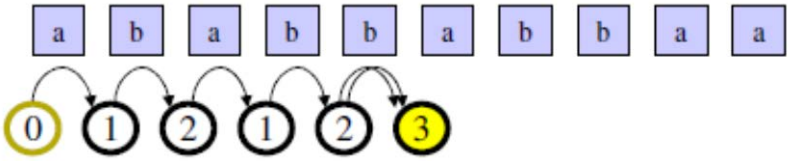


ΒΗΜΑ 9

Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας

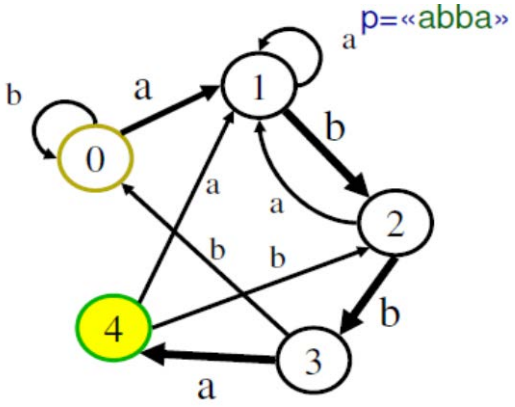


input	a	b
state 0	1	0
1	1	2
2	1	3
3	4	0
4	1	2

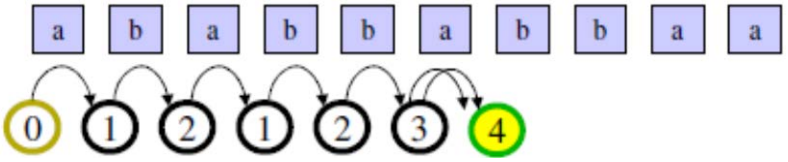


ΒΗΜΑ 10

Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας

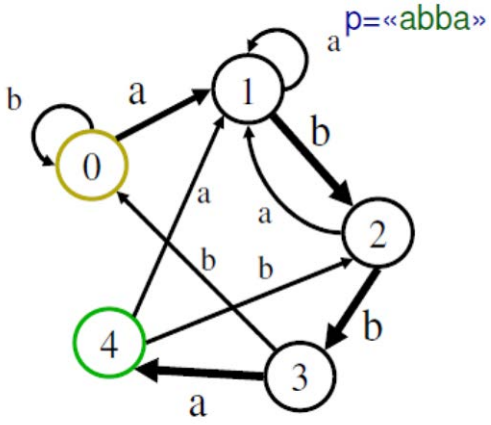


input	a	b
state 0	1	0
1	1	2
2	1	3
3	4	0
4	1	2

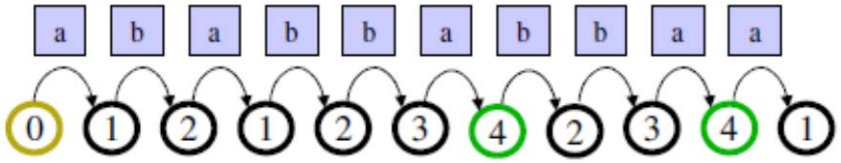


ΒΗΜΑ 11

Q είναι μια πεπερασμένη σειρά θέσεων
qo ε Q είναι η αρχική θέση
Q είναι μια σειρά θέσεων αποδοχής
Σ: εισαγωγή αλφαβήτου
δ:Q x Σ ->Q μεταβίβαση (πέρασμα) λειτουργίας



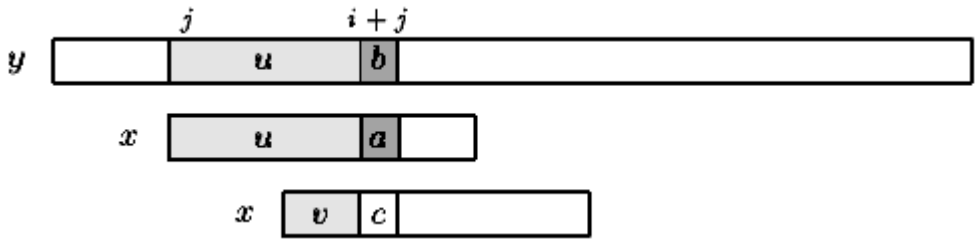
input	a	b
state 0	1	0
1	1	2
2	1	3
3	4	0
4	1	2



[10]

ΚΕΦΑΛΑΙΟ 4^{Ο}:Ανσλυση του Αλγορίθμου Knuth-Morris-Pratt

[8]Ο αλγόριθμος Knuth-Morris-Pratt σχεδιάστηκε το 1977 από τους Donald Knuth, Vaughan Pratt και JH Morris.Ο αλγόριθμος KMP συγκρίνει τους χαρακτήρες από αριστερά προς τα δεξιά χρησιμοποιώντας ένα προ-επεξεργαστικό βήμα που στοιχίζει O(m) χώρο και χρόνο ενώ η συνολική πολυπλοκότητα του αλγορίθμου είναι O(n+m) ανεξάρτητα από το συνολικό μέγεθος του αλφαβήτου των ακολουθιών. Ας δούμε λοιπόν πως λειτουργεί ο αλγόριθμος.



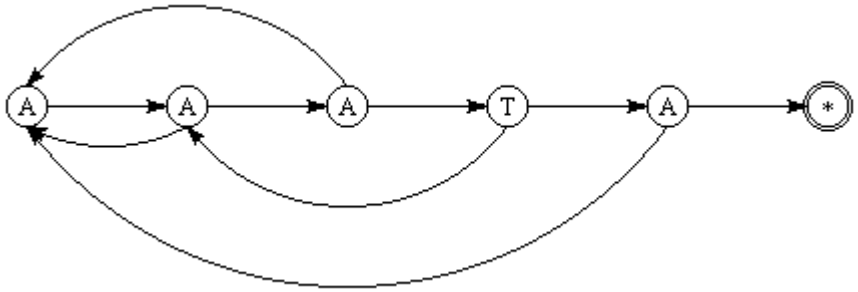
Σχήμα 1

Συγκρίνοντας το πρότυπο και την ακολουθία στους χαρακτήρες των θέσεων x[i] και y[i+j] αντίστοιχα παρατηρούμε ότι έχουμε ασυμφωνία (0<i<m). Για τις συγκρίσεις των προηγούμενων θέσεων ισχύει ότι : x[0...i-1]=y[j...i-j-1]=u, ενώ a=x[i]=/y[i+j]=/b.

Προκειμένου να μετατοπίσουμε το πρότυπο προς τα δεξιά, υποθέτουμε ότι κάποιο πρόθεμα (prefix) v του προτύπου ταιριάζει με κάποιο επίθεμα (suffix) του τμήματος u της ακολουθίας. Επίσης για να μην έχουμε ασυμφωνία και στον επόμενο χαρακτήρα, ο επόμενος χαρακτήρας μετά το πρόθεμα v πρέπει να είναι διάφορος του a.Το μέγιστο σε μήκος πρόθεμα v που ικανοποιεί αυτές τις συνθήκες ονομάζεται border του u (έχουμε ήδη ορίσει ως border το substring που εμφανίζεται στην αρχή και το τέλος μιας λέξης).

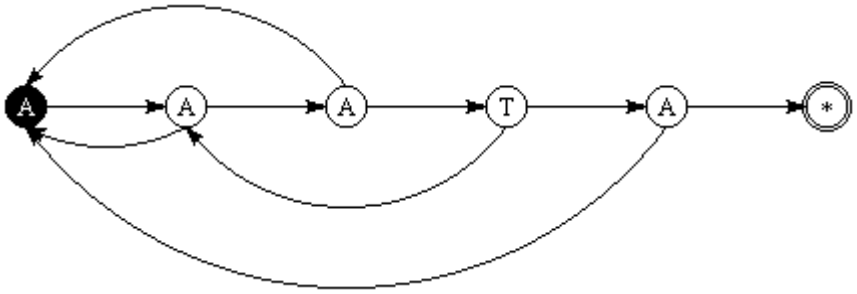
Ορίζουμε λοιπόν ως kmpNext[i] το μήκος του μέγιστου border του x[0...i-1] στο οποίο ο επόμενος χαρακτήρας c προς τα δεξιά είναι διάφορος του x[i] και στην περίπτωση που δεν υπάρχει τέτοιο border kmpNext[i]=-1, (για κάθε i, 0<i<m).Μετά τη μετατόπιση η σύγκριση των χαρακτήρων συνεχίζεται μεταξύ των x[kmpNext[i]] και y[i+j] χαρακτήρων.

Μετά από μια μετατόπιση του προτύπου,ο αλγόριθμος έχει ξεχάσει όλες τις πληροφορίες σχετικά με προηγούμενα σύμβολα. Έτσι, είναι πιθανό ότι εκ νέου συγκρίνει ένα σύμβολο-κείμενο με διαφορετικά σύμβολα μοτίβα ξανά και ξανά. Αυτό οδηγεί στην χειρότερη περίπτωση της πολυπλοκότητας του O(n,m) ***(n**:* το μήκος του κειμένου, ***m**:* μήκος του προτύπου). Παρακάτω λοιπόν παρουσιάζεται ο αλγόριθμος Knuth-Morris-Pratt ο οποίος χτίζει μια λειτουργία για την αποτυχία του προτύπου, εξετάζετε το πρότυπο p=AAATA:

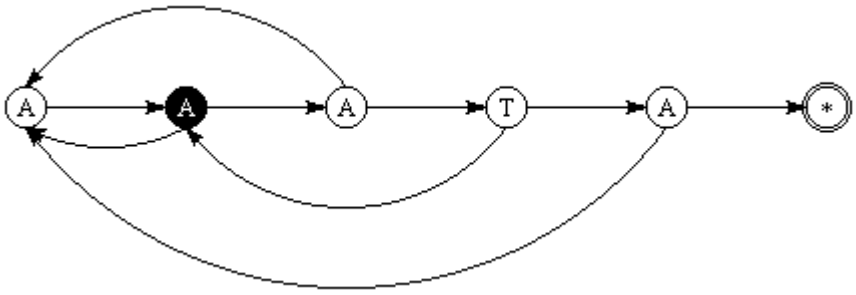


Γίνεται ξανά αναζήτηση στο κείμενο t=AATAAAATA:

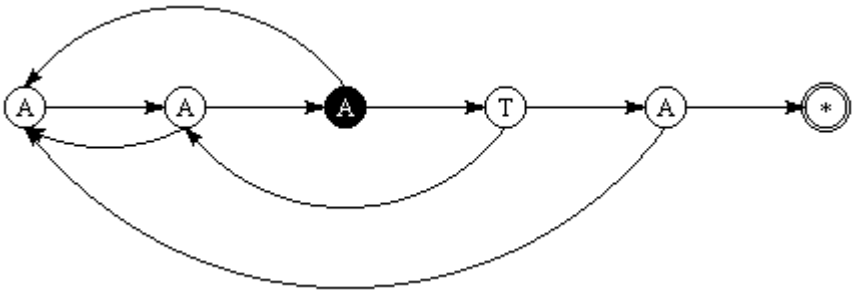
Άλφα Άλφα T Άλφα Άλφα Άλφα Άλφα T Άλφα



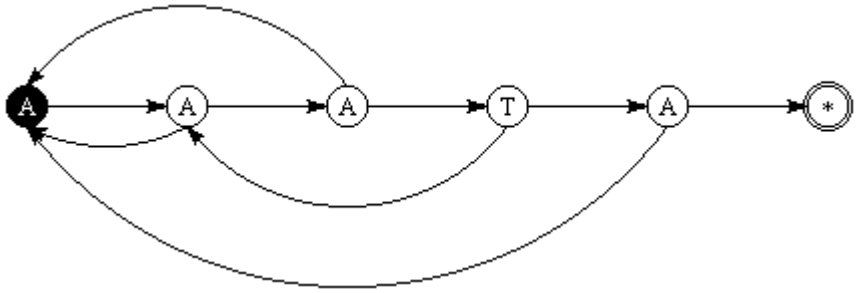
Άλφα Άλφα T Άλφα Άλφα Άλφα Άλφα T Άλφα



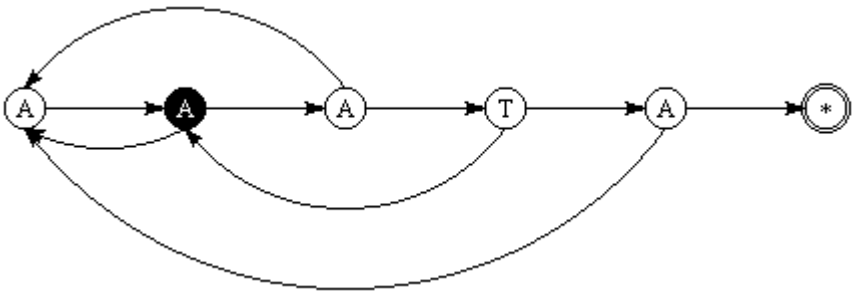
Άλφα Άλφα T Άλφα Άλφα Άλφα Άλφα T Άλφα



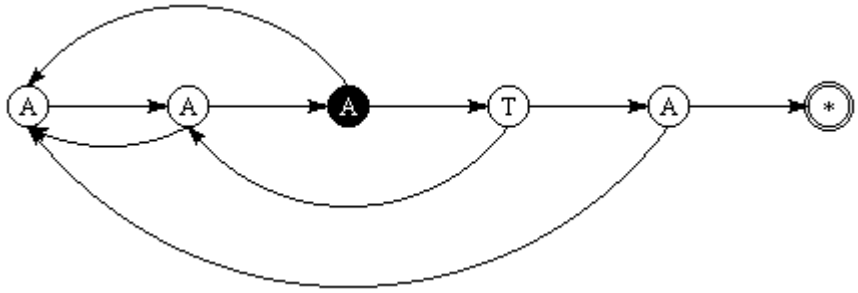
Άλφα Άλφα Τ Άλφα Άλφα Άλφα Άλφα Τ Άλφα



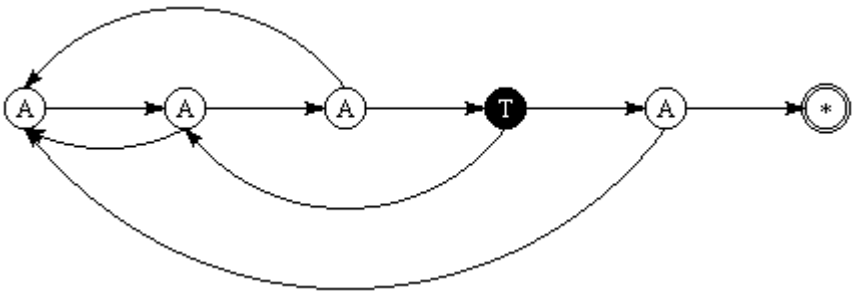
Άλφα Άλφα Τ **Άλφα** Άλφα Άλφα Άλφα Τ Άλφα



