Αποδοτικοί αλγόριθμοι για επεξεργασία αλληλουχιών (sequences) από τις υψηλού όγκου παραγωγής (throughput) τεχνολογίες

Θεόδωρος Χριστάκη
ΑΕΜ: 368
Πανεπιστήμιο Θεσσαλίας
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών Τηλεπικοινωνιών και Δικτύων

Επιβλέπων καθηγητής
Π. Δ. Μποζάνης

Φεβρουάριος 2013, Βόλος

Περίληψη

Μια νέα γενιά τεχνολογιών παραγωγής γενετικών δεδομένων σηματοδοτεί την απαρχή μιας νέας εποχής για τον προσδιορισμό αλληλουχίας (sequencing) του DNA, παράγοντας εκατομμύρια μικρές ακολουθίες (short reads) πολύ ταχύτερα και σε πολύ μικρότερο κόστος απ' ότι στο παρελθόν. Αυτή η παραγωγή μεγάλων ποσοτήτων γενετικών δεδομένων, με τη μορφή μικρών υποακολουθιών του DNA, των short reads, από στενά συσχετιζόμενα είδη ή οργανισμούς του ιδίου είδους, είναι ο κινητήριος μοχλός για την εφαρμογή που είναι γνωστή ως *επαναπροσδιορισμός αλληλουχίας* (re-sequencing). Ο αντίκτυπος αυτών των καινοτόμων συστημάτων καταγραφής αλληλουχίας επόμενης γενιάς στην κλινική Γενετική, θα είναι σίγουρα μεγάλος.

Σε αυτήν τη Διπλωματική Εργασία παρουσιάζουμε σχετικά καινούργιες αλγοριθμικές μεθόδους για την ανάλυση γενετικών δεδομένων επόμενης γενιάς. Ειδικότερα, στο Κεφάλαιο 3, παρουσιάζουμε το REadALigner (REAL), ένα αποτελεσματικό, ευαίσθητο και ακριβές πρόγραμμα αντιστοίχισης για την ευθυγράμμιση εκατομμυρίων short reads σε ένα γονιδίωμα αναφοράς. Στο Κεφάλαιο 4, παρουσιάζουμε το cREAL, μια απλή επέκταση του REAL, ειδικά σχεδιασμένη για την ευθυγράμμιση εκατομμυρίων short reads σε ένα γονιδίωμα αναφοράς με κυκλική δομή.

Μετά από μια σειρά πειραμάτων δείχνουμε ότι αυτά τα δύο προγράμματα μπορούν να ανταγωνιστούν, ή ακόμη και να ξεπεράσουν, τα τρέχοντα δημοφιλή προγράμματα ευθυγράμμισης short reads, όπως το Bowtie και το SOAP2, όσον αφορά την αποτελεσματικότητα, την ευαισθησία, και την ακρίβεια.

Περιεχόμενα

1 Εισαγωγή **1**

2 Ορισμοί **7**
2.1 Αλφάβητο και συμβολοσειρές 8
2.2 Ευθυγράμμιση συμβολοσειρών 10
2.3 Ασυμπτωτική πολυπλοκότητα 13
2.4 Βασικές δομές δεδομένων 14
2.4.1 Στοίβες και ουρές 15
2.4.2 Συνδεδεμένες λίστες 18

3 Ευθυγράμμιση γενετικών δεδομένων νέας γενιάς σε ένα γονιδίωμα **21**
3.1 Αλγόριθμος 22
3.2 Πειραματικά αποτελέσματα 29

4 Ευθυγράμμιση γενετικών δεδομένων νέας γενιάς σε ένα γονιδίωμα με κυκλική δομή **35**
4.1 Αλγόριθμος 36
4.2 Πειραματικά αποτελέσματα 38

5 Συμπεράσματα **43**

Κατάλογος Πινάκων

3.1 Κωδικοποίηση με δύο bits ανά βάση του αλφάβητου $\Sigma = \{\texttt{A,C,G,T}\}$ 23
3.2 Ευθυγράμμιση 25.000.000 short reads μήκους 64 bp στο ανθρώπινο χρωμόσωμα 6 (166, 880, 988 bp) 31
3.3 Ευθυγράμμιση 24.543.488 short reads μήκους 70 bp στην ακολουθία Drosophila melanogaster 3L (24, 543, 557 bp) 31
3.4 Ευθυγράμμιση 24, 163, 065 short reads μήκους 76 bp σε ολόκληρη την ακολουθία του ανθρώπινου γονιδιώματος 32
3.5 Ευθυγράμμιση 31.116.663 short reads μήκους 25 bp και 3.619.970 short reads μήκους 35 bp σε ολόκληρη την ακολουθία του γονιδιώματος του ποντικού 33

4.1 Ευθυγράμμιση 9, 105, 777 short reads μήκους 52 bp στο γονιδίωμα Bradyrhizobium Japonicum 40
4.2 Ευθυγράμμιση 3, 294, 805 short reads μήκους 64 bp στο γονιδίωμα Brucella melitensis 16M 40
4.3 Ευθυγράμμιση 6, 264, 333 short reads μήκους 72 bp στο γονιδίωμα Pseudomonas aeruginosa PAO1 40
4.4 Ευθυγράμμιση 2, 475, 055 short reads μήκους 76 bp στο γονιδίωμα Xylella fastidiosa M12 41
4.5 Ευθυγράμμιση 5, 288, 154 short reads μήκους 36 bp στο γονιδίωμα Escherichia Coli 41

Κατάλογος Σχημάτων

2.1 Μονά-συνδεδεμένη λίστα 19

3.1 Λίστα X_j, για κάθε $0 \leq j < \binom{q}{q-k}$, για $q = 4$ και $k = 2$, μετά το Βήμα (I) 27

3.2 Λίστα X_j, για κάθε $0 \leq j < \binom{q}{q-k}$, για $q = 4$ και $k = 2$, μετά το Βήμα (II) 28

4.1 Κυκλικό κείμενο $C(t)$. Το σημείο που η κεφαλή του βέλους αγγίζει το οβάλ βέλος είναι το σημείο που διασπάστηκε η συμβολοσειρά. Για να ψάξουμε για ένα μοτίβο p μήκους m σε αυτό το κυκλικό κείμενο, τα πρώτα $m-1$ γράμματα της συμβολοσειράς λαμβάνονται από το σημείο όπου ξεκινά η διάσπαση στο $t[0]$, και προστίθενται στο τέλος $t[n-1]$. 38

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου.

Κεφάλαιο 1

Εισαγωγή

Η καταγραφή αλληλουχίας (sequencing) του DNA περιλαμβάνει διάφορες μεθόδους και τεχνολογίες που χρησιμοποιούνται για τον προσδιορισμό της ακριβής σειράς των βάσεων των νουκλεοτιδίων—αδενίνη, γουανίνη, κυτοσίνη και θυμίνη—σε ένα μακρομόριο του DNA.

Η τεχνολογία του προσδιορισμού αλληλουχίας έχει εξελιχθεί σε μεγάλο βαθμό από την πρώτη αποκωδικοποίηση του ανθρώπινου γονιδιώματος, όταν οι παραδοσιακές μέθοδοι απαιτούσαν συνεργασία πολλών εργαστηρίων ανά τον κόσμο για χρόνια προκειμένου να φτάσουν στο επιθυμητό αποτέλεσμα. Οι παραδοσιακές μεθόδοι προσδιορισμού αλληλουχίας, που αναπτύχθηκαν στα μέσα της δεκαετίας του '70, αποτελούσαν το κύριο εργαλείο ανάλυσης του DNA για 30 περίπου χρόνια. Το 1977,ο Fred Sanger και ο Alan R. Coulson δημοσίευσαν δύο μεθοδολογικές μελέτες για τον ταχύ προσδιορισμό της αλληλουχίας του DNA [44, 42], οι οποίες θα έφερναν επανάσταση στη Βιολογία γενικότερα, παρέχοντας ένα εργαλείο ανάλυσης γονιδίων και, αργότερα, ολόκληρων γονιδιωμάτων. Οι μέθοδοι αυτοί βελτίωσαν σε μεγάλο βαθμό την αρχική τεχνική προσδιορισμού αλληλουχίας του DNA που ανέπτυξαν οι Maxam και Gilbert [32], που δημοσιεύτηκε την ίδια χρονιά, όπως και την παλαιότερη μέθοδο των ίδιων των Sanger και Coulson, που δημοσίευσαν δυο χρόνια νωρίτερα [43].

Με υπέρτατο στόχο την ανάλυση του ανθρώπινου γονιδιώματος, η ανάγκη για προσδιορισμό αλληλουχίας του DNA αυξήθηκε σε απρόσμενο βαθμό, οδηγώντας σε νέες εξελίξεις. Ο αυτοματισμός των εργαστηρίων και ο παραλληλισμός των διαδικασιών είχε ως αποτέλεσμα την ίδρυση εργοστασιακού τύπου εταιρειών, που ονομάζονται κέντρα καταγραφής αλληλουχίας, τα οποία στεγάζουν εκατοντάδες όργανα μελέτης του DNA. Ωστόσο, ακόμη και η επιτυχής ολοκλήρωση των δύο ανταγωνιστικών προγραμμάτων αποκωδικοποίησης του ανθρώπινου γονιδιώματος δεν ικανοποίησε την' πείνα' των βιολόγων για μεγαλύτερη απόδοση της όλης διαδικασίας και, ιδιαίτερα, για πιο οικονομική. Από τότε, η ταχύτητα, η ακρίβεια, η αποτελεσματικότητα και το κόστος των τεχνολογιών ανάλυσης βελτιώνεται όλο και περισσότερο.