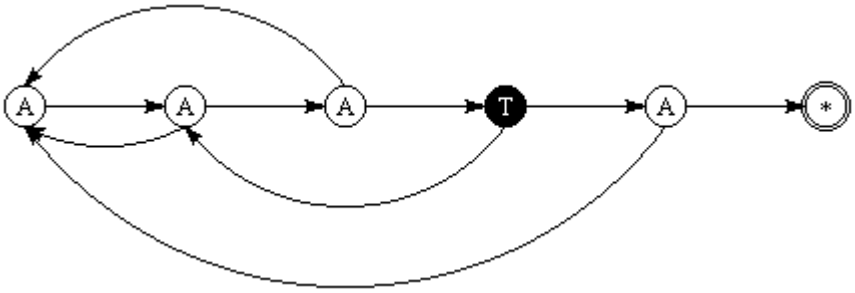
Άλφα Άλφα Τ **Άλφα Άλφα** Άλφα Άλφα Τ Άλφα



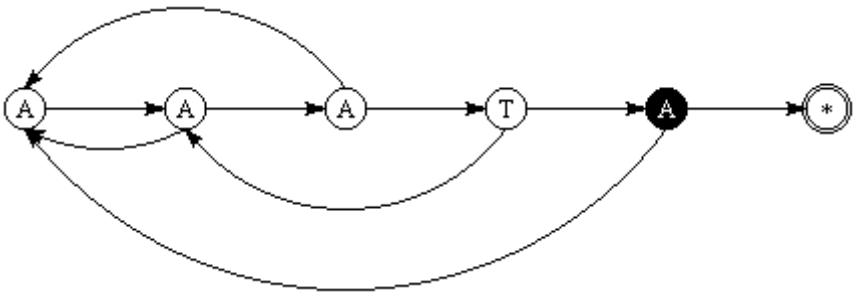
Άλφα Άλφα Τ **Άλφα Άλφα Άλφα** Άλφα Τ Άλφα



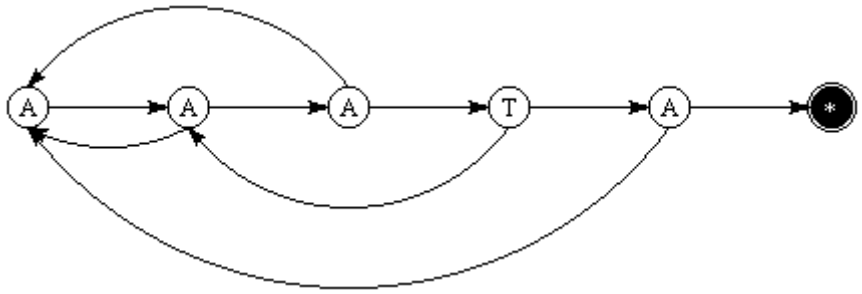
Άλφα Άλφα Τ Άλφα **Άλφα Άλφα Άλφα** Τ Άλφα



Άλφα Άλφα Τ Άλφα **Άλφα Άλφα Άλφα Τ** Άλφα



10. Άλφα Άλφα Τ Άλφα **Άλφα Άλφα Άλφα Τ Άλφα**



Ο αλγόριθμος της Knuth-Morris-Pratt [KMP]κάνει χρήση των πληροφοριών που αποκτήθηκαν από τις προηγούμενες συγκρίσεις σύμβολων. Δεν ξανά συγκρίνει ποτέ ένα σύμβολο κείμενο που έχει προσαρμόσει σε ένα σύμβολο μοτίβο. Ως αποτέλεσμα, η πολυπλοκότητα της αναζήτησης φάσης του Knuth-Morris-Pratt αλγορίθμου είναι σε ***O (n).***

Ωστόσο, η προεπεξεργασία του προτύπου είναι αναγκαία για να αναλύσει τη δομή της. Η προεπεξεργασία φάσης έχει πολυπλοκότητα *O* ***(m).*** Όσο m<=n , η συνολική πολυπλοκότητα του Knuth-Morris-Pratt αλγόριθμος είναι σε ***O (n).***

4.1 Ο Κώδικας C

```
void preKmp (char * x, int m, int kmpNext [])
{
  int i, j;

  i = 0;
  j = kmpNext[0] = -1;
  while (i < m) {
    while (j > -1 && x[i] != x[j])
      j = kmpNext[j];
    i++;
    j++;
    if (x[i] == x[j])
      kmpNext[i] = kmpNext[j];

    else
      kmpNext[i] = j;
  }
}
```

```
void KMP(char *x, int m, char *y, int n)
{
  int i, j, kmpNext[XSIZE];
```

```
  /* Preprocessing */
  preKmp(x, m, kmpNext);

  /* Searching */
  i = j = 0;
  while (j < n) {
    while (i > -1 && x[i] != y[j])
      i = kmpNext[i];
    i++; i + +?
    j++; j + +?
    if (i >= m)
      OUTPUT(j - i);
      i = kmpNext[i];
    }
  }
}
```

Πολυπλοκότητα:Οι τιμές του πίνακα kmpNext υπολογίζονται σε O(m) χώρο και χρόνο ενώ θέτουμε kmpNext[0]=-1. Η συνολική πολυπλοκότητα του αλγορίθμου είναι O(n+m) ανεξάρτητα από το μέγεθος του αλφαβήτου των ακολουθιών.

ΠΑΡΑΔΕΙΓΜΑ 1

Έστω το πρότυπο x=gcagagag, υπολογίζουμε τον πίνακα kmpNext

i	0	1	2	3	4	5	6	7	8
$x[i]$	G	C	A	G	A	G	A	G	
$kmpNext[i]$	-1	0	0	-1	1	-1	1	-1	1

```
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
g  c  a  g  a  g  a  g
```

ΒΗΜΑ 1:Στο πρώτο βήμα συγκρίνουμε το g με το g που είναι ίδια, το c με το c που είναι και αυτά ίδια και το a με το a που και αυτά είναι επίσης ίδια ενώ το ακριβώς επόμενο t δεν ίναι το ίδιο με το g με αποτέλεσμα να έχουμε μετατόπιση κατά 4 θέσεις (i-kmpNext [i]=3-(-1)=4)

```
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
        g c a g a g a g
```

ΒΗΜΑ 2:Μετά την μετατόπιση των 4 θέσεων συγκρίνουμε το g με το c όπου δεν υπάρχει αντιστοιχία άρα έχουμε μετατόπιση κατά 1 θέση (i-kmpNext [i]=0-(-1)=1)

```
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
```

ΒΗΜΑ 3:Στην συνέχεια συγκρίνουμε το g με το g, το c με το c, το a με το a, το g με το g, το a με το a, το g με το g, το a με το a και τέλος το g με το g τα οποία είναι όλα ίδια επομένως έχουμε ακριβής εύρεση του προτύπου και μετατόπιση του κατά 7 θέσεις.

```
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
                        g c a g a g a g
```

ΒΗΜΑ 4:Έπειτα από την μετατόπιση κατά 7 θέσεις συγκρίνουμε το g με το g, που είναι ίδια και το t με το c όπου δεν υπάρχει αντιστοιχία με αποτέλεσμα η κωδικολέξη να μετατοπιστεί κατά 1 θέση (i-kmpNext [i]=1-0=1)

```
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
                          g c a g a g a g
```

ΒΗΜΑ 5:Στο 5^{ο} βήμα συγκρίνουμε το g με το t όπου δεν υπάρχει αντιστοιχία άρα η κωδικολέξη μετατοπίζεται κατά 1 θέση (i-kmpNext [i]=0-(-1)=1)

```
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
                            g c a g a g a g
```

ΒΗΜΑ 6:Στο επόμανο βήμα συγκρίνουμε το a με το g όπου και εδώ δεν υπάρχει αντιστοιχία με αποτέλεσμα η κωδικολέξη να μετατοπίζεται κατά 1 θέση (i-kmpNext [i]=0-(-1)=1)

```
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
                              g c a g a g a g
```

ΒΗΜΑ 7:Μετά από την μετατόπιση κατά 1 θεση της κωδικολέξης συκρίνουμε και εδώ το a με το g όπου δεν υπάρχει αντιστοιχία άρα η κωδικολέξη μετατοπίζεται κατά 1 θέση (i-kmpNext [i]=0-(-1)=1)

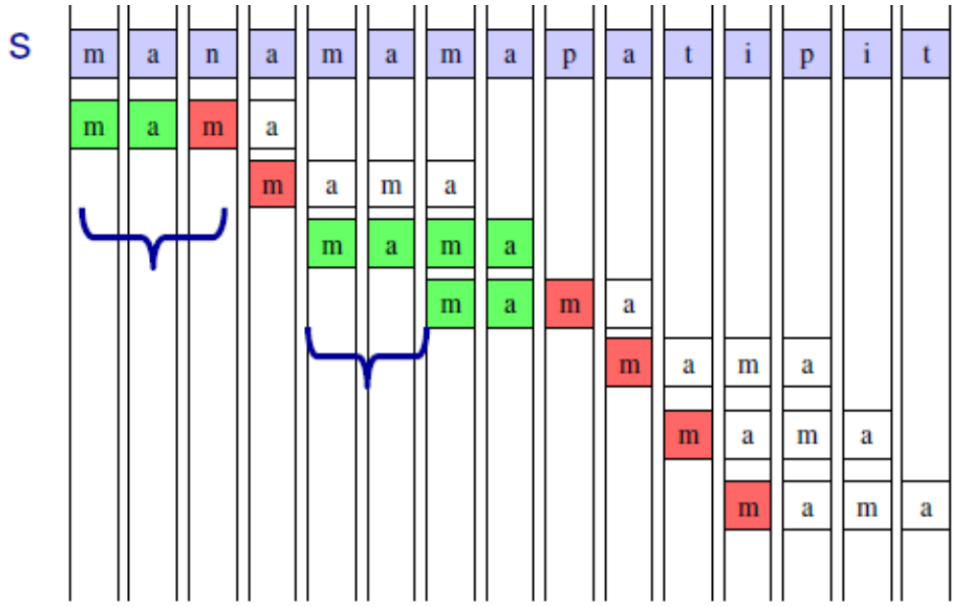
g c a t c g c a g a g a g t a t a c a g t a c g
1
g c a g a g a g

ΒΗΜΑ 8:Και τέλος συγκρίνουμε το a με το g όπου δεν υπάρχει ανιστοιχία με αποτέλεσμα η κωδικολέξη να μετατοπιστη κατά 1 θέση (i-kmpNext [i]=0-(-1)=1)

Συνολικά ο αλγόριθμος εκτελεί 18 συγκρίσεις χαρακτήρων στο συγκεκριμένο παράδειγμα.[8]

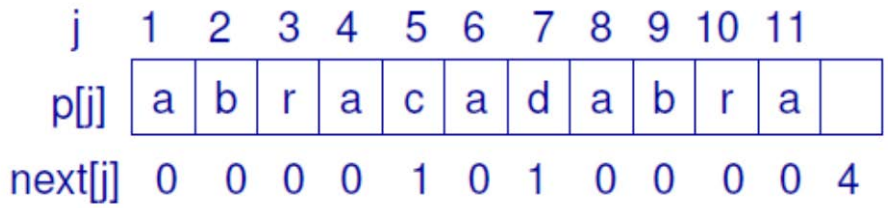
ΠΑΡΑΔΕΙΓΜΑ 2

Μία γενική ιδέα για τον αλγόριθμο knuth-morris-pratt
p=¨mama¨

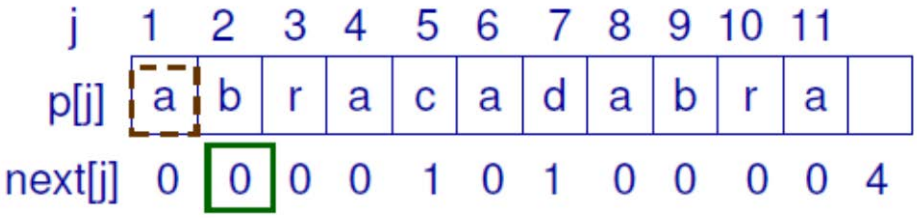


και η υλοποίηση του αλγορίθμου με ένα παράδειγμα.