Τα πρώτα σημάδια της μεταμόρφωσης της αγοράς προσδιορισμού αλληλουχίας εμφανίστηκαν το 2005, με τη δημοσίευση ορόσημο της τεχνολογίας sequencing-by-synthesis (SBS), που αναπτύχθηκε από την 454 Life Sciences [31], και το πρωτόκολλο multiplex polony sequencing του εργαστηρίου του George Church [46]. Η τεχνολογία SBS, η οποία χρησιμοποιεί καταγραφή αλληλουχιών με πυροσφωρική ανίχνευση (pyrosequencing) για συλλογή δεδομένων, ξεκίνησε αρχικά με την παραγωγή μικρών αλληλουχιών(ακολουθιών)μ ήκους 100 ζευγών βάσεων (bp), τα οποία μετά από 16 μήνες στην αγορά αυξήθηκαν σε 250 bp. Νεώτερες εξελίξεις ανέβασαν τον πήχη στα 500 bp, πλησιάζοντας το σημερινό μήκος ανάγνωσης του Sanger sequencing, 750 bp.

Εκτός από το μήκος ανάγνωσης, ο αριθμός των ακολουθιών που μπορεί να παραχθεί σε μια και μόνο διαδικασία ενός οργάνου για ένα δεδομένο κόστος,

είναι μια άλλη σημαντική πτυχή. Οι ανησυχίες αυτές έχουν αντιμετωπιστεί από τους ανταγωνιστές της 454, των οποίων τα συστήματα παράγουν έως και δέκα φορές περισσότερα δεδομένα, με όμως πολύ μικρότερο μήκος ανάγνωσης, των 150 ή λιγότερων bp. Σήμερα, τρία εμπορικά συστήματα καταγραφής αλληλουχίας επόμενης γενιάς είναι διαθέσιμα: το σύστημα Genome Sequencer FLX της Roche (454), που πωλείται από την Roche Applied Sciences, το HiSeq sequencer της Illumina, και το πιο πρόσφατο σύστημα SOLiD της Applied Biosystem. Πρόσθετοι ανταγωνιστές, οι οποίοι πιστεύεται ότι είναι έτοιμοι να εισέλθουν στην αγορά, αφορούν συστήματα καταγραφής αλληλουχίας τρίτης γενιάς(ονομάζονται επίσης και συστήματα επόμενης γενιάς) που βασίζονται σε ανάλυση ενός μορίου (single-molecule analysis), και αναπτύσσονται από τις εταιρείες VisiGen και Helicos.

Ο αντίκτυπος αυτών των καινοτόμων συστημάτων καταγραφής αλληλουχίας επόμενης γενιάς στην κλινική Γενετική, θα είναι σίγουρα μεγάλος. Η χαμηλής κλίμακας, στοχευμένη ανάλυση γονιδίων/μεταλλάξεων που κυριαρχεί σήμερα στον τομέα της κλινικής Γενετικής, θα αντικατασταθεί τελικά από μεγάλης κλίμακας ανάλυση των γονιδίων/νόσων και ολόκληρων των δικτύων τους, ειδικά για τις λεγόμενες σύνθετες διαταραχές [38, 39, 48]. Τελικά, το κλινικό όφελος από την καταγραφή ολόκληρου του γονιδιώματος θα αντισταθμίσει το κόστος της διαδικασίας, επιτρέποντας στις δοκιμές αυτές να πραγματοποιούνται σε τακτική βάση για διαγνωστικούς σκοπούς, ή ίσως με τη μορφή ενός προσυμπτωματικού ελέγχου, που θα μπορούσε να χρησιμοποιηθεί ως οδηγός για εξατομικευμένες ιατρικές θεραπείες καθόλη τη διάρκεια ζωής του ατόμου [54].

Ο στόχος της παραγωγής μεγάλων ποσοτήτων γενετικών δεδομένων, με τη μορφή μικρών ακολουθιών (short reads), από στενά συσχετιζόμενα είδη ή οργανισμούς του ιδίου είδους, είναι ο κινητήριος μοχλός για την εφαρμογή που είναι γνωστή ως *επαναπροσδιορισμός αλληλουχίας* (re-sequencing), η οποία ασχολείται με τα δεδομένα με εντελώς διαφορετικό τρόπο από ότι στην *εκ νέου συναρμολόγηση αλληλουχίας* (de novo assemblies of genomes). Στον επαναπροσδιορισμό της αλληλουχίας, η συναρμολόγηση κατευθύνεται από μια ακολουθία αναφοράς και απαιτεί πολύ λιγότερη κάλυψη (8-12×)—κάθε βάση στην τελική αλληλουχία είναι παρούσα, κατά μέσο όρο, σε 8-12 short reads—από την εκ νέου συναρμολόγηση γονιδιωμάτων (25-70×). Απαιτεί την ύπαρξη υψηλής ποιότητας γονιδιώματος κάποιου εκπροσώπου των ειδών(ή γονιδίωμα αναφοράς), και η τεχνολογία προσδιορισμού αλληλουχίας χρησιμοποιείται για την παραγωγή short reads από το γονιδίωμα ενός άλλου αντιπροσώπου(ο δότης). Εάν ήταν εφικτό να προσδιοριστεί το γονιδίωμα του δότη από τα short reads, το να βρεθούν οι διαφορές μεταξύ των δύο γονιδιωμάτων, θα είναι σχετικά απλό. Ωστόσο, η εκ νέου συναρμολόγηση του γονιδιώματος από δεδομένα της τεχνολογίας προσδιορισμού αλληλουχίας επόμενης γενιάς (short reads), μπορεί να παράξει μόνο μικρά τμήματα του γονιδιώματος, που ονομάζονται contigs [33],

καθώς η παρουσία επαναλήψεων στο γονιδίωμα καθιστά δύσκολο ή αδύνατο να συγκεντρωθούν μεγαλύτερα τμήματα. Αντ΄ αυτού, στον επαναπροσδιορισμό αλληλουχίας, τα short reads συγκρίνονται(αντιστοιχίζονται) με την αλληλουχία αναφοράς, και έτσι οι μεταλλάξεις αναγνωρίζονται μέσω της ανάλυσης των αντιστοιχιζόμενων short reads. Μία πρόσφατη μελέτη, με τη χρήση αυτής της προσέγγισης, προσδιόρισε την αλληλουχία δέκα μιτοχονδριακών γονιδιωμάτων θηλαστικών [16], δίνοντας έτσι τη δυνατότητα πραγματοποίησης πληθυσμιακών γενετικών μελετών που βασίζονται σε πλήρη μιτοχονδριακά γονιδιώματα και όχι μόνο σε μικρά τμήματα ακολουθιών. Ωστόσο, προσπάθειες για την εκ νέου συναρμολόγηση αλληλουχίας απλούστερων γονιδιωμάτων έχουν ήδη αρχίσει [47], και μια πρώτη προσπάθεια για τη συναρμολόγηση του ανθρώπινου γονιδιώματος έχει επίσης δημοσιευθεί πρόσφατα [29].

Το πρώτο βασικό βήμα για την ανακάλυψη των μεταλλάξεων στο γονιδίωμα του δότη είναι η αντιστοίχιση των short reads σε μια ακολουθία αναφοράς. Η αντιστοίχιση τόσων πολλών short reads σε μια τέτοια μακρά αλληλουχία αναφοράς είναι ένα πολύ δύσκολο υπολογιστικά έργο που δεν μπορεί να πραγματοποιηθεί επαρκώς από τα παραδοσιακά προγράμματα ευθυγράμμισης, όπως το BLAST [1], το FASTA [40], ή το BLAT [22]. Ως εκ τούτου, πρόσφατα, ένα ευρύ φάσμα προγραμμάτων ευθυγράμμισης short reads έχει δημοσιευθεί προς τον σκοπό αυτόν, δίνοντας έμφαση στις διάφορες πτυχές αυτής της πρόκλησης.

Μια πρώτη γενιά προγραμμάτων ευθυγράμμισης short reads χρησιμοποίησε μεθόδους βασισμένους σε πίνακες κατακερματισμού (hashing tables). Το πρώτο αποτελεσματικό πρόγραμμα που αναπτύχθηκε ήταν το ELAND [8], το οποίο ενσωματώθηκε στο πακέτο επεξεργασίας δεδομένων Solexa της Illumina. Το ELAND ευρετηριοποιεί short reads μήκους 20-32 bp, και επιτρέπει μέχρι δύο αναντιστοιχίες στην ευθυγράμμιση. Το SOAP [27], αντί τα short reads, ευρετηριοποιεί το γονιδίωμα αναφοράς, και επιτρέπει είτε ορισμένες αναντιστοιχίες ή ένα συνεχόμενα κενό (gap), μήκους που κυμαίνεται από ένα έως τρία, στην ευθυγράμμιση. Το κενό μπορεί να τοποθετηθεί είτε στο short read είτε στην ακολουθία αναφοράς. Το SeqMap [20] ευρετηριοποιεί τα short reads και προσφέρει μεγαλύτερη ευελιξία όσον αφορά στην ευθυγράμμιση. Επιτρέπει έως και πέντε μικτές αντικαταστάσεις, διαγραφές και προσθήκες βάσεων. Το MAQ [26] ευρετηριοποιεί τα short reads, αλλά δεν υποστηρίζει ευθυγραμμίσεις με κενό. Ωστόσο, είναι το πρώτο πρόγραμμα ευθυγράμμισης short reads που αξιολογεί την αξιοπιστία της ευθυγράμμισης με τη βαθμολογία ποιότητας σε κάθε επιμέρους αντιστοίχιση, αξιολογώντας την πιθανότητα η πραγματική αντιστοίχιση να μην είναι αυτή που βρέθηκε. Επιπλέον, αξιοποιεί πλήρως τις επιμέρους πληροφορίες ενός ζεύγους short reads για να διορθώσει λάθος ευθυγραμμίσεις, για να προσθέσει αξιοπιστία διορθώνοντας ευθυγραμμίσεις και για να αντιστοιχηθούν με ακρίβεια τα short reads σε επαναλαμβανόμενες υπο-αλληλουχίες αν ένα από τα δύο short reads είναι ευθυγραμμισμένο με μεγάλη σιγουριά.