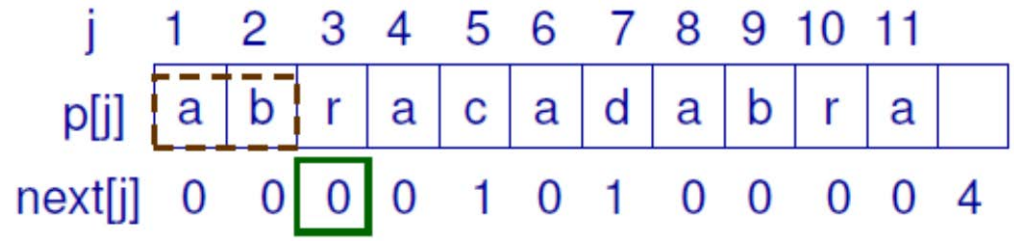
next[j]=είναι το καταλληλότερο πρόθεμα του p[1...j-1] το οποίο επίσης είναι και επίθεμα και οι χαρακτήρες που ακολουθούν το πρόθεμα και επίθεμα είναι διαφορετικοί



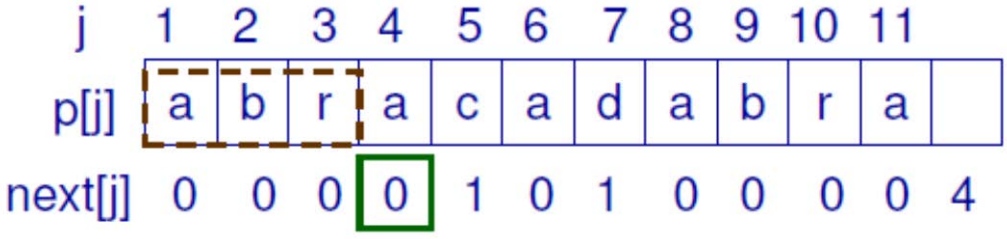
BHMA 1



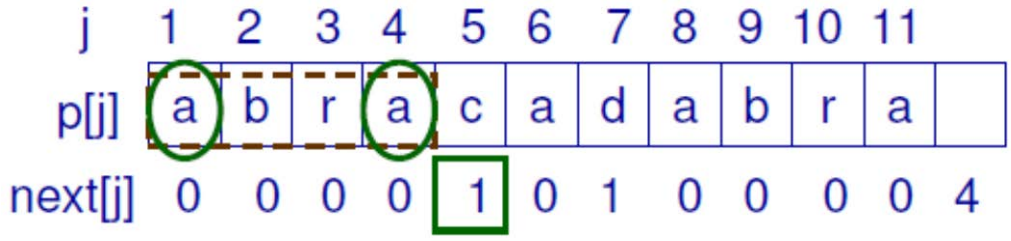
BHMA 2



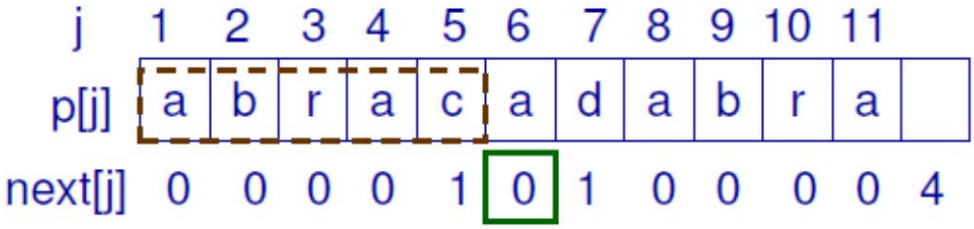
BHMA 3



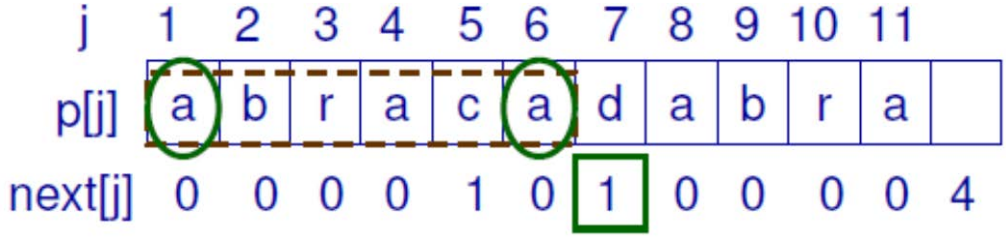
BHMA 4



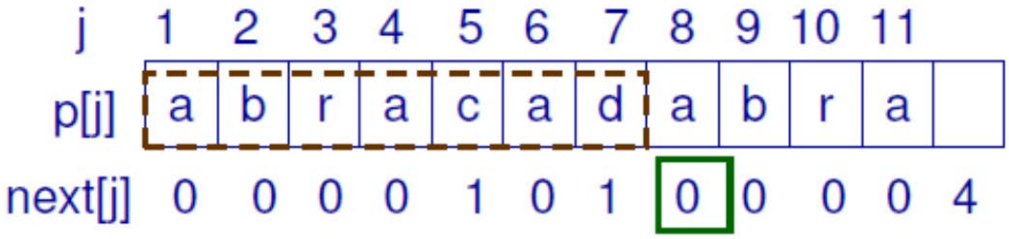
BHMA 5



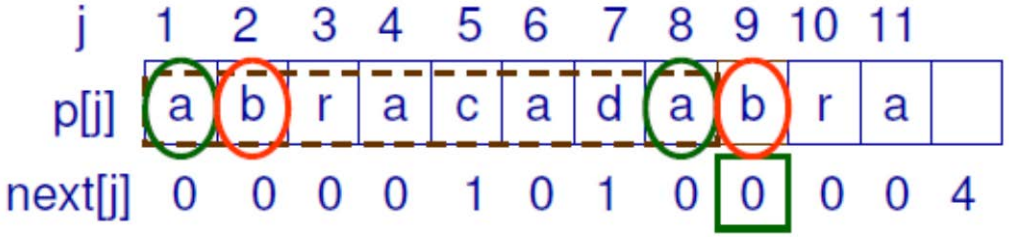
BHMA 6



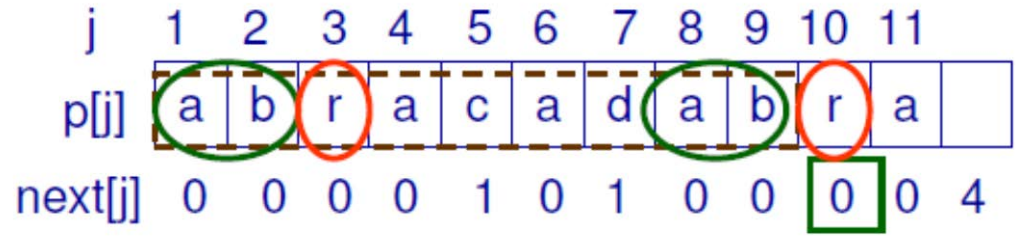
BHMA 7



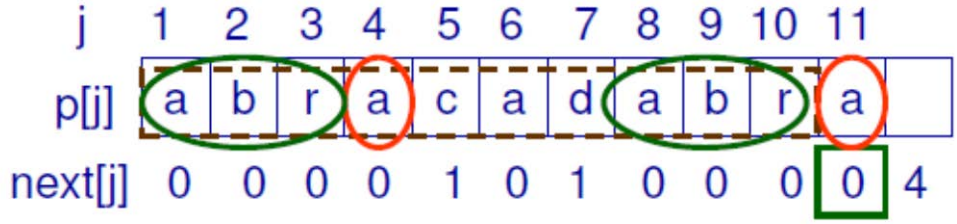
BHMA 8



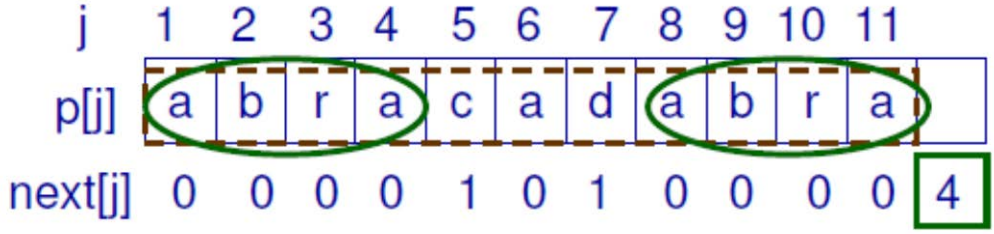
BHMA 9



BHMA 10



BHMA 11



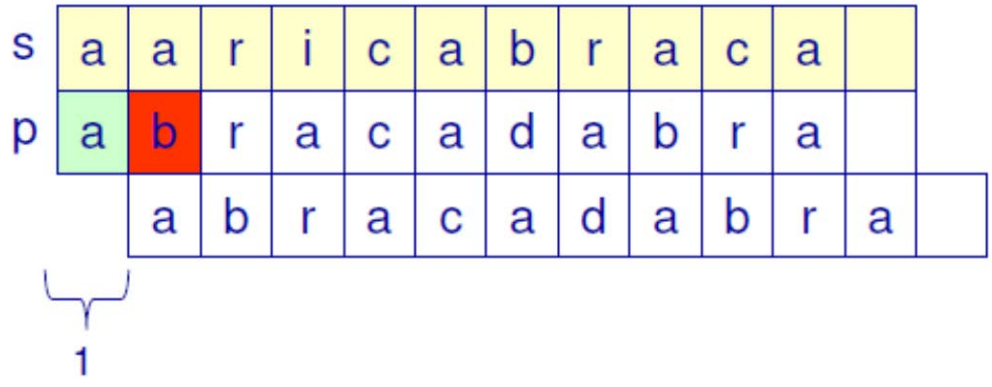
next[j]=είναι το καταλληλότερο πρόθεμα του p[1...j-1] το οποίο επίσης είναι και επίθεμα και οι χαρακτήρες που ακολουθούν το πρόθεμα και επίθεμα είναι διαφορετικοί

j	1	2	3	4	5	6	7	8	9	10	11	
p[j]	a	b	r	a	c	a	d	a	b	r	a	
next[j]	0	0	0	0	1	0	1	0	0	0	0	4
j-next[j]-1	0	1	2	3	3	5	5	7	8	9	10	7

οι j-next[j]-1 θέσεις μπορούν ασφαλή να μεταφερθούν εάν οι χαρακτήρες αντιστοιχούν με το j-1 και όχι με το j-th

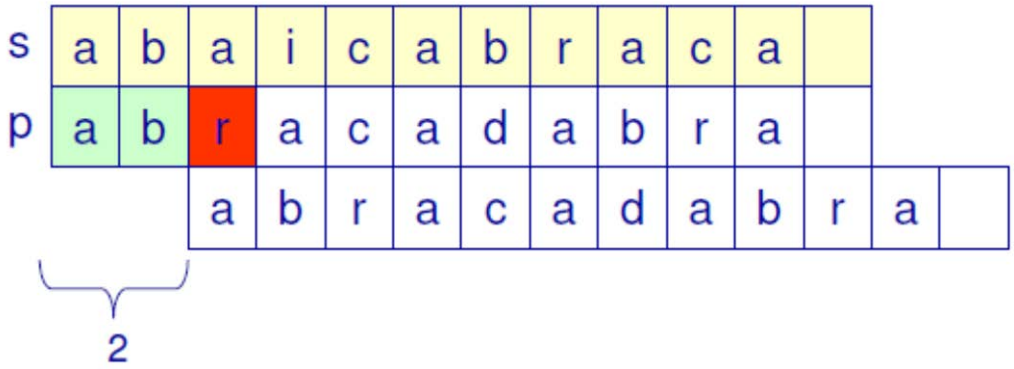
ΠΑΡΑΔΕΙΓΜΑ:Αντιστοιχία μέχρι τον 2^{ο} χαρακτήρα

j	1	2	3	4	5	6	7	8	9	10	11	
p[j]	a	b	r	a	c	a	d	a	b	r	a	
next[j]	0	0	0	0	1	0	1	0	0	0	0	4
j-next[j]-1	0	**1**	2	3	3	5	5	7	8	9	10	7



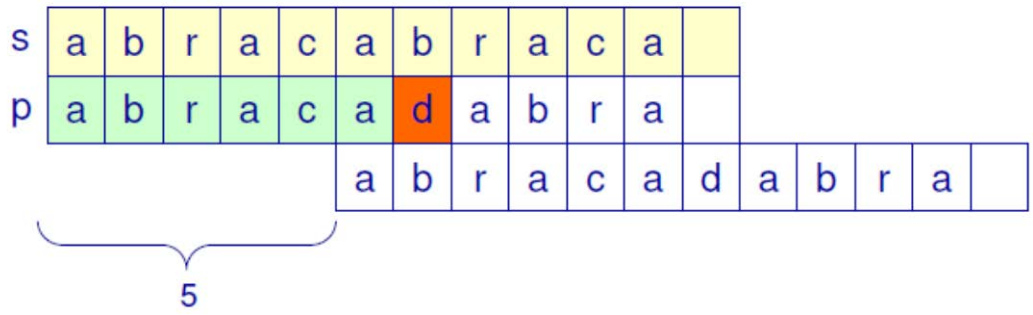
ΠΑΡΑΔΕΙΓΜΑ:Αντιστοιχία μέχρι τον 3^{ο} χαρακτήρα

j	1	2	3	4	5	6	7	8	9	10	11	
p[j]	a	b	r	a	c	a	d	a	b	r	a	
next[j]	0	0	0	0	1	0	1	0	0	0	0	4
j-next[j]-1	0	1	**2**	3	3	5	5	7	8	9	10	7