Η σημερινή δεύτερη γενιά προγραμμάτων ευθυγράμμισης short reads χρησιμοποιεί τον Burrows-Wheeler Transform (BWT) [4] για τη δημιουργία ενός μόνιμου ευρετηρίου της αλληλουχίας αναφοράς, το οποίο μπορεί να επαναχρησιμοποιηθεί σε ευθυγραμμίσεις, και είναι σε θέση να επιτύχει πολύ καλή ταχύτητα και αποτελεσματικότητα μνήμης. Το Bowtie [24] ευρετηριοποιεί την ακολουθία αναφοράς χρησιμοποιώντας ένα σχήμα βασισμένο στο BWT και το ευρετήριο FM [11]. Χρησιμοποιεί τον αλγόριθμο αντιστοίχισης αλληλουχιών των Ferragina και Manzini [11] για την αναζήτηση στο ευρετήριο FM. Εισάγει, επίσης, δύο επεκτάσεις του προβλήματος της κατά προσέγγιση αντιστοίχισης ακολουθιών (approximate string matching), επιτρέποντας διαφορές μεταξύ τους: έναν αλγόριθμο οπισθοδρόμησης με επίγνωση ποιότητας που επιτρέπει αναντιστοιχίες και ευνοεί ευθυγραμμίσεις υψηλής ποιότητας, και ένα σχήμα διπλής ευρετηριοποίησης, μια στρατηγικής για να αποφευχθεί η οπισθοδρόμηση. Ακολουθεί επίσης μια τακτική παρόμοια με αυτή του MAQ, επιτρέποντας ένα μικρό αριθμό αναντιστοιχιών εντός της ευθυγράμμισης υψηλής ποιότητας, και θέτει ανώτατο όριο για το άθροισμα των βαθμολογιών ποιότητας (quality scores) των αναντιστοιχιών στην ευθυγράμμιση. Δηλαδή το Bowtie δεν ενδιαφέρεται απλά μόνο για τον αριθμό των αναντιστοιχιών αλλά λαμβάνει υπόψιν και τη βαθμολογία ποιότητας των βάσεων στις αντίστοιχες θέσεις. Το BWA [25], ο διάδοχος του MAQ, χρησιμοποιεί αναζήτηση προς τα πίσω [11, 30] με BWT, και είναι σε θέση να μιμηθεί αποτελεσματικά την από πάνω προς τα κάτω προσπέλαση των δεδομένων στο prefix trie του γονιδιώματος αναφοράς με σχετικά μικρό αποτύπωμα μνήμης [23], και να μετρήσει τον ακριβή αριθμό των εμφανίσεων μιας συμβολοσειράς σε γραμμικό χρόνο, ανεξάρτητα από το μέγεθος του γονιδιώματος. Για την κατά προσέγγιση αναζήτηση, το BWA ελέγχει δειγματολειπτικά από το prefix trie τις διαφορετικές συμβολοσειρές που έχουν απόσταση (edit distance) μικρότερη του k από το short read. Επειδή οι ακριβείς επαναλήψεις υποακολουθιών του γονιδιώματος αναφοράς καταλήγουν σε ένα μονοπάτι στο prefix trie, δε χρειάζεται να ευθυγραμμιστεί το short read με κάθε αντίγραφο της επανάληψης. Το SOAP2 [28], σε αντίθεση με το SOAP, χρησιμοποιεί επίσης το BWT για να ευρετηριοποιήσει και να αποθηκεύεσει την ακολουθία της αναφοράς στη δευτερεύουσα μνήμη, αντί να την ευρετηριοποιήσει στην κύρια μνήμη. Αναζητεί μια ακριβή ευθυγράμμιση με την κατασκευή ενός πίνακα κατακερματισμού για να επιταχύνει την αναζήτηση για τη θέση εμφάνισης του short read στο BWT ευρετήριο. Για παράδειγμα, για έναν πίνακα κατακερματισμού υποακολουθιών μήκους 13 bp, το ευρετήριο θα πρέπει να αποτελείται από $2^{2.13}$ τμήματα(2-bits-per-base κωδικοποίηση), και πολύ λίγες αλληλεπιδράσεις αναζήτησης είναι επαρκείς για να προσδιορίσουν την ακριβή θέση εμφάνισης του τμήματος. Για κατά προσέγγιση αναζήτηση, δηλαδή, αναντιστοιχίες, διαγραφές και προσθήκες, εφαρμόζεται η γνωστή τακτική του διαχωρισμού σε ακριβή τμήματα [37].

Σε αυτή τη Διπλωματική Εργασία παρουσιάζουμε αλγοριθμικές μεθόδους για την ανάλυση γενετικών δεδομένων επόμενη γενιάς. Ειδικότερα, το επίκεντρο αυτής της Εργασίας είναι σχετικά με την εφαρμογή του επαναπροσδιορισμού αλληλουχίας. Στο Κεφάλαιο 2, παρουσιάζουμε τους βασικούς ορισμούς και συμβολισμούς που χρησιμοποιούνται σε ολόκληρο την Εργασία. Στο Κεφάλαιο 3, παρουσιάζουμε το REadALigner (REAL), ένα αποτελεσματικό, ευαίσθητο και ακριβές πρόγραμμα ευθυγράμμισης για την αντιστοίχιση εκατομμυρίων short reads με ένα γονιδίωμα αναφοράς. Στο Κεφάλαιο 4, παρουσιάζουμε το cREAL, μια απλή επέκταση του REAL, ειδικά σχεδιασμένη για τη αντιστοίχιση εκατομμυρίων short reads σε ένα γονιδίωμα αναφοράς με κυκλική δομή. Τέλος, καταλήγουμε συνοπτικά στο Κεφάλαιο 5.

Κεφάλαιο 2

Ορισμοί

Σε αυτό το κεφάλαιο, εισάγουμε βασικούς ορισμούς και συμβολισμούς πάνω στο αλφάβητο και τις συμβολοσειρές, στην ευθυγράμμιση συμβολοσειρών, στην ασυμπτωτική πολυπλοκότητα αλγορίθμων και στις βασικές δομές δεδομένων. Τα περισσότερα από αυτά είναι πολύ κοινά και έτσι παραθέτουμε αναφορές μόνο σε πρότυπη βιβλιογραφία.

2.1 Αλφάβητο και συμβολοσειρές

Σε αυτήν την ενότητα εισάγουμε βασικούς ορισμούς και συμβολισμούς πάνω στο αλφάβητο και τις συμβολοσειρές από το [9].

M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on Strings*. Cambridge University Press, USA, 2007

Ορισμός Αλφάβητο)
Ένα αλφάβητο Σ *είναι ένα πεπερασμένο μη κενό σύνολο του οποίου τα στοιχεία λέγονται* σύμβολα *ή* γράμματα.

Ορισμός Συμβολοσειρά)
Μια συμβολοσειρά *σε ένα αλφάβητο* Σ *είναι μια πεπερασμένη, πιθανώς κενή, ακολουθία στοιχείων του* Σ.

Ορισμός Κενή συμβολοσειρά)
Η ακολουθία μηδέν συμβόλων λέγεται κενή συμβολοσειρά *και συμβολίζεται ως* ε.

Το σύνολο όλων των συμβολοσειρών ενός αλφαβήτου Σ συμβολίζεται ως Σ^*. Το σύνολο όλων των συμβολοσειρών ενός αλφαβήτου Σ εκτός της κενής συμβολοσειράς ε συμβολίζεται ως Σ^+.

Ορισμός Μήκος συμβολοσειράς)
Το μήκος μιας συμβολοσειράς x *ορίζεται ως το μήκος της ακολουθίας η οποία σχετίζεται με τη συμβολοσειρά* x *και συμβολίζεται ως* $|x|$.

Συμβολίζουμε ως $x[i]$, για κάθε $0 \leq i < |x|$, το γράμμα στο δείκτη i του x. Κάθε δείκτης i, για κάθε $0 \leq i < |x|$, είναι μια θέση στη x όταν $x \neq \varepsilon$. Ισχύει ότι το iοστό σύμβολο του x είναι το σύμβολο στη θέση $i-1$ της x και ότι

$$x = x[0\,..\,|x|-1]$$

Ορισμός Κυκλική συμβολοσειρά)
Η κυκλική συμβολοσειρά $y = C(x)$, η οποία σχηματίζεται από τη συμβολοσειρά $x = x[0\,..\,|x|-1]$, ορίζεται ως

$$y[|x|r + i] = x[i]$$

για κάθε $0 \leq i < |x|$, όπου $r \geq 0$.

Η κυκλική συμβολοσειρά μπορεί να εκληφθεί ως μια συμβολοσειρά η οποία δεν έχει δεξιότερη ούτε αριστερότερη θέση.

Ορισμός Ταυτότητα μεταξύ δύο συμβολοσειρών)
Η ταυτότητα μεταξύ δύο συμβολοσειρών x και y ορίζεται ως

$$x = y$$

αν και μόνον αν

$$|x| = |y| \text{ και } x[i] = y[i], \text{ για κάθε } 0 \leq i < |x|.$$

Ορισμός Παράθεση συμβολοσειρών)
Η παράθεση δύο συμβολοσειρών x και y είναι η συμβολοσειρά των συμβόλων της x ακολουθούμενη από αυτά της y. Συμβολίζεται ως xy.

Ορισμός Δύναμη συμβολοσειράς)
Για κάθε συμβολοσειρά x και κάθε φυσικό αριθμό n, ορίζουμε ως τη ποστή δύναμη της συμβολοσειράς x ως $x^0 = \varepsilon$ και $x^k = x^{k-1}x$, για κάθε $1 \leq k \leq n$, και τη συμβολίζουμε ως x^n.

Ορισμός Υποσυμβολοσειρά)
Μια συμβολοσειρά x είναι υποσυμβολοσειρά της συμβολοσειράς y αν υπάρχουν δύο συμβολοσειρές u και v, έτσι ώστε $y = uxv$.

Ορισμός Κανονική υποσυμβολοσειρά)
Μια υποσυμβολοσειρά x μια συμβολοσειράς y λέγεται κανονική αν $x \neq y$.

Ορισμός Υπερσυμβολοσειρά)
Μια συμβολοσειρά x είναι υπερσυμβολοσειρά της συμβολοσειράς y αν υπάρχουν δύο συμβολοσειρές u και v, έτσι ώστε $x = uyv$.