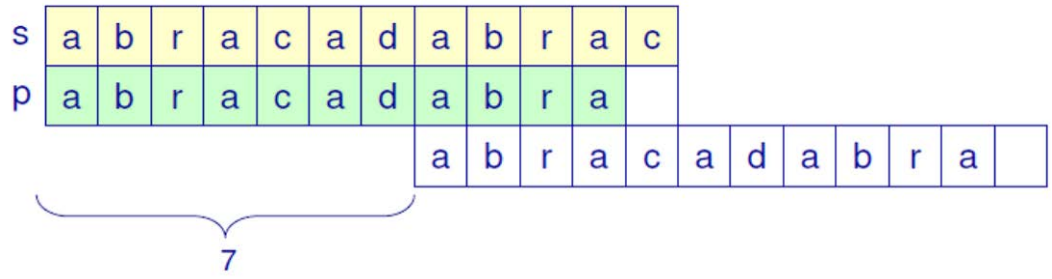
ΠΑΡΑΔΕΙΓΜΑ:Αντιστοιχία μέχρι τον 7^{ο} χαρακτήρα

j	1	2	3	4	5	6	7	8	9	10	11	
p[j]	a	b	r	a	c	a	d	a	b	r	a	
next[j]	0	0	0	0	1	0	1	0	0	0	0	4
j-next[j]-1	0	1	2	3	3	5	**5**	7	8	9	10	7



ΠΑΡΑΔΕΙΓΜΑ:Ολοκληρωτική αντιστοιχία

j	1	2	3	4	5	6	7	8	9	10	11	
p[j]	a	b	r	a	c	a	d	a	b	r	a	
next[j]	0	0	0	0	1	0	1	0	0	0	0	4
j-next[j]-1	0	1	2	3	3	5	5	7	8	9	10	7



ΚΕΦΑΛΑΙΟ 5^{Ο}:Ανάλυση του Αλγορίθμου Shift-Or

[9]Ο αλγόριθμος Shift-Or έχει κάποια χαρακτηριστικά τα οποια είναι τα ακόλουθα:

- χρησιμοποιεί αριθμητικές τεχνικές για την σύγκριση των χαρακτήρων
- ο αλγόριθμος είναι αποδοτικός όταν το μήκος του προτύπου δεν ξεπερνά το μήκος λέξης της μηχανής.
- ο αλγόριθμος χρησιμοποιεί ένα προ-επεξεργαστικό βήμα που στοιχίζει O(m+σ) χώρο και χρόνο ενώ η συνολική πολυπλοκότητα του αλγορίθμου είναι O(n) ανεξάρτητα από το μέγεθος του αλφαβήτου των ακολουθιών.
- ο αλγόριθμος προσαρμόζεται εύκολα και σε περιπτώσεις προσεγγιστικής εύρεσης προτύπου

Ο Shift-Or αλγόριθμος, χρησιμοποιεί αριθμητικές τεχνικές. Έστω R ένα bit array μεγέθους m.Το διάνυσμα Rj αποθηκεύει τις τιμές του πίνακα R, μετά την επεξεργασία του χαρακτήρα y[j] της ακολουθίας, όπως φαίνεται στο σχήμα α. Κάθε διάνυσμα περιλαμβάνει τις πληροφορίες με το ταίριασμα όλων των προθεμάτων του x που σταματούν στη θέση j(0<i<m-1):

$$Rj[i+1]=\begin{cases} 0, & x[0...i]=y[j-i...j] \\ 1 & \end{cases}$$

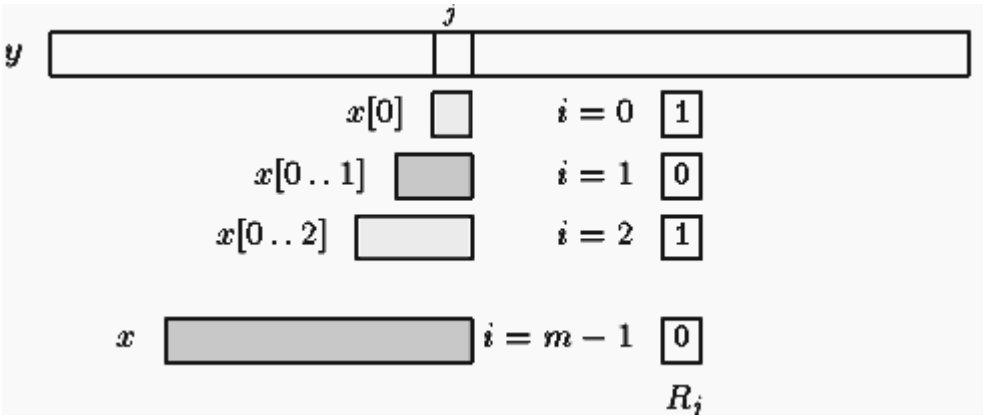
Το διάνυσμα Rj+1 υπολογίζεται χρησιμοποιώντας το διάνυσμα Rj, ως εξής :

$$Rj+1[i+1]=\begin{cases} 0, & x[i+1]=y[j+1] \\ 1 & \end{cases}$$

και

$$Rj+1[0]=\begin{cases} 0, & x[0]=y[j+1] \\ 1 & \end{cases}$$

Εάν Rj+1[m+1]=0, τότε έχουμε πλήρη εύρεση προτύπου.



Η μετάβαση από το διάνυσμα Rj στο Rj+1 μπορεί να επιταχυνθεί ως εξής:

Για κάθε χαρακτήρα cεΣ, έστω το διάνυσμα μεγέθους m Sc για κάθε 0<i<m-1, και Sc=0 αν και μόνο αν x[i]=c, το οποίο αποθηκεύει τις θέσεις εμφάνισης του χαρακτήρα c στο πρότυπο. Το διάνυσμα Sc υπολογίζεται σε ένα προ-επεγεργαστικό βήμα.

Εκτέλεση της πράξης :Rj+1=Shift(Rj) Or Sy[j+1]

5.1 Ο Κώδικας C

```
  int preScan (char *x, int m, unsigned int S[])
{
    unsigned int j, lim;
    int i;
    for (i = 0; i < ASIZE; ++i)
      S[i] = ~0;
    for (lim = i = 0, j = 1; i < m; ++i, j <<= 1) {
      S[x[i]] &= ~j;
      lim |= j;
    }
    lim = ~(lim>>1);
    return(lim);
  }

  void Shift-Or(char *x, int m, char *y, int n)
{
    unsigned int lim, state;
    unsigned int S[ASIZE];
    int j;
    if (m > WORD)
      error("Shift-Or: Use pattern size <= word size");
```

```
/* Preprocessing */
lim = preSo(x, m, S);

/* Searching */
for (state = ~0, j = 0; j < n; ++j)
  state = (state<<1) | S[y[j]];
  if (state < lim)
    OUTPUT(j - m + 1);
}
```

Πολυπλοκότητα:Ο αλγόριθμος χρησιμοποιεί ένα προ-επεξεργαστικό βήμα που στοιχίζει O(m+σ) χώρο και χρόνο ενώ η συνολική πολυπλοκότητα του αλγορίθμου είναι O(n) ανεξάρτητα από το μέγεθος του αλφαβήτου των ακολουθιών.

ΠΑΡΑΔΕΙΓΜΑ:Έστω το πρότυπο x=gcagagag υπολογίζουμε το διάνυσμα SC.

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
		G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
0	G	0	1	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	1	1	0	1	1	1	0
1	C	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	A	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	G	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	A	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	G	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
6	A	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
7	G	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1

Όπως παρατηρούμε R12 [7]=0, οπότε έχουμε μία πλήρη εμφάνιση του προτύπου στη θέση 12-8+1=5[9]

5.2Ανάλυση των αλγορίθμων και αναφορα σε συμπερασματα

Αφού περιγράψαμε τον κάθε αλγόριθμο ξεχωριστά θα κάνουμε μια γρήγορη ανασκόπηση στους 3 αλγορίθμους και θα αναφέρουμε τα συμπεράσματα που βγάλαμε.

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|= (≈|pattern|), απαιτούνται O(n/m) συγκρίσεις.

4.Χωρίς match ο χρόνος είναι O(n).

5.Υπάρχουν παραλλαγές με worst-case O(n+m) χρόνο

Ανάλυση του αλγορίθμου Knuth-Morris-Pratt σε χρόνο:

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

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

Ανάλυση του αλγορίθμου Shift-Or σε χρόνο:

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

Σε χρόνο O(m*σ) υπολογίζω, τα διανύσματα SC

ΣΥΜΠΕΡΑΣΜΑΤΑ

Δοκιμή: Τυχαία πρότυπο, τυχαίο κείμενο και αγγλική γλώσσα

Best: Η Boyer-Moore-Horspool Algorithm

Μειονέκτημα: προεπεξεργασία χώρο και το χρόνο (εξαρτάται από το αλφάβητο / μέγεθος πρότυπο)

Μικρές πρότυπο: Το Shift-Or Algorithm

Μεγάλες αλφάβητο: Η Knuth-Morris-Pratt Algorithm

Άλλα: Η Boyer-Moore Algorithm

"Δεν με νοιάζει": Το Shift-Or Algorithm

ΚΕΦΑΛΑΙΟ 6: Αλγόριθμοι Σύγκρισης Ακολουθιών Βιολογικών Δεδομένων

Στόχος της άσκησης είναι η εισαγωγή στην χρήση βιολογικών βάσεων ακολουθιών πρωτεϊνών και DNA και προγραμμάτων στοίχισης ακολουθιών χρησιμοποιώντας Web-based μέσα. Η στοίχιση δυο ακολουθιών (pairwise alignment), η πολλαπλή στοίχιση (multiple alignment) καθώς και η αναζήτηση ομόλογων πρωτεϊνών μέσα σε μια μεγάλη βάση δεδομένων, είναι θέματα που απασχολούν χρόνια τους επιστήμονες, και αποκτούν όσο περνάει ο καιρός, όλο και μεγαλύτερη σημασία, καθώς αυξάνεται ραγδαία το μέγεθος των διαθέσιμων βάσεων δεδομένων, αλλά και η ευκολία πρόσβασης σ΄αυτές για τους ερευνητές.

Θα παρουσιαστούν παρακάτω τα βασικά στοιχεία της θεωρίας για στοιχίσεις μακρομορίων (DNA και πρωτεϊνών), και οι βασικοί αλγόριθμοι που χρησιμοποιούνται τόσο για μια προς μια στοίχιση των ακολουθιών (Smith & Waterman, BLAST, FASTA) όσο και για πολλαπλή στοίχιση (CLUSTALW, CLUSTALX). Όπως είναι εμφανές, οι μέθοδοι στοίχισης είναι πολύ χρήσιμοι στον ερευνητή, για να ταυτοποιήσει μια άγνωστη πρωτεΐνη, αλλά και για να εξαγάγει πληροφορίες τόσο εξελικτικές όσο και λειτουργικές.

Το BLAST και το FASTA αποτελούν τους πιο συχνά χρησιμοποιούμενους ευριστικούς (heuristic) αλγορίθμους και βρίσκονται διαθέσιμοι στο διαδίκτυο και σε τοπικές εκδόσεις. Ο όρος ευριστικός (heuristic) χρησιμοποιείται για αλγόριθμους που βρίσκουν λύσεις μεταξύ όλων των πιθανών, αλλά δεν εγγυώνται ότι θα βρουν τη βέλτιστη. Τέτοιου είδους αλγόριθμοι, βρίσκουν μια λύση κοντινή στη βέλτιστη (close to the best one) γρήγορα και εύκολα.
Αν υποθέσουμε ότι έχουμε 2 ακολουθίες πρωτεϊνών μήκους *η*.

$$\mathbf{x} = x_1, x_2, \ldots, x_n \qquad \mathbf{y} = y_1, y_2, \ldots, y_m$$

αποδεικνύεται ότι οι πιθανές στοιχίσεις των δυο αυτών ακολουθιών είναι:

$$\binom{2n}{n} = \frac{(2n!)}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{2\pi n}} \qquad \textbf{(1)}$$

Ο αριθμός αυτός είναι τεράστιος ακόμα και για μικρές ακολουθίες, και γι' αυτό το λόγο έχουν αναπτυχθεί αλγόριθμοι δυναμικού προγραμματισμού οι οποίοι βρίσκουν την καλύτερη στοίχιση των ακολουθιών στο μικρότερο δυνατό χρόνο (Smith & Waterman, 1981). Για να γίνει η στοίχιση αυτή χρησιμοποιείται ένα είδος score, το οποίο μετράει την ομοιότητα τον αμινοξέων. Στην πιο απλή περίπτωση χρησιμοποιούμε το παρακάτω score

$$s(x_i, y_i) = \begin{cases} 1, & \text{αν} \quad x_i = y_i \\ -\infty, & \text{αν} \quad x_i \neq y_i \end{cases} \qquad \textbf{(2)}$$

Αντίστοιχες συνεισφορές στο score (αρνητικές) έχουν οι προσθήκες κενών, ενώ αξίζει να αναφερθεί ότι άλλη ποινή ορίζουμε για την προσθήκη κενών (gap open penalty) και άλλη για την επέκταση τους (gap extension penalty). Συνηθίζεται πλέον, εκτός αν υπάρχουν ειδικοί λόγοι, να μην χρησιμοποιείται ο παραπάνω τρόπος υπολογισμού του score, αλλά να λαμβάνουμε υπόψη μας και τις εξελικτικές αλλά και λειτουργικές σχέσεις των αμινοξέων, και αυτό πραγματοποιείται με τη χρήση των πινάκων υποκαταστάσεως.