Ορισμός Πρόθεμα συμβολοσειράς)
Έστω ότι $y = uxv$, όπου $x, y, u,$ και v είναι συμβολσειρές. Αν $u = \varepsilon$, τότε η x λέγεται πρόθεμα της y.

Ορισμός Κατάληξη συμβολοσειράς)
Έστω ότι $y = uxv$, όπου $x, y, u,$ και v είναι συμβολσειρές. Αν $v = \varepsilon$, τότε η x λέγεται κατάληξη της y.

Ορισμός Εμφάνιση συμβολοσειράς)
Έστω ότι η x είναι μια μη κενή συμβολοσειρά και η y μια συμβολοσειρά. Λέμε ότι υπάρχει μια εμφάνιση της x στη y, ή, πιο απλά, ότι η x εμφανίζεται στη y, όταν η x είναι μια υποσυμβολοσειρά της y.

Κάθε εμφάνιση της x μπορεί να χαρακτηριστεί από μια θέση στη y. Έτσι λέμε ότι η x εμφανίζεται στη *θέση εκκίνησης* i στη y όταν $y[i \,..\, i + |x| - 1] = x$. Είναι μερικές φορές ευκολότερο να χρησιμοποιούμε τη *θέση κατάληξης* $i + |x| - 1$.

Παράδειγμα
Έστω ότι έχουμε τις $x = \texttt{αβα}$ και $y = \texttt{βαβααβαβα}$. Οι θέσεις εκκίνησης και κατάληξης, όπου η x εμφανίζεται στη y, είναι

i	0	1	2	3	4	5	6	7	8
$y[i]$	β	α	β	α	α	β	α	β	α
θέσεις εκκίνησης		1			4		6		
θέσεις κατάληξης				3			6		8

2.2 Ευθυγράμμιση συμβολοσειρών

Σε αυτήν την ενότητα, εισάγουμε τις έννοιες της απόστασης μεταξύ συμβολοσειρών, των λειτουργιών επεξεργασίας συμβολοσειρών, και της ευθυγράμμισης συμβολοσειρών από το [9].

M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on Strings*. Cambridge University Press, USA, 2007

Μας ενδιαφέρει η έννοια της ομοιότητας μεταξύ δύο συμβολοσειρών x και y μήκους m και n, αντίστοιχα, ή κατά ανάλογο τρόπο, της απόστασης μεταξύ αυτών των δύο συμβολοσειρών.

Ορισμός Απόσταση μεταξύ δύο συμβολοσειρών)
Λέμε ότι μια συνάρτηση $\delta : \Sigma^* \times \Sigma^* \to \mathbb{R}$ είναι μια απόσταση στο Σ^* αν και οι τέσσερεις προϋποθέσεις ικανοποιούνται, για κάθε $u, v \in \Sigma^*$:

- Θετικότητα*: $\delta(u, v) \geq 0$*
- Διαχωρισμός*: $\delta(u, v) = 0$ αν και μόνον αν $u = v$*

- Συμμετρία: $\delta(u, v) = \delta(v, u)$
- Τριγωνική ανισότητα: $\delta(u, v) \leq \delta(u, w) + \delta(w, v)$, *για κάθε* $w \in \Sigma^*$

Οι αποστάσεις ορίζονται και ως λειτουργίες οι οποίες μπορούν να μετατρέψουν τη x στη y. Υπάρχουν τρεις τύποι βασικών λειτουργιών. Ονομάζονται *λειτουργίες επεξεργασίας*:

- *αντικατάσταση* ενός συμβόλου της x σε μια δοθείσα θέση από ένα γράμμα της y
- *αφαίρεση* ενός συμβόλου της x σε μια δοθείσα θέση
- *πρόσθεση* ενός συμβόλου της y στη x σε μια δοθείσα θέση

Το κόστος *cost*, το οποίο έχει θετική ακέραια τιμή, σχετίζεται με κάθε λειτουργία ξεχωριστά. Για $\alpha, \beta \in \Sigma$, συμβολίζουμε ως:

- $\mathsf{sub}(\alpha, \beta)$: το κόστος αντικατάστασης με το σύμβολο β του συμβόλου α
- $\mathsf{del}(\alpha)$: το κόστος αφαίρεσης του συμβόλου α
- $\mathsf{ins}(\beta)$: το κόστος πρόσθεσης του συμβόλου β

Σε αυτήν την εργασία, υποθέτουμε ότι αυτά τα κόστη είναι ανεξάρτητα από τις θέσεις στις οποίες εφαρμόζονται οι αντίστοιχες λειτουργίες και ότι ισχύουν τα μοναδιαία κόστη ως εξής: $\mathsf{sub}(\alpha, \beta) = \mathsf{del}(\beta, \alpha) = \mathsf{del}(\alpha) = \mathsf{ins}(\beta) = 1$, όπου $\alpha, \beta \in \Sigma$.

Ορισμός Απόσταση επεξεργασίας)
Από τα μοναδιαία κόστη θέτουμε

$$\delta_E = \min\{\textit{κόστος του } \sigma : \sigma \in S_{x,y}\}$$

όπου $S_{x,y}$ *είναι το σύνολο των διαδοχικών βασικών λειτουργιών επεξεργασίας για τη μετατροπή της* x *στη* y *και το κόστος ενός στοιχείου* $\sigma \in S_{x,y}$ *είναι το άθροισμα των κόστων των λειτουργιών επεξεργασίας της ακολουθίας* σ*. Η συνάρτηση* δ_E *είναι τότε μια απόσταση στο* Σ^* *και καλείται* απόσταση επεξεργασίας.

Ορισμός Απόσταση Χάμινγκ)
Η απόσταση Χάμινγκ*, η οποία συμβολίζεται ως* δ_H*, ορίζεται για δύο συμβολοσειρές του ιδίου μήκους ως ο αριθμός των θέσεων στις οποίες οι δύο συμβολοσειρές έχουν διαφορετικά σύμβολα. Η απόσταση Χάμινγκ είναι ένα συγκεκριμένο είδος απόστασης επεξεργασίας κατά το οποίο επιτρέπεται μόνο η λειτουργία της αντικατάστασης. Αυτό ισοδυναμεί με το να θέσουμε* $\mathsf{del}(\alpha) = \mathsf{inf}(\alpha) = +\infty$*, για κάθε* $\alpha \in \Sigma$.

Μια *ευθυγράμμιση* μεταξύ δύο συμβολοσειρών $x, y \in \Sigma^*$, των οποίων τα μήκη είναι m και n, αντίστοιχα, είναι ένας τρόπος οπτικοποίησης της ομοιότητας τους.

Ορισμός Ευθυγράμμιση μεταξύ δύο συμβολοσειρών)
Μια ευθυγράμμιση μεταξύ δύο συμβολοσειρών x και y είναι μια συμβολοσειρά z στο

$$(\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\}) \setminus (\{\varepsilon,\varepsilon\})$$

της οποίας η προβολή στο πρώτο συστατικό είναι η x, και η προβολή στο δεύτερο συστατικό είναι η y. Έτσι, αν η z είναι μια ευθυγράμμιση μήκους p μεταξυ των x και y, έχουμε ότι

$$z = (x'_0, y'_0)(x'_1, y'_1) \dots (x'_{p-1}, y'_{p-1})$$
$$x = x'_0 x'_1 \dots x'_{p-1}$$
$$y = y'_0 y'_1 \dots y'_{p-1}$$

όπου $x'_i \in \Sigma \cup \{\varepsilon\}$ και $y'_i \in \Sigma \cup \{\varepsilon\}$, για κάθε $0 \leq i < p$.

Μια ευθυγράμμιση μπορεί να συμβολιστεί και ως

$$\begin{pmatrix} x'_0 & x'_1 & \dots & x'_{p-1} \\ y'_0 & y'_1 & \dots & y'_{p-1} \end{pmatrix}$$

Ένα *ζευγάρι ευθυγράμμισης* του τύπου (α, β), όπου $\alpha, \beta \in \Sigma$, δηλώνει την αντικατάσταση με το σύμβολου β του συμβόλου α, ή, ανάλογα, τη *διαφορά* μεταξύ του συμβόλου α και του συμβόλου β. Ένα *ζευγάρι ευθυγράμμισης* του τύπου (α, ε), όπου $\alpha \in \Sigma$, δηλώνει την αφαίρεση του συμβόλου α. Τέλος, ένα *ζευγάρι ευθυγράμμισης* (ε, β), όπου $\beta \in \Sigma$, δηλώνει την πρόσθεση του συμβόλου β.

Ορισμός Κόστος ευθυγράμμισης)
Ορίζουμε το κόστος ενός ζευγαριού ευθυγράμμισης ως

- $\mathit{cost}(\alpha, \beta) = \mathit{sub}(\alpha, \beta)$
- $\mathit{cost}(\alpha, \varepsilon) = \mathit{del}(\alpha)$
- $\mathit{cost}(\varepsilon, \beta) = \mathit{ins}(\beta)$

όπου $\alpha, \beta \in \Sigma$. Το κόστος μιας ευθυγράμμισης ορίζεται ως το άθροισμα των κόστων που σχετίζονται με το κάθε ζευγάρι ευθυγράμμισης που την αποτελούν.

Παράδειγμα
Έστω ότι έχουμε τις συμβολοσειρές $x =$ `ACGA` *και* $y =$ `ATGCTA`*. Μια ευθυγράμμιση μεταξύ των* x *και* y *είναι*

$$\begin{pmatrix} ACG{-}{-}A \\ ATGCTA \end{pmatrix}$$

Δοθέντων των παρακάτω ζευγαριών ευθυγράμμισης

Λειτουργία	*Ζευγάρι ευθυγράμμισης*	*Cost*
αντικατάσταση με `A` *του* `A`	*(*`A`*,*`A`*)*	*0*
αντικατάσταση με `T` *του* `C`	*(*`C`*,*`T`*)*	*1*
αντικατάσταση με `G` *του* `G`	*(*`G`*,*`G`*)*	*0*
πρόσθεση του `C`	*(-,*`C`*)*	*1*
πρόσθεση του `T`	*(-,*`T`*)*	*1*
αντικατάσταση με `A` *του* `A`	*(*`A`*,*`A`*)*	*0*

αυτή η ευθυγράμμιση είναι βέλτιστη εφόσον το κόστος της, $0+1+0+1+1+0=3$*, ισούται με την απόσταση επεξεργασίας των δύο συμβολοσειρών.*

2.3 Ασυμπτωτική πολυπλοκότητα

Σε αυτήν την ενότητα, εισάγουμε την έννοια της ασυμπτωτικής πολυπλοκότητας από το [7].

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition.* The MIT Press, 2001

Η ασυμπτωτική πολυπλοκότητα εφαρμόζεται σε συναρτήσεις,ο ι οποίες συνήθως χαρακτηρίζουν το χρόνο εκτέλεσης των αλγορίθμων. Ωστόσο η ασυμπτωτική πολυπλοκότητα μπορεί να εφαρμοστεί σε συναρτήσεις που χαρακτηρίζουν άλλες πτυχές του αλγορίθμου όπως η χωρική πολυπλοκότητα.
Έστω ότι έχουμε δύο συναρτήσεις $f : \mathbb{N} \to \mathbb{N}$ και $g : \mathbb{N} \to \mathbb{N}$.

Ορισμός Θ-συμβολισμός)
Συμβολίζουμε ως $\Theta(g(n))$ *το σύνολο των συναρτήσεων*

$$\Theta(g(n)) = \{f(n) : \textit{υπάρχουν θετικές σταθερές } c_1, c_2, \textit{ και } n_0 \in \mathbb{N} :$$

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \ \forall \ n \geq n_0\}$$

Ορισμός $\mathcal{O}$-συμβολισμός)
Συμβολίζουμε ως $\mathcal{O}(g(n))$ *το σύνολο των συναρτήσεων*

$$\mathcal{O}(g(n)) = \{f(n) : \textit{υπάρχει θετική σταθερά } c \textit{ και } n_0 \in \mathbb{N} :$$

$$0 \leq f(n) \leq cg(n) \ \forall \ n \geq n_0\}$$

Η συνάρτηση f είναι *γραμμική* αν $f(n) = \Theta(n)$, *τετραγωνική* αν $f(n) = \Theta(n^2)$, *κυβική* αν $f(n) = \Theta(n^3)$, *λογαριθμική* αν $f(n) = \Theta(\log n)$, ή *εκθετική* αν υπάρχει $a > 0$ για το οποίο $f(n) = \Theta(a^n)$.

2.4 Βασικές δομές δεδομένων

Σε αυτήν την ενότητα, εισάγουμε βασικές δομές δεδομένω από το [7].

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, 2001

Τα σύνολα είναι θεμελιώδη στην επιστήμη της πληροφορικής όπως είναι και στα μαθηματικά. Ενώ στα μαθηματικά τα σύνολα μπορεί να είναι άπειρα και είναι αναλλοίωτα, τα σύνολα που χρησιμοποιούνται στους αλγορίθμους είναι συνήθως πεπερασμένα και μπορούν να μεγαλώσουν, να μικρύνουν και να αλλάξουν με την πάροδο του χρόνου. Τέτοια σύνολα λέγονται *δυναμικά*.

Σε μια τυπική υλοποίηση ενός δυναμικού συνόλου, κάθε στοιχείο του συνόλου αναπαρίσταται από ένα αντικείμενο του οποίου τα ορίσματα είναι μια *πλειάδα* $(a_0, a_1, \ldots, a_n)$, όπου $n \geq 0$, που μπορεί να επεξεργασθεί αν έχουμε ένα δείκτη στο αντικείμενο.

Πιο κάτω παραθέτουμε μια λίστα τυπικών λειτουργιών ενός δυναμικού συνόλου:

- empty(): μια λειτουργία η οποία δημιουργεί και επιστρέφει ένα κενό σύνολο
- is-empty(S): μια λειτουργία η οποία επιστρέφει true αν το σύνολο S είναι κενό και false διαφορετικά
- search(S, k): μια λειτουργία αναζήτησης η οποία, δοθέντος ενός συνόλου S και μιας τιμής κλειδιού k, επιστρέφει ένα δείκτη x σε κάποιο στοιχείο του S, έτσι ώστε $x.key = k$, ή nil αν δεν υπάρχει τέτοιο στοιχείο που να ανήκει στο S
- insert(S, x): μια λειτουργία μετατροπής η οποία αυξάνει το σύνολο S με το στοιχείο στο οποίο δείχνει ο δείκτης x

- delete(S, x): μια λειτουργία μετατροπής η οποία, δοθέντος ενός δείκτη x σε ένα στοιχείο του συνόλου S, διαγράφει το στοιχείο από το S

2.4.1 Στοίβες και ουρές

Οι στοίβες και οι ουρές είναι δυναμικά σύνολα στα οποία το στοιχείο που διαγράφεται από το σύνολο με τη λειτουργία delete είναι προκαθορισμένο.

Στην στοίβα το στοιχείο που διαγράφεται από το σύνολο είναι αυτό που έχει προστεθεί πιο πρόσφατα: η στοίβα υλοποιεί την πολιτική του *last-in first-out*. Η λειτουργία insert σε μια στοίβα καλείται push (Algorithm PUSH(S, x)), και η λειτουργία delete, η οποία δεν παίρνει καμία παράμετρο, καλείται pop (Algorithm POP(S)). Μπορούμε να υλοποιήσουμε μια στοίβα με το πολύ n στοιχεία με έναν πίνακα S$[0\,..\,n-1]$. Ο πίνακας έχει μια παράμετρο S.top η οποία δείχνει στο στοιχείο που έχει προστεθεί πιο πρόσφατα. Η στοίβα αποτελείται από τα στοιχεία S$[0\,..$ S.$top]$, όπου το S$[0]$ είναι το στοιχείο στο κάτω μέρος της στοίβας και το S[S.top] είναι το στοιχείο στο πάνω μέρος. Οι λειτουργίες empty και is-empty σε μια στοίβα καλούνται empty-stack και is-stack-empty (Algorithm IS-STACK-EMPTY(S)), αντίστοιχα. Αν κάνουμε pop σε μια άδεια στοίβα τότε λέμε ότι η στοίβα υποχειλίζει (*underflows*).Αν S.top υπερβαίνει το $n-1$ (Algorithm IS-STACK-FULL(S)), τότε λέμε ότι η στοίβα υπερχειλίζει (*overflows*). Κάθε μια από τις λειτουργίες της στοίβας απαιτεί σταθερό χρόνο εκτέλεσης.

ALGORITHM IS-STACK-EMPTY(S)
1: **if** S.$top = -1$ **then**
2: **return** true;
3: **else**
4: **return** false;

ALGORITHM IS-STACK-FULL(S)
1: **if** S.$top = n - 1$ **then**
2: **return** true;
3: **else**
4: **return** false;

Παρομοίως, σε μια *ουρά*, το στοιχείο που διαγράφεται από το σύνολο είναι πάντοτε αυτό που υπάρχει στο σύνολο τον περισσότερο χρόνο:η ουρά υλοποιεί την πολιτική *first-in first-out*. Η ουρά έχει το *head* και το *tail*. Όταν

ALGORITHM PUSH(S, x)
1: **if** IS-STACK-FULL(S) **then**
2: **return** overflow;
3: **else**
4: S.$top \leftarrow$ S.$top + 1$;
5: S[S.top] $\leftarrow x$;

ALGORITHM POP(S)
1: **if** IS-STACK-EMPTY(S) **then**
2: **return** underflow;
3: **else**
4: S.$top \leftarrow$ S.$top - 1$;
5: **return** S[S.$top + 1$];

ένα στοιχείο προστίθεται, παίρνει τη θέση στο *tail* της ουράς. Το στοιχείο που διαγράφεται είναι πάντοτε αυτό που βρίσκεται στο *head* της ουράς.Η λειτουργία insert σε μια ουρά καλείται enqueue (Algorithm ENQUEUE(Q, x)), και η λειτουργία delete, η οποία δεν παίρνει καμία παράμετρο, καλείται dequeue (Algorithm DEQUEUE(Q)). Μπορούμε να υλοποιήσουμε μια ουρά με το πολύ $n-1$ στοιχεία με έναν πίνακα Q$[0\,..\,n-1]$. Η ουρά έχει μια παράμετρο Q.*head* που δείχνει στο *head*. Η παράμετρος Q.*tail* δείχνει την επόμενη θέση στην οποία ένα νέο στοιχείο θα προστεθεί στην ουρά. Τα στοιχεία στην ουρά βρίσκονται στις θέσεις Q.*head*, Q.*head* $+ 1, \ldots,$ Q.*tail* $- 1$, όπου μπορούμε να τα προσπελάσουμε κυκλικά. Δηλαδή μετά τη θέση 0 αμέσως ακολουθεί η θέση $n-1$. Αν κάνουμε dequeue σε μια άδεια ουρά τότε λέμε ότι η ουρά υποχειλίζει (*underflows*).Αν Q.*head* = Q.*tail* $+ 1$ τότε η ουρά είναι γεμάτη (Algorithm IS-QUEUE-FULL(Q)), και μια προσπάθεια να προσθέσουμε ένα στοιχείο προκαλεί υπερχείλιση. Ο ψευδοκώδικας υποθέτει ότι $n =$ Q.*length*. Κάθε μια από τις λειτουργίες της ουράς απαιτεί σταθερό χρόνο εκτέλεσης.