6.1 ΠΙΝΑΚΕΣ ΥΠΟΚΑΤΑΣΤΑΣΗΣ ΚΑΙ ΠΟΙΝΕΣ ΓΙΑ ΤΑ ΚΕΝΑ

Μια από τις πιο σημαντικές παραμέτρους που πρέπει να λαμβάνουμε υπ' όψη μας στη χρήση των προγραμμάτων τοπικής στοίχισης, είναι το είδος του πίνακα υποκατάστασης (substitution matrix) που χρησιμοποιούμε και το ύψος της ποινής για τα κενά που ενδέχεται να εισαχθούν (gap penalties), για να γίνει η στοίχιση.

Ανεξαρτήτως του αλγορίθμου που χρησιμοποιούμε, σε γενικές γραμμές, το μέγεθος το οποίο καθορίζει τη στοίχιση είναι το score, το οποίο παίρνει κάποια τιμή θετική ή αρνητική για κάθε αντιστοίχιση των αμινοξέων, από κάποιο 20Χ20 πίνακα υποκαταστάσεως. Οι πίνακες αυτοί [PAM (Point Accepted Mutation), BLOSUM (BLOcks SUbstitution Matrix) κλπ] εκφράζουν σε γενικές γραμμές την εξελικτική σχέση των 20 αμινοξέων, π.χ. η βαλίνη και η ισολευκίνη θα έχουν σχεδόν πάντα θετική συνεισφορά στο score, αν συναντηθούν σαν ζεύγος, ενώ η αργινίνη και η φενυλαλανίνη αρνητική. Πρέπει να τονιστεί ότι ο κάθε πίνακας είναι φτιαγμένος για να πραγματοποιεί καλύτερες στοιχίσεις, σε μια δεδομένη εξελικτική σχέση των πρωτεϊνών. 'Αλλος δηλαδή πίνακας ταιριάζει στην περίπτωση που έχουμε δυο πολύ ομόλογες πρωτεΐνες και άλλος για δυο πρωτεΐνες πολύ απομακρυσμένες εξελικτικά. Σε μια αναζήτηση όμως έναντι ολόκληρης της βάσης είναι καλύτερο να χρησιμοποιούμε τους πιο αποδεκτούς πίνακες, όπως τον BLOSUM62.

Όσον αφορά στην ποινή για τα κενά, συνηθίζεται να ορίζουμε μεγαλύτερη ποινή για το πρώτο κενό (gap open penalty) και μικρότερη για τα επόμενα (gap extension penalty). Είναι προφανές ότι μικρές ή μηδενικές ποινές θα οδηγούν σε ολικές στοιχίσεις ενώ πολύ μεγάλες αρνητικές $\rightarrow -\infty$ θα οδηγούν σε πιο «αυστηρές» στοιχίσεις, χωρίς κενά. Παρακάτω δίνονται οι δυο πιο γνωστοί πίνακες υποκαταστάσεως ο BLOSUM62, και ο PAM250, όπου βλέπουμε τα παρόμοια (εξελικτικά και φυσικοχημκά) αμινοξέα να δίνουν θετικό score αν συναντηθούν ως ζεύγος ενώ τα διαφορετικά να δίνουν αρνητικό. Παρατηρείστε ότι παρ' όλο που δίνουν ανάλογα αποτελέσματα αυτά δεν είναι **εντελώς όμοια**. Η διαφορά των δυο πινάκων φαίνεται απο τα δύο μεγέθη Entropy (σχετική εντροπια - Kullback & Leibler) και απο το Expected (αναμενόμενο score ανά κατάλοιπο).

BLOSUM62

```
#  BLOSUM Clustered Scoring Matrix in 1/2 Bit Units
#  Cluster Percentage: >= 62
#  Entropy =   0.6979, Expected =  -0.5209
   A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V  B  Z  X  *
A  4 -1 -2 -2  0 -1 -1  0 -2 -1 -1 -1 -1 -2 -1  1  0 -3 -2  0 -2 -1  0 -4
R -1  5  0 -2 -3  1  0 -2  0 -3 -2  2 -1 -3 -2 -1 -1 -3 -2 -3 -1  0 -1 -4
N -2  0  6  1 -3  0  0  0  1 -3 -3  0 -2 -3 -2  1  0 -4 -2 -3  3  0 -1 -4
D -2 -2  1  6 -3  0  2 -1 -1 -3 -4 -1 -3 -3 -1  0 -1 -4 -3 -3  4  1 -1 -4
C  0 -3 -3 -3  9 -3 -4 -3 -3 -1 -1 -3 -1 -2 -3 -1 -1 -2 -2 -1 -3 -3 -2 -4
Q -1  1  0  0 -3  5  2 -2  0 -3 -2  1  0 -3 -1  0 -1 -2 -1 -2  0  3 -1 -4
E -1  0  0  2 -4  2  5 -2  0 -3 -3  1 -2 -3 -1  0 -1 -3 -2 -2  1  4 -1 -4
G  0 -2  0 -1 -3 -2 -2  6 -2 -4 -4 -2 -3 -3 -2  0 -2 -2 -3 -3 -1 -2 -1 -4
H -2  0  1 -1 -3  0  0 -2  8 -3 -3 -1 -2 -1 -2 -1 -2 -2  2 -3  0  0 -1 -4
I -1 -3 -3 -3 -1 -3 -3 -4 -3  4  2 -3  1  0 -3 -2 -1 -3 -1  3 -3 -3 -1 -4
L -1 -2 -3 -4 -1 -2 -3 -4 -3  2  4 -2  2  0 -3 -2 -1 -2 -1  1 -4 -3 -1 -4
K -1  2  0 -1 -3  1  1 -2 -1 -3 -2  5 -1 -3 -1  0 -1 -3 -2 -2  0  1 -1 -4
M -1 -1 -2 -3 -1  0 -2 -3 -2  1  2 -1  5  0 -2 -1 -1 -1 -1  1 -3 -1 -1 -4
F -2 -3 -3 -3 -2 -3 -3 -3 -1  0  0 -3  0  6 -4 -2 -2  1  3 -1 -3 -3 -1 -4
P -1 -2 -2 -1 -3 -1 -1 -2 -2 -3 -3 -1 -2 -4  7 -1 -1 -4 -3 -2 -2 -1 -2 -4
```

```
S  1 -1  1  0 -1  0  0  0 -1 -2 -2  0 -1 -2 -1  4  1 -3 -2 -2  0  0  0 -4
T  0 -1  0 -1 -1 -1 -1 -2 -2 -1 -1 -1 -1 -2 -1  1  5 -2 -2  0 -1 -1  0 -4
W -3 -3 -4 -4 -2 -2 -3 -2 -2 -3 -2 -3 -1  1 -4 -3 -2 11  2 -3 -4 -3 -2 -4
Y -2 -2 -2 -3 -2 -1 -2 -3  2 -1 -1 -2 -1  3 -3 -2 -2  2  7 -1 -3 -2 -1 -4
V  0 -3 -3 -3 -1 -2 -2 -3 -3  3  1 -2  1 -1 -2 -2  0 -3 -1  4 -3 -2 -1 -4
B -2 -1  3  4 -3  0  1 -1  0 -3 -4  0 -3 -3 -2  0 -1 -4 -3 -3  4  1 -1 -4
Z -1  0  0  1 -3  3  4 -2  0 -3 -3  1 -1 -3 -1  0 -1 -3 -2 -2  1  4 -1 -4
X  0 -1 -1 -1 -2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -2  0  0 -2 -1 -1 -1 -1 -1 -4
* -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4  1
```

PAM250

```
# This matrix was produced by "pam" Version 1.0.6 [28-Jul-93]
# PAM 250 substitution matrix, scale = ln(2)/3 = 0.231049
# Expected score = -0.844, Entropy = 0.354 bits
# Lowest score = -8, Highest score = 17
   A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V  B  Z  X  *
A  2 -2  0  0 -2  0  0  1 -1 -1 -2 -1 -1 -3  1  1  1 -6 -3  0  0  0  0 -8
R -2  6  0 -1 -4  1 -1 -3  2 -2 -3  3  0 -4  0  0 -1  2 -4 -2 -1  0 -1 -8
N  0  0  2  2 -4  1  1  0  2 -2 -3  1 -2 -3  0  1  0 -4 -2 -2  2  1  0 -8
D  0 -1  2  4 -5  2  3  1  1 -2 -4  0 -3 -6 -1  0  0 -7 -4 -2  3  3 -1 -8
C -2 -4 -4 -5 12 -5 -5 -3 -3 -2 -6 -5 -5 -4 -3  0 -2 -8  0 -2 -4 -5 -3 -8
Q  0  1  1  2 -5  4  2 -1  3 -2 -2  1 -1 -5  0 -1 -1 -5 -4 -2  1  3 -1 -8
E  0 -1  1  3 -5  2  4  0  1 -2 -3  0 -2 -5 -1  0  0 -7 -4 -2  3  3 -1 -8
G  1 -3  0  1 -3 -1  0  5 -2 -3 -4 -2 -3 -5  0  1  0 -7 -5 -1  0  0 -1 -8
H -1  2  2  1 -3  3  1 -2  6 -2 -2  0 -2 -2  0 -1 -1 -3  0 -2  1  2 -1 -8
I -1 -2 -2 -2 -2 -2 -2 -3 -2  5  2 -2  2  1 -2 -1  0 -5 -1  4 -2 -2 -1 -8
L -2 -3 -3 -4 -6 -2 -3 -4 -2  2  6 -3  4  2 -3 -3 -2 -2 -1  2 -3 -3 -1 -8
K -1  3  1  0 -5  1  0 -2  0 -2 -3  5  0 -5 -1  0  0 -3 -4 -2  1  0 -1 -8
M -1  0 -2 -3 -5 -1 -2 -3 -2  2  4  0  6  0 -2 -2 -1 -4 -2  2 -2 -2 -1 -8
F -3 -4 -3 -6 -4 -5 -5 -5 -2  1  2 -5  0  9 -5 -3 -3  0  7 -1 -4 -5 -2 -8
P  1  0  0 -1 -3  0 -1  0  0 -2 -3 -1 -2 -5  6  1  0 -6 -5 -1 -1  0 -1 -8
S  1  0  1  0  0 -1  0  1 -1 -1 -3  0 -2 -3  1  2  1 -2 -3 -1  0  0  0 -8
T  1 -1  0  0 -2 -1  0  0 -1  0 -2  0 -1 -3  0  1  3 -5 -3  0  0 -1  0 -8
W -6  2 -4 -7 -8 -5 -7 -7 -3 -5 -2 -3 -4  0 -6 -2 -5 17  0 -6 -5 -6 -4 -8
Y -3 -4 -2 -4  0 -4 -4 -5  0 -1 -1 -4 -2  7 -5 -3 -3  0 10 -2 -3 -4 -2 -8
V  0 -2 -2 -2 -2 -2 -2 -1 -2  4  2 -2  2 -1 -1 -1  0 -6 -2  4 -2 -2 -1 -8
B  0 -1  2  3 -4  1  3  0  1 -2 -3  1 -2 -4 -1  0  0 -5 -3 -2  3  2 -1 -8
Z  0  0  1  3 -5  3  3  0  2 -2 -3  0 -2 -5  0  0 -1 -6 -4 -2  2  3 -1 -8
X  0 -1  0 -1 -3 -1 -1 -1 -1 -1 -1 -1 -1 -2 -1  0  0 -4 -2 -1 -1 -1 -1 -8
* -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8  1
```

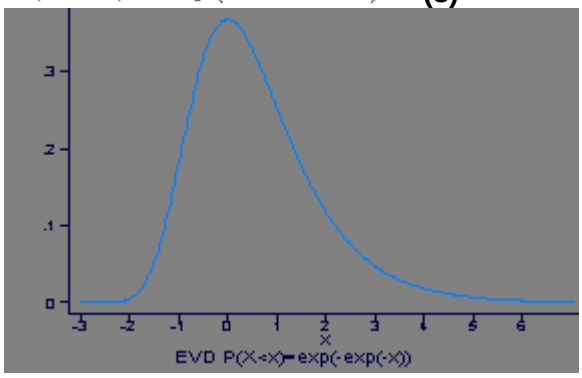
6.2 ΥΠΟΛΟΓΙΣΜΟΣ ΤΗΣ ΣΤΑΤΙΣΤΙΚΗΣ ΣΗΜΑΝΤΙΚΟΤΗΤΑΣ (SIGNIFICANCE)

'Ενα πολύ μεγάλο θέμα που προκύπτει στη στοίχιση ακολουθιών είναι η εύρεση της Στατιστικής Σημαντικότητας, των αποτελεσμάτων. Συγκεκριμμένα μας ενδιαφέρει το πως μπορούμε να διαχωρίσουμε «τυχαία» ευρήματα από «σημαντικά». Το P-value ενός στατιστικού ελέγχου (γιατι περί τέτοιου πρόκειται) είναι

η πιθανότητα, ένα αποτέλεσμα τόσο ακραίο ή και περισσότερο, να έχει προκύψει κατα τύχη δεδομένου ότι ισχύει η μηδενική υπόθεση (στην περίπτωση μας, ότι οι δύο ακολουθίες που συγκρίναμε δεν έχουν καμία σχέση μεταξύ τους). Είναι προφανές ότι αναφερόμαστε σε παραμετρικό έλεγχο και χρειάζεται να ξέρουμε την κατανομή που ακολουθεί η τυχαία μεταβλητή που μας ενδιαφέρει, το score δηλαδή.