ALGORITHM IS-QUEUE-EMPTY(Q)
1: **if** Q.*tail* = Q.*head* **then**
2: **return** true;
3: **else**
4: **return** false;

```
ALGORITHM Is-Queue-Full(Q)
1: if Q.head = Q.tail + 1 then
2:    return true;
3: else
4:    return false;
```

```
ALGORITHM Enqueue(Q, x)
1: if Is-Queue-Full(S) then
2:    return overflow;
3: else
4:    Q[Q.tail] ← x;
5:    if Q.tail = Q.length − 1 then
6:       Q.tail ← 0;
7:    else
8:       Q.tail ← Q.tail + 1;
```

```
ALGORITHM Dequeue(Q)
1: if Is-Queue-Empty(S) then
2:    return underflow;
3: else
4:    x ← Q[Q.head];
5:    if Q.head = Q.length − 1 then
6:       Q.head ← 0;
7:    else
8:       Q.head ← Q.head + 1;
9:    return x;
```

2.4.2 Συνδεδεμένες λίστες

Μια *συνδεδεμένη λίστα* είναι μια δομή δεδομένων στην οποία τα στοιχεία τοποθετούνται με γραμμική σειρά. Σε αντίθεση με τους πίνακες στους οποίους αυτή η γραμμική σειρά προσδιορίζεται από τους δείκτες του πίνακα, η σειρά σε μια συνδεδεμένη λίστα προσδιορίζεται από το δείκτη στο κάθε στοιχείο. Το iοστό στοιχείο της λίστας L συμβολίζεται ως $\mathsf{L}(i)$. Κάθε στοιχείο μιας *διπλά-συνδεδεμένης λίστας* είναι ένα στοιχείο του οποίου τα ορίσματα είναι μια *πλειάδα* $(a_0, a_1, \ldots, a_n)$, τέτοια ώστε $n \geq 0$, και δύο άλλοι παράμετροι: $next$ και $prev$. Δοθέντος ενός στοιχείου x σε μια λίστα, ο δείκτης $x.next$ δείχνει στο επόμενο στοιχείο της λίστας, και ο δείκτης $x.prev$ στο προηγούμενο.Αν $x.prev = \mathsf{nil}$, το στοιχείο x δεν ακολουθεί κανένα στοιχείο. Είναι δηλαδή το πρώτο στοιχείο, ή το *head* (κεφάλι), της λίστας.Αν $x.next = \mathsf{nil}$, το στοιχείο x δεν ακολουθείται από κανένα στοιχείο. Είναι δηλαδή το τελευταίο στοιχείο, ή το *tail* (ουρά) , της λίστας. Αν μια λίστα είναι *μονά-συνδεδεμένη* (βλέπε το Σχήμα 2.1), παραβλέπουμε το δείκτη $prev$ σε κάθε στοιχείο. Η παράμετρος $\mathsf{L}.head$ δείχνει το πρώτο στοιχείο της λίστας.Αν $\mathsf{L}.head = \mathsf{nil}$, τότε η λίστα είναι κενή.

Η λειτουργία search σε μια λίστα καλείται list-search (Algorithm LIST-SEARCH(L, k)). Βρίσκει το στοιχείο με κλειδί k, όπου k είναι μια παράμετρος, ή ένα υποσύνολο παραμέτρων, της πλειάδας $(a_0, a_1, \ldots, a_n)$, στη λίστα L με μια απλή γραμμική αναζήτηση, επιστρέφοντας ένα δείκτη στο στοιχείο αυτό.Αν δεν υπάρχει τέτοιο στοιχείο με κλειδί k στη λίστα, τότε επιστρέφει nil. Απαιτεί γραμμικό χρόνο εκτέλεσης.

ALGORITHM LIST-SEARCH(L, k)
1: $x \leftarrow \mathsf{L}.head$;
2: **while** $x \neq \mathsf{nil}$ **and** $x.key \neq k$ **do**
3: $\quad x \leftarrow x.next$;
4: **return** x;

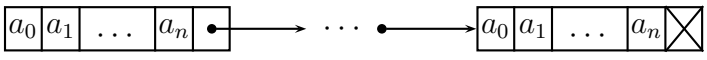
Η λειτουργία insert σε μια λίστα καλείται list-insert (Algorithm LIST-INSERT(L, x)). Προσθέτει το στοιχείο x στο μπροστινό μέρος της λίστας L. Απαιτεί σταθερό χρόνο εκτέλεσης.

Η λειτουργία delete σε μια λίστα καλείται list-delete (Algorithm LIST-DELETE(L, x)). Δοθέντος ενός δείκτη στο στοιχείο x, διαγράφει το x από τη λίστα L ενημερώνοντας τους δείκτες. Απαιτεί σταθερό χρόνο εκτέλεσης.

Οι λειτουργίες empty και is-empty σε μια συνδεδεμένη λίστα καλούνται empty-list και is-list-empty, αντίστοιχα.

ALGORITHM LIST-INSERT(L, x)
1: $x.next \leftarrow \mathsf{L}.head$;
2: **if** $\mathsf{L}.head \neq$ nil **then**
3: $\quad \mathsf{L}.head.prev \leftarrow x$;
4: $\mathsf{L}.head \leftarrow x$;
5: $x.prev \leftarrow$ nil;

ALGORITHM LIST-DELETE(L, x)
1: **if** $x.prev \neq$ nil **then**
2: $\quad x.prev.next \leftarrow x.next$;
3: **else**
4: $\quad \mathsf{L}.head \leftarrow x.next$;
5: **if** $x.next \neq$ nil **then**
6: $\quad x.next.prev \leftarrow x.prev$;



Σχήμα2.1 Μονά-συνδεδεμένη λίστα

Κεφάλαιο 3

Ευθυγράμμιση γενετικών δεδομένων νέας γενιάς σε ένα γονιδίωμα

Το υλικό που παρουσιάζουμε σε αυτό το κεφάλαιο έχει δημοσιευτεί στο [14].

Kimon Frousios, Costas S. Iliopoulos, Laurent Mouchard, Solon P. Pissis, and German Tischler. REAL: an efficient REad ALigner for next generation sequencing reads. In Aidong Zhang, Mark Borodovsky, Gultekin Özsoyoglu, and Armin R. Mikler, editors, *Proceedings of the first ACM International Conference on Bioinformatics and Computational Biology (BCB 2011)*, pages 154–159, USA, 2010. ACM

Σε αυτό το κεφάλαιο, παρουσιάζουμε το REad ALigner (REAL) [14], ένα αποτελεσματικό, ευαίσθητο και ακριβές πρόγραμμα ευθυγράμμισης short reads, για την αντιστοίχιση εκατομμυρίων short reads σε ένα γονιδίωμα. Το REAL υιοθετεί τακτικές παρόμοιες με αυτές που παρουσιάστηκαν στο [2] και [3], για ακριβή και προσεγγιστική αντιστοίχιση ακολουθιών, αντίστοιχα. Το REAL προ-επεξεργάζεται την ακολουθία αναφοράς, βασιζόμενο στο μήκος των short reads, χρησιμοποιώντας κωδικοποίηση με δύο bits ανά βάση του αλφάβητου του DNA, λειτουργίες σε επίπεδο λέξεων και ταξινόμηση, για να δημιουργήσει ένα ευρετήριο της ακολουθίας στην κυρίως μνήμη. Μετά μετατρέπει κάθε short read σε μια μοναδική αριθμητική τιμή, χρησιμοποιώντας κωδικοποίηση με δύο bits ανά βάση του αλφάβητου του DNA, και, τέλος, χρησιμοποιεί τη γνωστή τακτική του διαχωρισμού σε ακριβή τμήματα [37], τη δυαδική αναζήτηση, και λειτουργίες σε επίπεδο λέξεων για να επιτύχει την ευθυγράμμιση.

Συμβολίζουμε τα παραγόμενα short reads με το σύνολο $\{p_1, p_2, \ldots, p_r\}$, όπου συνήθως $r > 10^6$, και τα αποκαλούμε μοτίβα (patterns). Το μήκος m κάθε μοτίβου που παράγεται από τα συστήματα Solexa της Illumina, είναι συνήθως ανάμεσα σε 25 και 150 bp. Συμβολίζουμε την ακολουθία αναφοράς μήκους n, συνήθως $n > 10^6$, με t, και την αποκαλούμε κείμενο (text).

Ορίζουμε το πρόβλημα της ευθυγράμμισης εκατομμυρίων short reads σε μια ακολουθία αναφοράς, ως εξής.

Πρόβλημα Ευθυγράμμιση short reads)
Βρείτε κατά πόσο το μοτίβο $p_i = p_i[0\,..\,m-1]$, για κάθε $1 \leq i \leq r$, όπου $p_i \in \Sigma^+$, $\Sigma = \{\texttt{A,C,G,T}\}$, εμφανίζεται, με το πολύ k-αναντιστοιχίες, στο κείμενο $t = t[0\,..\,n-1]$, όπου $t \in \Sigma^+$.

3.1 Αλγόριθμος

Χρησιμοποιούμε λειτουργίες σε επίπεδο λέξεων για την προεπεξεργασία του κειμένου t, μετατρέποντας την κάθε υποσυμβολοσειρά μήκους m του t σε μια υπογραφή. Παίρνουμε την υπογραφή $\sigma(x)$, όπου $\sigma : \Sigma^+ \to \mathbb{Z}^+$, μιας μη κενής

συμβολοσειράς x, μετατρέποντας τη x σε μια μοναδική αριθμητική τιμή χρησιμοποιώντας κωδικοποίηση με δύο bits ανά βάση του αλφάβητου $\Sigma = \{\texttt{A,C,G,T}\}$ (βλέπε το σχετικό πίνακα 3.1).

Πίνακας3.1 Κωδικοποίηση με δύο bits ανά βάση του αλφάβητου $\Sigma = \{\texttt{A,C,G,T}\}$

Γράμμα	Δυαδική αναπαράσταση
A	00
C	01
G	10
T	11

Παράδειγμα
Η υπογραφή $\sigma(x)$ *της συμβολοσειράς* $x =$ ***AGCAT*** *είναι* 0010010011.

Παράδειγμα
Η υπογραφή $\sigma(x)$ *της συμβολοσειράς* $x =$ ***TACGC*** *είναι* 1100011001.

Παράδειγμα
Η υπογραφή $\sigma(x)$ *της συμβολοσειράς* $x =$ ***GGGTCTA*** *είναι* 10101011011100.

Υλοποιούμε την κατά προσέγγιση αντιστοίχιση δύο συμβολοσειρών με την τακτική του διαχωρισμού σε ακριβή τμήματα. Θεωρούμε την μη κενή συμβολοσειρά x ως το μοτίβο και τη μη κενή συμβολοσειρά y ως το κείμενο. Θέλουμε να βρούμε εμφανίσεις του x στο y με το πολύ k-αναντιστοιχίες. Η τακτική του διαχωρισμού σε ακριβή τμήματα ορίζεται ως ακολούθως.