`'Εχει αποδειχθεί (Altschul and Gish,1996) ότι η τυχαία μεταβλητή του score όταν αναφερόμαστε σε τοπικές στοιχίσεις, ακολουθεί ασυμπτωτικά την κατανομή των ακραίων τιμών (Extreme Value Distribution-EVD) η οποία έχει α.σ.κ. (αθροιστική συνάρτηση κατανομής) :

$$P\{S \le x\} = \exp\{- Kmne^{-\lambda x}\} \quad \textbf{(3)}$$



όπου m,n τα μήκη των υπό σύγκριση ακολουθιών και K, λ οι παράμετροι της κατανομής. Στην περίπτωση που συγκρίνουμε μια ακολουθία μήκους n, με μια ολόκληρη βάση δεδομένων, τότε το m είναι το συνολικό άθροισμα των αμινοξέων, που περιέχονται στις πρωτεϊνες της βάσης. Η πιθανότητα (P-value) να βρεθεί μια τιμή τοσο ακραία, η και πιο ακραία απο x θα είναι

$$P\{S \ge x\} = 1 - \exp\{- Kmne^{-\lambda x}\} \quad \textbf{(4)}$$

Η ίδια η ύπαρξη των HSP (High-scoring Segment Pair), σαν σπάνια ενδεχόμενα περιγράφεται απο την κατανομή Poisson με αναμενόμενη τιμή (E-value) να δίνεται από τη σχέση :

$$E(S \ge x) = Kmne^{-\lambda x} \quad \textbf{(5)}$$

Η τιμή αυτή συμβολίζει την αναμενόμενη (expected) τιμή για την εμφάνιση περιοχών με score τουλάχιστο ίσο με x στη βάση δεδομένων που μελετάμε. Επειδή η τιμή του score εξαρτάται αποκλειστικά απο τον πίνακα υποκατάστασης (substitution matrix) π.χ. PAM, BLOSUM κλπ, και από τις ποινές για την εμφάνιση των κενών (gap penalties) δεν μας προσφέρει χρήσιμα συμπεράσματα, καθώς δύο αποτελέσματα που έχουν προκύψει από διαφορετικό σύνολο παραμέτρων δεν είναι συγκρίσιμα. Έτσι είναι προτιμότερο να χρησιμοποιούμε μια «κανονικοποιημένη» κλίμακα ως εξής:

$$S_{bit} = \frac{\lambda S_{raw} - \ln K}{\ln 2} \quad \textbf{(6)}$$

όπου S_{raw} είναι το score που υπολογίστηκε με συγκεκριμένες παραμέτρους, και K, λ οι παράμετροι της κατανομής που είδαμε παραπάνω. Έτσι η αναμενόμενη τιμή για την εμφάνιση των HSP, είναι πλέον :

$$E(S_{bit}) = m.n.2^{-S_{bit}} \quad \textbf{(7)}$$

Πρέπει να τονιστεί εδώ ότι καθώς ισχύει η προσέγγιση :

$$1 - \exp\left(-\exp\left(-t\right)\right) \approx 1 - \left(1 - \exp\left(-t\right)\right) = \exp\left(-t\right) \quad \textbf{(8)}$$

και καθώς $P = 1 - e^{-E}$ (σχ. 3,4,5), αν το E-value είναι πολύ μικρό, τότε $P \approx E.$
'Ετσι το BLAST χρησιμοποιεί μόνο το E-value, το οποίο γίνεται πιο εύκολα κατανοητό, απο τους βιολόγους, και επι πλέον πέρνει τιμές απο 0 εώς τον αριθμό των ακολουθιών στη βάση. Όσον αφορά την εκτίμηση των παραμέτρων K και λ, το BLAST χρησιμοποιεί εκτιμητές μέγιστης πιθανοφάνειας, από μια προ-κατασκευασμένη βάση αντίστοιχης σύστασης, και για το λόγο αυτό είναι και πιο γρήγορο απο το FASTA, το οποίο χρησιμοποιεί τα αποτελέσματα από τις ίδιες τις πρωτεΐνες της βάσης στην οποία κάνει τον έλεγχο.

Αξίζει να σημειωθεί ότι στις αρχικές εκδόσεις του BLAST, δεν επιτρεπόταν στοιχιση με κενά, αλλά αυτό τροποποιήθηκε στην βελτιωμένη έκδοση. Όταν αναφερόμαστε σε τέτοιες στοιχίσεις δεν υπάρχουν μαθηματικές αποδείξεις για την κατανομή του score, αλλά όλες οι ενδείξεις συγκλίνουν στο ότι και σ' αυτή την περίπτωση ισχύουν τα παραπάνω αναφερθέντα για τη στατιστική σημαντικότητα (Arratia and Waterman, 1994; Pearson and Wood, 2001).

Κάτι που πρέπει να τονίσουμε τέλος είναι ότι το BLAST, όσον αφορά στο μήκος της βάσης δεδομένων (m) αλλα και της δοθείσας ακολουθίας (n), κάνει έναν ειδικό υπολογισμό και τα ελλατώνει κατά ένα ποσοστό, έτσι ώστε να ανταπεξέλθει στο λεγόμενο "edge effect". Το τελευταίο σημαίνει ότι καθώς μια στοίχιση προχωράει και πλησιάζει στο τέλος τη ακολουθίας, ελλατώνονται οι πιθανοί συνδιασμοί αμινοξέων που μπορούν να προστεθούν, και άρα αλλοιώνονται τα αποτελέσματα για τη στατιστική σημαντικότητα. Έτσι τα τροποποιημένα μεγέθη είναι:

$$m = m_1 - \frac{\log(Km_1n_1)}{H} \quad \text{και} \quad n = n_1 - \frac{\log(Km_1n_1)}{H} \quad \textbf{(9)}$$

όπου m_1, n_1 τα αρχικά μήκη, K η σταθερά της κατανομής EVD, και H η σχετική εντροπία του πίνακα υποκατάστασης.

Παρακάτω φαίνεται ενα δείγμα από output του BLAST για 2 ακολουθίες, όπου γραμοσκιασμένα φαίνονται τα ονόματα των ακολουθιών, το score και το E-value (expect), οι παράμετροι της κατανομής (K,λ), ο πίνακας (BLOSUM62) και οι ποινές για τα κενά. Φαίνονται επίσης τα αρχικά m_1, n_1 (length of database =279182324, length of query =367) και τα τροποποιημένα (m, n) (effective length of database=186375216, effective length of query=245), ενώ εμφανίζεται και το γινόμενο τους m*n (effective search space =45661927920)

Sequence 1 sp|P06996|OMPC_ECOLI Outer membrane protein C precursor Length367(1 .. 367)
Sequence 2 sp|P21420|NMPC_ECOLI Outer membrane porin protein NMPC Length365(1 .. 365)

NOTE: The statistics (bitscore and expect value) is calculated based on the size of nr database
Score = 444 bits (1143), Expect = e-124
Identities = 226/351 (64%), Positives = 260/351 (73%), Gaps = 20/351 (5%)

```
Query:                                                                25
YNKDGNKLDLYGKVDGLHYFSDNKDVDGDQTYMRLGFKGETQVTDQLTGYGQWEYQIQGN
84
        YNKD NKLDLYGKV+  HYFS N   DGD TY RLGFKGETQ+ DQLTG+GQWEY+ +GN
Sbjct:                                                                27
YNKDSNKLDLYGKVNAKHYFSSNDADDGDTTYARLGFKGETQINDQLTGFGQWEYEFKGN
86

Query: 85  SAENENNS--WTRVAFAGLKFQDVGSFDYGRNYGVVYDVTSWTDVLPEFGGDTY-
GSDNF 141
         AE++ +S   TR+AFAGLKF D GS DYGRNYGV YD+ +WTDVLPEFGGDT+  +D F
Sbjct:                                                                87
RAESQGSSKDKTRLAFAGLKFGDYGSIDYGRNYGVAYDIGAWTDVLPEFGGDTWTQTDVF
146

Query:                                                               142
MQQRGNGFATYRNTDFFGLVDGLNFAVQYQGKNGNPSGEGFTSGVTNNGRDALRQNGDGV
201
        M QR  G ATYRN DFFGLVDGLNFA QYQGKN     + +T G          NGDG
Sbjct:     147      MTQRATGVATYRNNDFFGLVDGLNFAAQYQGKNDRSDFDNYTEG----------
NGDGF 195

Query:                      202                      GGSITYDYEGFGIGGAISSSKRTDAQNTAAYI-----
GNGDRAETYTGGLKYDANNIYLA 256
        G S TY+YEGFGIG   + S RTD Q  A  +     +G  AE +  GLKYDANNIYLA
Sbjct:                                                               196
GFSATYEYEGFGIGATYAKSDRTDTQVNAGKVLPEVFASGKNAEVWAAGLKYDANNIYLA 255

Query:                                                               257
AQYTQTYNATRVGSLGWANKAQNFEAVAQYQFDFGLRPSLAYLQSKGKNLGRGYDDEDIL
316
         Y++T N T       ANKAQNFEAVAQYQFDFGLRPS+AYLQSKGK+LG  + D+D++
Sbjct:    256    TTYSETQNMTVFADHFVANKAQNFEAVAQYQFDFGLRPSVAYLQSKGKDLG-
VWGDQDLV 314

Query: 317 KYVDVGATYYFNKNMSTYVDYKINLLDDNQFTRDAGINTDNIVALGLVYQF 367
        KYVDVGATYYF KNMST+VDYKINLLD N FT+  G++TD+IVA+GLVYQF
Sbjct: 315 KYVDVGATYYFKKNMSTFVDYKINLLDKNDFTKALGVSTDDIVAVGLVYQF 365
```

Lambda K H

0.315 0.137 0.408

Gapped
Lambda K H
0.267 0.0410 0.140

Matrix: BLOSUM62
Gap Penalties: Existence: 11, Extension: 1
Number of Hits to DB: 2095
Number of Sequences: 0
Number of extensions: 207
Number of successful extensions: 6
Number of sequences better than 10.0: 1
Number of HSP's better than 10.0 without gapping: 1
Number of HSP's successfully gapped in prelim test: 0
Number of HSP's that attempted gapping in prelim test: 0
Number of HSP's gapped (non-prelim): 1
length of query: 367
length of database: 279,182,324
effective HSP length: 122
effective length of query: 245
effective length of database: 186,375,216
effective search space: 45661927920
effective search space used: 45661927920
T: 9
A: 40
X1: 16 (7.3 bits)
X2: 129 (49.7 bits)
X3: 129 (49.7 bits)
S1: 42 (22.0 bits)
S2: 72 (32.3 bits)

6.3 FASTA

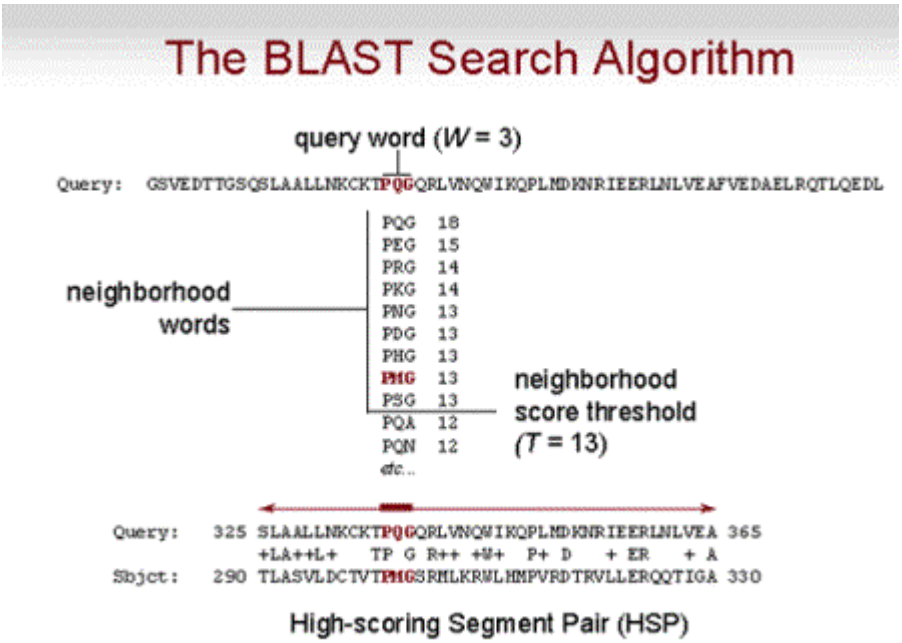
Είναι ο πρώτος ευριστικός (heuristic) αλγόριθμος ευρείας χρήσεως αναζήτησης ομοιοτήτων σε βιολογικές βάσεις. Συγκρίνει μια πρωτεϊνική ακολουθία με μια άλλη πρωτεϊνική ακολουθία ή με μια πρωτεϊνική βάση δεδομένων. Καθώς επίσης και μια DNA ακολουθία με μια άλλη ακολουθία DNA ή με μια DNA βάση.
Το πρόγραμμα **FASTA3** βρίσκεται στο site του EBI (European Bioinformatics Institute) (http://www.ebi.ac.uk/fasta33/). Ψάχνει να βρει τις βέλτιστες τοπικές στοιχίσεις από την ανίχνευση της ακολουθίας για μικρές αντιστοιχίες που ονομάζονται "λέξεις" (words). Αρχικά, υπολογίζεται το σύνολο των τμημάτων ("init1") στα οποία υπάρχουν πολλαπλές "λέξεις" (words). Τα αποτελέσματα διάφορων τμημάτων μπορούν να αθροιστούν για να παράγουν ένα "initn". Μια βελτιστοποιημένη στοίχιση που περιλαμβάνει ανοίγματα (gaps) εμφανίζεται ως "opt". Η ευαισθησία και η ταχύτητα της αναζήτησης είναι αντιστρόφως σχετιζόμενες και ελεγχόμενες από τη μεταβλητή "k-tup" που προσδιορίζει το μέγεθος μιας λέξης (word) (Pearson and Lipman, 1988). Το FASTA υπολογίζει τη στατιστική σημαντικότητα "επί τόπου" (on the fly) από το σύνολο των δεδομένων, (πιο ακριβές αλλά έχει προβλήματα αν το σύνολο των δεδομένων είναι πολύ μικρό).