Διαχωρίζουμε το x σε ένα σύνολο $\{x_0, \ldots, x_{q-1}\}$ τμημάτων $q > k$, έτσι ώστε το $x_i \in \Sigma^+$, για κάθε $0 \leq i < q$, και κατασκευάζουμε τις λίστες X_i που περιέχουν τις θέσεις εμφάνισης του x_i στο y. Για κάθε μια από τις πιθανότητες επιλογής $q - k$ από τα q τμήματα, συγχωνεύουμε τις σχετικές λίστες θέσεων χρησιμοποιώντας τα ανάλογα offset των θέσεων. Αυτό μας παρέχει $\binom{q}{q-k}$ λίστες με υποψήφιες θέσεις εμφάνισης. Η ένωση X αυτών των συγχωνευμένων λιστών αποτελεί ένα υπερσύνολο των θέσεων εμφάνισης του x στο y με το πολύ k-αναντιστοιχίες. Κατασκευάζουμε τη λίστα εμφάνισης του x στο y φιλτράροντας το X με τη χρήση ενός αλγορίθμου για να διαπιστωθεί εάν οι υποψήφιες θέσεις υπάρχουν με το πολύ k-αναντιστοιχίες.

Παράδειγμα
Σκεφτείτε την αναζήτηση για ένα μοτίβο x στο κείμενο y με το πολύ μια αναντιστοιχία. Εμείς διαχωρίζουμε το x σε τρία τμήματα: x_0, x_1 και x_2. Πρέπει να εξετάσουμε τρία ζεύγη τμημάτων: (x_0, x_1), (x_1, x_2) και (x_0, x_2). Οι δύο πρώτοι συνδυασμοί μπορούν να βρεθούν εύκολα χρησιμοποιώντας ένα ευρετήριο για το y, δηλαδή χρειαζόμαστε μόνο να αναζητήσουμε τα μοτίβα x_0x_1 και x_1x_2. Το τρίτο απαιτεί τη συγχώνευση των λιστών με τις υποψήφιες θέσεις εμφάνισης.

Ως εκ τούτου, έχουμε λάβει το ακόλουθο βοηθητικό λήμμα.

Λήμμα
Δοθέντος ενός συνόλου $\{x_0, \ldots, x_{q-1}\}$ από q τμήματα, όπου $x_i \in \Sigma^+$, για κάθε $0 \leq i < q$, μιας μη κενής συμβολοσειράς x, και του μέγιστου επιτρεπόμενου αριθμού αναντιστοιχιών $k < q$, οποιεσδήποτε από τις k-αναντιστοιχίες δεν μπορούν να υπάρξουν την ίδια στιγμή σε τουλάχιστον $q - k$ τμήματα του x.

Απόδειξη Απευθείας από την αρχή του περιστερώνα—αν n αντικείμενα πρέπει να τοποθετηθούν σε m θέσεις, όπου $n > m$, τότε τουλάχιστον μία θέση πρέπει να περιέχει περισσότερα από ένα αντικείμενα.

Χωρίς βλάβη της γενικότητας, θέτουμε $q - k = k$ και $\lceil \frac{m \log |\Sigma|}{q-k} \rceil \leq w$, όπου w είναι το μέγεθος της λέξης του υπολογιστή. Διαχωρίζουμε τη δυαδική αναπαράσταση της $\sigma(x)$, σε ένα σύνολο $\{\sigma(x_0), \sigma(x_1), \ldots, \sigma(x_{q-1})\}$ των $q > k$ τμήματα, όπου $\sigma(x_i) \in \mathbb{Z}^+$, για κάθε $0 \leq i < q$.

Συμβολίζουμε με $C_j(\sigma(x)) = \{\sigma(x_{a_0}), \sigma(x_{a_1}), \ldots, \sigma(x_{a_{q-k-1}})\}$, όπου $a_1 < a_2 < \ldots < a_{q-k-1}$, τους $\binom{q}{q-k}$ πιθανούς συνδυασμούς των $q - k$ τμημάτων του

$\{\sigma(x_0), \sigma(x_1), \ldots, \sigma(x_{q-1})\}$, έτσι ώστε αν

$C_{j+1}(\sigma(x)) = \{\sigma(x_{b_0}), \sigma(x_{b_1}), \ldots, \sigma(x_{b_{q-k-1}})\}$, τότε

$\sum_{i=0}^{q-k-1} a_i \leq \sum_{i=0}^{q-k-1} b_i$, για κάθε $0 \leq j < \binom{q}{q-k} - 1$

Συμβολίζουμε με $D_j(\sigma(x)) = \{\sigma(x_{a_0}), \sigma(x_{a_1}), \ldots, \sigma(x_{a_{k-1}})\}$, όπου $a_0 < a_1 < \ldots < a_{k-1}$, τους $\binom{q}{q-k}$ πιθανούς συνδυασμούς των υπόλοιπων k τμημάτων του

$\{\sigma(x_0), \sigma(x_1), \ldots, \sigma(x_{q-1})\}$, έτσι ώστε αν

$D_{j+1}(\sigma(x)) = \{\sigma(x_{b_0}), \sigma(x_{b_1}), \ldots, \sigma(x_{b_{k-1}})\}$, τότε

$$\sum_{i=0}^{k-1} a_i \leq \sum_{i=0}^{k-1} b_i, \text{ για κάθε } 0 \leq j < \binom{q}{q-k} - 1$$

Παράδειγμα
Έστω ότι έχουμε $x =$ ***ACCGATCA***, $q = 4$, *και* $k = 2$. *Αν διαχωρίσουμε τη* $\sigma(x) = 0001011000110100$ *σε ένα σύνολο* $\{0001, 0110, 0011, 0100\}$ *των* $q = 4$ *τμημάτων της* x, *τότε* $C_0 = \{0001, 0110\}$, $D_0 = \{0001, 0110\}$, $C_1 = \{0001, 0011\}$, $D_1 = \{0001, 0011\}$, $C_2 = \{0001, 0100\}$, $D_2 = \{0001, 0100\}$, $C_3 = \{0110, 0011\}$, $D_3 = \{0110, 0011\}$, $C_4 = \{0110, 0100\}$, $D_4 = \{0110, 0100\}$, $C_5 = \{0011, 0100\}$, $D_5 = \{0011, 0100\}$.

Παράδειγμα
Έστω ότι έχουμε $x =$ ***GGGTCTAGT***, $q = 3$, *και* $k = 2$. *Αν διαχωρίσουμε τη* $\sigma(x) = 101010110111001011$ *σε ένα σύνολο* $\{101010, 110111, 001011\}$ *των* $q = 3$ *τμημάτων της* x, *τότε* $C_0 = \{101010\}$, $D_0 = \{101010, 110111\}$, $C_1 = \{110111\}$, $D_1 = \{101010, 001011\}$, $C_2 = \{001011\}$, $D_2 = \{110111, 001011\}$.

Ο σκοπός μας είναι να προεπεξεργαστούμε το t και να κατασκευάσουμε ένα σύνολο λιστών X_j, για κάθε $0 \leq j < \binom{q}{q-k}$. Κάθε X_j κρατά την πλοιάδα $e^j = (u, s, NF)$, όπου το $e^j.u$ αναπαριστά τη θέση εκκίνησης της υποσυμβολοσειράς x του t, το $e^j.s$ αναπαριστά την παράθεση των υπογραφών των τμημάτων $C_j(\sigma(x))$, και το $e^j.NF$ είναι ένας δείκτης στην πλοιάδα $e^l = (u, s, NF)$ στην X_l, όπου το $e^l.s$ αναπαριστά την παράθεση των υπόλοιπων τμημάτων στο $D_l(\sigma(x))$.

Ορίζουμε τις πιο κάτω λειτουργίες:

- $\texttt{f}(q, k, j)$: μια λειτουργία η οποία, δοθέντων των ακεραίων q, k, και j, επιστρέφει έναν ακέραιο l, έτσι ώστε

$$C_j(\sigma(x)) = \{\sigma(x_{a_0}), \sigma(x_{a_1}), \ldots, \sigma(x_{a_{q-k-1}})\} \text{ και}$$

$$D_l(\sigma(x)) = \{\sigma(x_{b_0}), \sigma(x_{b_1}), \ldots, \sigma(x_{b_{k-1}})\}, \text{ τότε}$$

$$C_j(\sigma(x)) \cup D_l(\sigma(x)) = \{\sigma(x_0), \sigma(x_1), \ldots, \sigma(x_{q-1})\}$$

  για την υπογραφή $\sigma(x)$ της συμβολοσειράς x.

- $\texttt{conc}(C)$: μια λειτουργία η οποία, δοθέντος του συνόλου $C = \{\sigma(x_0), \sigma(x_1), \ldots, \sigma(x_{q-1})\}$ των q τμημάτων, έτσι ώστε $\sigma(x_i) \in \mathbb{Z}^+$, για κάθε $0 \leq i < q$, της υπογραφής $\sigma(x)$, της συμβολοσειράς x, επιστρέφει τη $\sigma(x)$.

- `bs`$(X,\sigma(x))$: μια λειτουργία δυαδικής αναζήτησης η οποία, δοθέντος της ταξινομημένης λίστας X των πλειάδων και μιας υπογραφής $\sigma(x)$ της συμβολοσειράς x ως κλειδί, επιστρέφει ένα σύνολο $\{e_{a_0}, e_{a_1}, \ldots, e_{a_{v-1}}\}$ των πλειάδων από τη X, έτσι ώστε $e_{a_i}.s = \sigma(x)$, για κάθε $0 \leq i < v$.

- `bitop`$(\sigma(x), \sigma(y))$: μια λειτουργία σε επίπεδο λέξης μηχανής η οποία, δοθέντος δύο υπογραφών $\sigma(x)$ και $\sigma(y)$ δύο συμβολοσειρών x και y, αντίστοιχα, έτσι ώστε $|x| = |y|$ και $\lceil \log|\Sigma| \rceil |x| \leq w$, επιστρέφει την απόσταση $\delta_H(x, y)$ σε σταθερό χρόνο.