6.3.1ΠΑΡΑΛΛΑΓΕΣ

fasta3, - εξερευνεί μια πρωτεϊνική ή νουκλεοτιδική (DNA) βάση ακολουθιών για παρόμοιες ακολουθίες.
fastx/y3 - συγκρίνει μια νουκλεοτιδική ακολουθία (DNA) μεταφρασμένη σε όλα τα πλαίσια ανάγνωσης (reading frames) με μια βάση πρωτεϊνικών ακολουθιών του NCBI.
tfastx/y3 - συγκρίνει μια πρωτεΐνη με μια μεταφρασμένη DNA βάση δεδομένων.
fastf3 - συγκρίνει μείγμα πεπτιδίων με μια πρωτεϊνική βάση δεδομένων.
bic_sw - συγκρίνει μια πρωτεϊνική ή νουκλεοτιδική (DNA) ακολουθία με μια βάση ακολουθιών χρησιμοποιώντας τον αλγόριθμο των Smith-Waterman. (Smith & Waterman, 1981)

6.4BLAST

(Basic Local Alignment Search Tool) (Altschul et al, 1990)
Ο αλγόριθμος **BLAST** χρησιμοποιείται για τη σύγκριση μιας ακολουθίας με μια βάση δεδομένων, έχει αρχικά αναπτυχθεί και διατηρείται από το NCBI (National Center for Biotechnology Information) (http://www.ncbi.nlm.nih.gov/BLAST/) (αλλά λόγω του ότι διατίθεται ελεύθερα υπάρχουν εκδόσεις του και σε άλλες αντίστοιχες βάσεις. Είναι ένας ευριστικός (heuristic) αλγόριθμος σύγκρισης ακολουθιών βελτιστοποιημένης ταχύτητας που χρησιμοποιείται για να ψάχνει σε βάσεις ακολουθιών την άριστη τοπική στοίχιση με μια αναζήτηση. Η αρχική αναζήτηση γίνεται για μια λέξη μήκους "W" (3 στο blastp) που δίνει αποτελέσματα για τουλάχιστον "T", όταν συγκρίνεται με την ζητούμενη ακολουθία, χρησιμοποιώντας έναν δεδομένο πίνακα υποκαταστάσεων (substitution matrix). Οι επιτυχείς λέξεις που έχουν score T ή μεγαλύτερο επεκτείνονται και προς τις δύο κατευθύνσεις σε μια απόπειρα να παραχθούν στοιχίσεις που να υπερβαίνουν το προκαθορισμένο κατώφλι (threshold) "S". Οι περιοχές που ικανοποιούν αυτή τη συνθήκη ονομάζονται HSP (High-scoring Segment Pair). Η παράμετρος "T" καθορίζει την ταχύτητα και την ευαισθησία της αναζήτησης (Εικόνα 1). Ενώ το BLAST υπολογίζει τις παραμέτρους της EVD (Extreme Value Distribution), απο τις οποίες θα υπολογίσει τη στατιστική σημαντικότητα από εξομοιώσεις που έχει πραγματοποιήσει από πριν, το FASTA τις υπολογίζει από όλες τις άλλες ακολουθίες της βάσης δεδομένων και για αυτό το λόγο είναι και πιο αργό.



Εικονα 1: Ο αλγόριθμος του BLAST

Ο αλγόριθμος BLAST είναι μια διαισθητική (ευριστική) μέθοδος αναζήτησης που αναζητάει λέξεις μήκους W (προκαθορισμένο = 3 στο blastp) που σημειώνουν score μεγαλύτερο απο ένα προκαθορισμένο όριο (Threshold-T) όταν στοιχίζονται με την δοθείσα ακολουθία (query) και με ένα δεδομένο πίνακα υποκατάστασης. Οι λέξεις στη βάση δεδομένων που σημειώνουν score επεκτείνονται και προς τις δυο κατευθύνσεις σε μία προσπάθεια να μεγιστοποιηθεί το score, σύμφωνα με κάποιο κριτήριο ευαισθησίας (επίσης προκαθορισμένο απο πριν, συνήθως 10) του E-value. Στη συγκεκριμμένη περίπτωση η τριπλέτα PQG έχει βρεθεί να στοιχίζεται στην PMG, και καθώς χρησιμοποιείται ο πίνακας BLOSUM62, το score της στοίχισης είναι 13>T (P-P =7, G-G=6, Q-M=0). Κατόπιν αυτή η τριπλέτα θα επεκταθεί και προς τις δυο κατευθύνσεις για να δώσει τη μέγιστη τοπική στοίχιση. Τα HSPs που πληρούν αυτά τα κριτήρια θα αναφερθούν από το BLAST, υπό τον όρο ότι δεν υπερβαίνουν τον αριθμό των επιτρεπτών προς εμφάνιση ακολουθιών.

Η περιοχή αναζήτησης ('Search' box) του BLAST δέχεται διαφορετικούς τύπους μορφοποιημένης εισόδου και αυτόματα καθορίζει τη μορφή. Οι τρεις μορφοποήσεις (formats) με τις οποίες μπορεί να εισαχθεί μια ακολουθία και να γίνει η αναζήτηση είναι:

FASTA – FORMAT

Η ακολουθία σε μορφή FASTA ξεκινάει με μια γραμμή περιγραφής (single-line description), και ακολουθείται από γραμμές ακολουθίας με 60 χαρακτηρες για τα αμινοξέα σε κάθε γραμμή. Η γραμμή περιγραφής ξεχωρίζει από τις γραμμές της ακολουθίας επειδή στην αρχή της γραμμής περιγραφής υπάρχει το σύμβολο (">"). Ένα παράδειγμα ακολουθίας σε μορφή FASTA είναι:
>gi|129295|sp|P01013|OVAX_CHICK GENE X PROTEIN (OVALBUMIN-RELATED)

QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQESKPVQMM
CMNNSFNVATLPA

EKMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEWTNPNTMEKRRVKVYL
PQMKIEEKYNL

TSVMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIK
HSPESEQFRAD

HPFLFLIKHNPTNTIVYFGRYWSP

Εισάγωντας μια ακολουθία σε μορφή FASTA δεν επιτρέπεται να υπάρχουν άδειες γραμμές ενδιάμεσα. Οι ακολουθίες παρουσιάζονται με τους πρότυπους αμινοξικούς και νουκλεοτιδικούς κώδικες (IUB/IUPAC), με μερικές εξαιρέσεις. Ο κώδικας γραμμάτων (letter codes) των νουκλεοτιδίων είναι:

A --> adenosine	M --> A C (amino)
C --> cytidine	S --> G C (strong)
G --> guanine	W --> A T (weak)
T --> thymidine	B --> G T C
U --> uridine	D --> G A T
R --> G A (purine)	H --> A C T
Y --> T C (pyrimidine)	V --> G C A
K --> G T (keto)	N --> A G C T (any)
- --> gap of indeterminate length	

Για τα προγράμματα που πραγματοποιούν αναζητήσεις αμινοξικών ακολουθιών ο κώδικας γραμμάτων των αμινοξέων ειναι:

A alanine	P proline
B aspartate or asparagine	Q glutamine
C cystine	R arginine
D aspartate	S serine
E glutamate	T threonine
F phenylalanine	U selenocysteine
G glycine	V valine
H histidine	W tryptophan
I isoleucine	Y tyrosine
K lysine	Z glutamate or glutamine
L leucine	X any
M methionine	* translation stop
N asparagine	- gap of indeterminate length

Σκέτη ακολουθία (Bare Sequence)

Είναι μόνο οι γραμμές της ακολουθίας χωρίς τη FASTA γραμμή ορισμού.

Προσδιοριστές (Identifiers)

Είναι ένας αριθμός πρόσβασης (accession) στο NCBI. Αυτοί οι προσδιοριστές ακολουθιών του NCBI έχουν μια συγκεκριμένη σύνταξη που περιγράφεται μέσα στη βάση (ftp://ftp.ncbi.nlm.nih.gov/blast/documents/blastdb.txt)

6.4.1ΠΑΡΑΛΛΑΓΕΣ

Υπάρχουν πολλές παραλλαγές του προγράμματος BLAST, οι οποίες εκτελούν διαφορετικές εργασίες.
blastn - συγκρίνει μια νουκλεοτιδική ακολουθία (DNA) με μια βάση νουκλεοτιδικών ακολουθιών (DNA). Η αναζήτηση γίνεται και στις δύο αλυσίδες. Είναι ένα πρόγραμμα βελτιστοποιημένης ταχύτητας, όχι όμως και ευαισθησίας.
blastp - συγκρίνει την ζητούμενη αμινοξική ακολουθία με μια βάση πρωτεϊνικών ακολουθιών (σύγκριση πρωτεΐνης με πρωτεΐνες).
blastx - συγκρίνει μια άγνωστη νουκλεοτιδική ακολουθία (DNA) μεταφρασμένη σε όλα τα πλαίσια ανάγνωσης (reading frames) με μια βάση πρωτεϊνικών ακολουθιών του NCBI. Χρησιμοποιείται για την έυρεση πιθανών μεταφρασμένων πρωτεϊνικών προϊόντων μιας άγνωστης νουκλεοτιδικής ακολουθίας.
tblastn - συγκρίνει την ζητούμενη πρωτεϊνική ακολουθία με μια βάση νουκλεοτιδικών ακολουθιών (DNA) του NCBI που μεταφράζεται δυναμικά σε όλα τα πλαίσια ανάγνωσης (reading frames).
tblastx - μετατρέπει μια νουκλεοτιδική ακολουθία (DNA) σε μια πρωτεϊνική ακολουθία σε όλα τα πλαίσια ανάγνωσης (reading frames) και μετά τη συγκρίνει με μια βάση νουκλεοτιδικών ακολουθιών του NCBI η οποία έχει μεταφραστεί σε όλα τα πλαίσια ανάγνωσης (reading frames).
BLAST2 - Ονομάζεται εξελιγμένο (advanced) BLAST. Εκτελεί στοιχίσεις που περιέχουν κενά (gapped alignments).
MEGABLAST - είναι ένα πρόγραμμα που χρησιμοποιεί έναν πλεονεκτικό ("greedy algorithm") αλγόριθμο (Miller *et al*, 2000), για αναζήτηση στοίχισης νουκλεοτιδικών ακολουθιών. Χρησημοποιείται για στοίχιση ακολουθιών με μικρές διαφορές και είναι 10 φορές γρηγορότερο από παρόμοια προγράμματα. Ενδείκνυται για σύγκριση μεταξύ μεγάλων ακολουθιών.
PSI-BLAST - (Position Specific Iterated BLAST) χρησιμοποιεί σταθερή αναζήτηση, στην οποία οι ακολουθίες που θα βρεθούν στον πρώτο γύρο αναζητήσεων χρησιμοποιούνται για να χτίσουν ένα αποτελεσματικό μοντέλο για τους επόμενους κύκλους αναζητήσεων.
PHI-BLAST - (Pattern Hit Initiated BLAST) συνδυάζει το ταίριασμα ενός πρότυπου φυσιολογικής έκφρασης με μια συγκεκριμένη θέση που επαναλαμβάνεται στην πρωτεϊνική ακολουθία.
RPS-BLAST - συγκρίνει μια πρωτεϊνική ακολουθία ως προς την βάση Conserved Domain Database (CD-Search).
Αναλυτικό TUTORIAL για το BLAST υπάρχει στην διεύθυνση:
http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html

6.5CLUSTALW

Το CLUSTALW είναι το πιο διαδεδομένο πρόγραμμα πολλαπλής στοίχισης βιολογικών ακολουθιών. Το πρόγραμμα χρησιμοποιεί έναν ιδιαίτερα πολύπλοκο αλγόριθμο προδευτικής στοίχισης (progressive alignment) για να κάνει σταδιακή στοίχιση πολλαπλών πρωτεϊνικών ή νουκλεοτιδικών (DNA) ακολουθιών. Όλες οι ακολουθίες πρέπει να είναι ένα αρχείο, η μία μετά την άλλη. Το πρόγραμμα CLUSTALW βρίσκεται στο EBI (European Bioinformatics Institute) (http://www.ebi.ac.uk/clustalw/).
Το CLUSTALX (Thompson et al., 1994) είναι μια έκδοση του προγράμματος, το οποίο χρησιμοποιεί το περιβάλλον των X-Windows για να τρέχει το πρόγραμμα CLUSTALW. Το CLUSTALX δίνει την επιλογή δύο τρόπων στοίχισης ακολουθιών. Την πολλαπλή στοίχιση (*multiple alignment mode*) και την στοίχιση με βάση κάποιο profile(*profilealignmentmode*), στη διεύθυνση http://www.ebi.ac.uk/clustalw/help.html)

Για να γίνει πολλαπλή στοίχιση ενός συνόλου ακολουθιών πρέπει να επιλεγεί ο τρόπος πολλαπλής στοίχισης (*multiple alignment mode*). Όλες οι ακολουθίες συγκρίνονται μεταξύ τους, επιτρέποντας την κατασκευή ενός δενδρογράμματος (dendrogram) που δείχνει την κατά προσέγγιση ομάδοποίηση των ακολουθιών λόγο της ομοιότητας τους. Η τελική πολλαπλή στοίχιση των ακολουθιών γίνεται χρησιμοποιώντας ως οδηγό αυτό το δενδρόγραμμα.

Ο δεύτερος τρόπος στοίχισης ακολουθιών είναι η στοιχιση με βάση κάποιο profile (*profile alignment mode*). Υπάρχουν οι περιοχές στοιχείων των ακολουθιών (sequence data areas), που επιτρέπουν την στοίχιση τους. Τα profiles χρησιμοποιούνται επίσης για την προσθήκη μιας νέας ακολουθίας σε μια παλιά στοίχιση ή χρησιμοποιούν μια δευτεροταγή δομή για να κατευθύνουν τη διαδικασία της στοίχισης. Τα κενά (*gaps*) παλαιότερων στοιχίσεων δηλώνονται με το χαρακτήρα "-".
Μετά το τέλος των στοιχιζόμενων ακολουθιών επακολουθεί μια σειρά που μπορεί να περιέχει τους παρακάτω χαρακτήρες, η σημασία των οποίων είναι η εξής:
* = αυτή η στήλη της στοίχισης περιέχει ίδιο το αμινοξικό κατάλειπο σε όλες τις ακολουθίες (ή όμοιες βάσεις αν στοιχίζονται ακολουθίες DNA)
: = αυτή η στήλη της στοίχισης περιέχει διαφορετικά αλλά υψηλά διατηρημένα (παρόμοια) αμινοξέα.
. = αυτή η στήλη της στοίχισης περιέχει διαφορετικά αμινοξέα τα οποία είναι κατα κάποιον τρόπο όμοια.
= αυτή η στήλη της στοίχισης περιέχει ανόμοια αμινοξέα ή κενά (διαφορετικές βάσεις σε στοίχιση ακολουθιών DNA)

Τα φυλογενετικά δέντρα (**Phylogenetic trees**) μπορούν να υπολογιστούν από παλαιότερες στοιχίσεις ή μετά από μια πολλαπλή στοίχιση ενώ η στοίχιση βρίσκεται ακόμα σε εξέλιξη. Το **CLUSTAL** μπορεί να εκτελέσει πολλαπλές στοιχίσεις για περισσότερες απο 100 νουκλεοτιδικές (DNA) ή πρωτεϊνικές ακολουθίες αποτελούμενες μέχρι και 5000 κατάλοιπα (περιλαμβανομένων των κενών στην τελική στοίχιση).

ΕΠΙΛΟΓΟΣ

[14]Τα τελευταία χρόνια οι υπολογιστές κατακτούν σημαντική θέση σε κάθε τομέα της ζωής μας αλλά και σε αρκετούς τομείς διαφόρων επιστημών. Η Βιοπληροφορική αποτελεί ένα σύγχρονο τομέα έρευνας και ανάπτυξης τόσο για τους μοριακούς βιολόγους όσο και για τους επιστήμονες της πληροφορικής. Η συνεργασία των δύο αυτών επιστημών χαρακτηρίζεται αρκετά υποσχόμενη και με ιδιαίτερη σημασία αφού έρχεται να ρίξει φως στην ερμηνεία και το ρόλο της γονιδιακής πληροφορίας και κατ' επέκταση σε αρκετές διαδικασίες της ζωής που ζητούν ερμηνεία. Η Βιοπληροφορική είναι αυτή η επιστήμη η οποία παρέχει τα εργαλεία και τις μεθόδους τα οποία υποστηρίζουν την ανάγκη για την εκμετάλλευση υπολογιστικής ισχύος και την εξαγωγή γνώσης από βιολογικά δεδομένα. Η έρευνα σε αυτήν την περιοχή περιλαμβάνει την ανάλυση γενετικής/γονιδιωματικής πληροφορίας, με στόχο την πρόβλεψη, ή τον ακριβή καθορισμό βιολογικών λειτουργιών. Για το σκοπό αυτό, συνδυάζονται πολλές επιστήμες από διαφορετικές περιοχές, όπως Γονιδιωματική, Πληροφορική, Φαρμακευτική, Μοριακή Βιολογία, Στατιστική, Φυλογενετική κλπ.[14]

[15]Ο εικοστός αιώνας ολοκληρώθηκε αφήνοντας παρακαταθήκη την ωριμότητα δύο μεγάλων επιστημονικών περιοχών, της Μοριακής Βιολογίας και της Πληροφορικής. Η Μοριακή Βιολογία είναι η αποκωδικοποίηση του φαινομένου της Ζωής σε επίπεδο μορίων ή, όπως θα μπορούσε να χαρακτηρισθεί, η Κοινωνιολογία των Μορίων. Η Πληροφορική ασχολείται με την επεξεργασία δεδομένων και τη βελτίωση της ταχύτητας και ακρίβειας, δίνοντας τη δυνατότητα για σχεδιασμό ακόμη και συστημάτων που υποκαθιστούν νοητικές διεργασίες.

Καθώς η Μοριακή Βιολογία έχει παράγει έναν τεράστιο όγκο πληροφοριών και η Πληροφορική έχει ωριμάσει για να μπορεί να αναγνωρίζει και να επεξεργάζεται τέτοιους όγκους δεδομένων, σκοπός είναι πια το «πάντρεμα» των δύο αυτών επιστημών με έναν κοινό στόχο, ο οποίος δεν είναι άλλος από τη μίμηση της λειτουργίας του κυττάρου στο περιβάλλον του Υπολογιστή. Ο νέος κλάδος που προκύπτει είναι ο κλάδος της Υπολογιστικής Μοριακής Βιολογίας ή κλάδος της Βιοπληροφορικής, που σιγά-σιγά θα καθιερωθεί ερμηνεύοντας από τη μία τη φυσιολογική και παθολογική λειτουργία του κυττάρου και επαναπροσδιορίζοντας από την άλλη τις δικές του αφετηρίες.

Η αποκωδικοποίηση του ανθρώπινου γονιδιώματος αποτελεί την απαρχή ραγδαίων εξελίξεων στις Επιστήμες της Ζωής. Το γεγονός ότι μέχρι τώρα έχει απλώς καταγραφεί η αλληλουχία των βάσεων (Αδενίνης, Κυτοσίνης, Γουανίνης, και Θυμίνης), δίνει μόνο τη δυνατότητα ανάγνωσης της γλώσσας του γενετικού υλικού, αλλά όχι την κατανόησή της. Στόχος της Βιοπληροφορικής είναι να δώσει στα επόμενα χρόνια το αναγκαίο μέσο κατανόησης της συμπεριφοράς των γονιδίων, να συμβάλλει στην καταγραφή του τρόπου δράσης τους και να δώσει απαντήσεις σε πολύπλοκα ερωτήματα. Η ανάλυση και η επεξεργασία των γονιδιωμάτων διάφορων οργανισμών θα συμβάλλει σημαντικά στην κατανόηση και περιγραφή σημαντικών λειτουργιών τους. Έτσι χρήσιμες πληροφορίες θα δοθούν στους γενετιστές για τη φυσιολογία και την εξέλιξη των οργανισμών. Παράλληλα όμως είναι δυνατή και η σύγκριση του γενετικού υλικού διαφορετικών οργανισμών, αναζητώντας κοινές συμπεριφορές, λειτουργίες και προβλήματα, εξάγοντας συμπεράσματα για τη μέχρι τώρα εξέλιξη των ειδών αλλά και για τις μελλοντικές εξελίξεις.

Η χρήση της Πληροφορικής για την ανάλυση του DNA θα συνεισφέρει αφενός στο να κατανοηθεί η γλώσσα του γενετικού υλικού και αφετέρου στο να εντοπισθούν και να αντιμετωπισθούν γενετικές ασθένειες. Καθώς η προδιάθεση των παιδιών για ορισμένες ασθένειες είναι κωδικοποιημένη στο γενετικό υλικό, με την κατάλληλη προληπτική ιατρική θεραπεία μπορούν έγκαιρα να αποφευχθούν οι παράγοντες που μπορούν να εκδηλώσουν την συγκεκριμένη ασθένεια. Η επεξεργασία των γονιδίων με μεθόδους της Πληροφορικής θα συμβάλλει στην κατανόηση των λειτουργιών των πρωτεϊνών, που τα γονίδια κωδικοποιούν. Έχοντας κατανοήσει τα χαρακτηριστικά και τις λειτουργίες των πρωτεϊνών, θα μπορούμε είτε να εμποδίζουμε είτε να προκαλούμε κάποιες από αυτές με σκοπό τον καθορισμό της εξέλιξης ασθενειών. Παράλληλα έχουν ωριμάσει οι συνθήκες για την επιλεκτική φαρμακευτική στόχευση και σύντομα αναμένεται η συνεισφορά της Βιοπληροφορικής και στο σχεδιασμό νέων φαρμάκων. Πιο συγκεκριμένα στόχος είναι να σχεδιαστεί ένα φάρμακο χτισμένο ειδικά πάνω στο γονιδιακό υπόστρωμα του κάθε ασθενούς, δηλαδή μια εξατομικευμένη φαρμακευτική αντιμετώπιση.

Για την πραγματοποίηση των παραπάνω στόχων η Πληροφορική μπορεί να προσφέρει τα εργαλεία και τις μεθόδους για την αποτελεσματική αναπαράσταση, αποθήκευση και επεξεργασία μεγάλου όγκου δεδομένων, όπως οι κατάλληλες Βάσεις Δεδομένων και Δομές Δεδομένων, ο Δυναμικός και ο Γραμμικός προγραμματισμός για την επίλυση προβλημάτων, Τεχνικές Ανάκτησης Πληροφορίας για την επεξεργασία του γενετικού υλικού, Μεθόδους Προσομοίωσης για τη μοντελοποίηση των λειτουργιών και συμπεριφορών των οργανισμών. Προσφέροντας αυτά τα όπλα στη Βιολογία μέσω της Βιοπληροφορικής, είναι βέβαιο ότι δημιουργούνται ερωτήσεις χαμηλού επιπέδου ως προς την πλήρη ερμηνεία της δομής και λειτουργίας των ζώντων οργανισμών, αλλά είναι όμως και αρκετά σοβαρές ώστε να προσελκύσουν πλήθος επιστημόνων και επενδύσεων, ανοίγοντας μεγάλους, αλλά και απρόβλεπτους ορίζοντες στην κατανόηση των νόμων της Ζωής.[15]

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]http://el.wikipedia.org/wiki/%CE%92%CE%B9%CE%BF%CF%80%CE%BB%CE%B7%CF%81%CE%BF%CF%86%CE%BF%CF%81%CE%B9%CE%BA%CE%AE

[2]http://el.wikipedia.org/wiki/DNA

[3]http://el.wikipedia.org/wiki/RNA

[4]http://el.wikipedia.org/wiki/%CE%A0%CF%81%CF%89%CF%84%CE%B5%CE%90%CE%BD%CE%B7

[5]http://www.ebi.ac.uk/2can/tutorials/nucleotide/blast6.html

[6]http://en.wikipedia.org/wiki/BLAST

[7]http://www-igm.univ-mlv.fr/~lecroq/string/node14.html

[8]http://www-igm.univ-mlv.fr/~lecroq/string/node8.html

[9]http://www-igm.univ-mlv.fr/~lecroq/string/node29.html

[10]http://www.csd.uoc.gr/~hy463/2005/slides/463_09_Text_Indexing_Organization.pdf

[11]http://en.wikipedia.org/wiki/Boyer%E2%80%93Moore%E2%80%93Horspool_algorithm

[12]http://www-igm.univ-mlv.fr/~lecroq/string/node18.html

[13]http://biophysics.biol.uoa.gr/courses/bioinformatics/BLAST_CLUSTAL/BLAST_CLUSTAL.html

[14] http://ekfe.reth.sch.gr/index.php?option=content&task=view&id=343

[15] http://www.ceid.upatras.gr/tech_news/21_06_04.pdf

ΒΙΒΛΙΟΓΡΑΦΙΑ

''Algorithms on Strings, Trees and Sequences", D. Gusfield, Cambridge University Press, 1999''

Πανεπιστημιακές σημειώσεις των : Α. Περδικούρη, Α. Τσακαλίδη, με τίτλο "Εισαγωγή στη Βιοπληροφορική".

Βιβλίο Βιολογίας Β' Λυκείου