Παράδειγμα
Δοθέντος των $x =$ `ACCGATCA`, $q = 4$, $k = 2$, *και* $j = 0$, *τότε* $f(q, k, j) = 5$ *εφόσον, δοθέντος του συνόλου* $\{0001, 0110, 0011, 0100\}$ *των* $q = 4$ *τμημάτων της* $\sigma(x) = 0001011000110100$ *και των* $C_0 = \{0001, 0110\}$ *και* $D_5 = \{0011, 0100\}$, *ισχύει ότι*

$$C_0 \cup D_5 = \{0001, 0110, 0011, 0100\}$$

Παράδειγμα
Δοθέντος των $x =$ `GGGTCTAGT`, $q = 3$, $k = 2$, *και* $j = 0$, *τότε* $f(q, k, j) = 2$ *εφόσον, δοθέντος του συνόλου* $\{101010, 110111, 001011\}$ *των* $q = 3$ *τμημάτων της* $\sigma(x) = 101010110111001011$ *και των* $C_0 = \{101010\}$ *και* $D_2 = \{110111, 001011\}$, *ισχύει ότι*

$$C_0 \cup D_2 = \{101010, 110111, 001011\}$$

Παράδειγμα
Δοθέντος των $x =$ `ACCGATCA`, $q = 4$, *και* $k = 2$, *τότε* $C_0 = \{0001, 0110\}$ *και* `conc`$(C_0) = 00010110$.

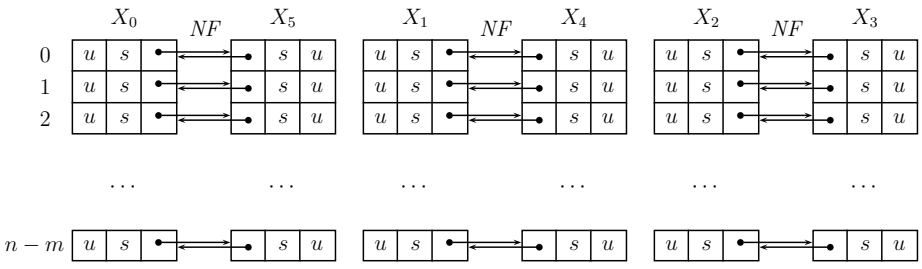
Παράδειγμα
Δοθέντος των $x =$ `GGGTCTAGT`, $q = 3$, *και* $k = 2$, *τότε* $D_2 = \{110111, 001011\}$ *και* `conc`$(D_2) = 110111001011$.

Παράδειγμα
Δοθέντος των $q = 4$, $k = 2$, $|\Sigma| = 4$, *και* $w = 64$, *τότε* $m \leq \frac{w(q-k)}{\log|\Sigma|} = 64$.

Η περιγραφή του προτεινόμενου αλγορίθμου για την επίλυση του Προβλήματος 1 είναι η ακόλουθη:

(ι) Δημιουργουμε τις λιστες

Διαχωρίζουμε το t σε ένα σύνολο $\{t_0, t_1, \ldots, t_{n-m}\}$ υποσυμβολοσειρών, έτσι ώστε $t_i = t[i\,..\,i+m-1]$, για κάθε $0 \leq i \leq n-m$, και υπολογίζουμε τη $\sigma(t_i)$. Ακολούθως υπολογίζουμε τα $C_j(\sigma(t_i))$, $l = \mathtt{f}(q,k,j)$, και $D_l(\sigma(t_i))$, για κάθε $0 \leq j < \binom{q}{q-k}/2$, και προσθέτουμε τις πλειάδες $e^j = (i, \mathtt{conc}(C_j(\sigma(t_i)), NF)$ και $e^l = (i, \mathtt{conc}(D_l(\sigma(t_i)), NF)$ στις λίστες X_j και X_l, αντίστοιχα. Ως αποτέλεσμα, δημιουργούμε ένα σύνολο από $\binom{q}{q-k}$ λίστες(βλ. Σχήμα 3.1).



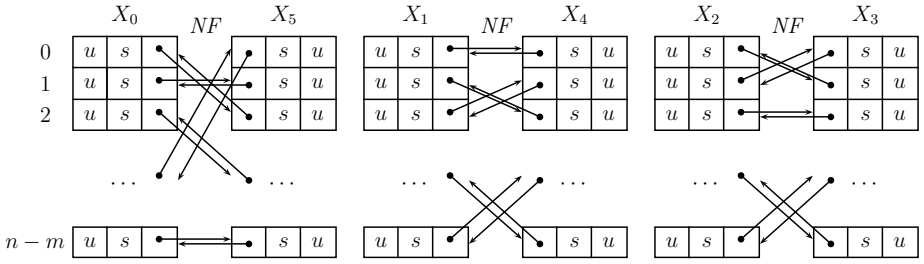
Σχήμα3.1 Λίστα X_j, για κάθε $0 \leq j < \binom{q}{q-k}$, για $q = 4$ και $k = 2$, μετά το Βήμα (I)

(II) Ταξινομουμε τις λιστες

Χρησιμοποιούμε τον αλγόριθμο radix sort για να ταξινομήσουμε τη X_j, για κάθε $0 \leq j < \binom{q}{q-k}$, σύμφωνα με το στοιχείο s –την υπογραφή– των πλειάδων της X_j, βεβαιώνοντας ότι, στην περίπτωση κατά την οποία ανταλλάξουμε τη θέση εμφάνισης δύο πλειάδων, διατηρούμε ότι ο δείκτης $e^j.NF$ συνεχίζει να δείχνει στο e^l, και ότι ο $e^l.NF$ συνεχίζει να δείχνει στο e^j (βλ. Σχήμα 3.2).

(III) Ευθυγραμμιση

Έστω ότι έχουμε ένα μοτίβο p μήκους m. Υπολογίζουμε τα $\sigma(p)$, $C_j(\sigma(p))$, $l = \mathtt{f}(q,k,j)$, και $D_l(\sigma(p))$, για κάθε $0 \leq j < \binom{q}{q-k}$. Στη συνέχει εκτελούμε τη δυαδική αναζήτηση $\mathtt{bs}(X_j, \mathtt{conc}(C_j(\sigma(p))))$, η οποία επιστρέφει ένα σύνολο πλειάδων $\{e^j_{a_0}, e^j_{a_1}, \ldots, e^j_{a_{v-1}}\}$. Αν υπάρχει το $e^l_{a_i}$ –προσέξτε ότι ο δείκτης $e^j_{a_i}.NF$ δείχνει στο $e^l_{a_i}$– για κάποιο $0 \leq i < v$, έτσι ώστε $\mathtt{bitop}(\mathtt{conc}(D_l(\sigma(p))), e^l_{a_i}.s) \leq k$, τότε το p εμφανίζεται στο t με το πολύ k-αναντιστοιχίες.



Σχήμα3.2 Λίστα X_j, για κάθε $0 \leq j < \binom{q}{q-k}$, για $q = 4$ και $k = 2$, μετά το Βήμα (II)

Στην πράξη, για πολύ μακρά μοτίβα,π. χ. $m > 48$, δεν είναι αρκετό να θέσουμε $q = 4$ και $k = 2$. Στην περίπτωση κατά την οποία $\lceil \frac{m \log |\Sigma|}{q-k} \rceil > w$, απαιτείται συνήθως $k > 2$. Εφαρμόζουμε την στρατηγική του *seed-and-extend*, κατά την οποία, εφαρμόζουμε το Βήμα (I), για $q = 4$ και $k' = 2$, μόνο για το πρόθεμα μήκους $m' < m$ του μοτίβου, το οποίο καλούμε *seed*, έτσι ώστε $\lceil \frac{m' \log |\Sigma|}{q-k'} \rceil \leq w$. Στο Βήμα (III), μόλις εντοπίσουμε μια επιτυχή ευθυγράμμιση με το πολύ k'-αναντιστοιχίες, τότε προσπαθούμε να επεκτείνουμε στα δεξιά την αντιστοίχιση για να ολοκληρώσουμε την ευθυγράμμιση. Συνεπώς μπορούμε να επιτρέψουμε μέχρι k'-αναντιστοιχίες στο *seed* μήκους m', και τις υπόλοιπες αναστοιχίες στην κατάληξη μήκους $m - m'$ του μοτίβου.

Ανάλυση πολυπλοκότητας Κατά την προ-επεξεργασία του κειμένου $t[0\,..\,n-1]$ από τα αριστερά προς τα δεξιά, υπολογίζουμε την υπογραφή $\sigma(t_i)$ της κάθε υποσυμβολοσειράς μήκους m του t, $t_i = t[i\,..\,i+m-1]$, για κάθε $0 \leq i \leq n-m$, και προσθέτουμε τις πλειάδες στη λίστα X_j, για κάθε $0 \leq j < \binom{q}{q-k}$, σε χρόνο $\Theta(\binom{q}{q-k})$. Μόλις υπολογίσουμε τη $\sigma(t_0)$, τότε κάθε $\sigma(t_i)$, για κάθε $1 \leq i \leq n-m$, μπορεί να υπολογιστεί σε σταθερό χρόνο χρησιμοποιώντας λειτουργίες σε επίπεδο λέξεων μηχανής, με αποτέλεσμα συνολικό χρόνο εκτέλεσης $\Theta(\binom{q}{q-k} n)$ για το Βήμα (I).

Στο Βήμα (II), ο χρόνος που απαιτείται για την ταξινόμηση της λίστας X_j, για κάθε $0 \leq j < \binom{q}{q-k}$, χρησιμοποιώντας τον radix sort, είναι $\Theta(\binom{q}{q-k} n)$. Η διατήρηση των δεικτών των πλειάδων απαιτεί επιπλέον σταθερό χρόνο.

Σύμφωνα με Λήμμα 1, το Βήμα (III) τρέχει σε χρόνο $\mathcal{O}(rv\binom{q}{q-k} \log n)$, όπου r είναι ο συνολικός αριθμός των μοτίβων, και v είναι ο μέγιστος αριθμός υποσυμ-

βολοσειρών μήκους m του t, οι οποίες περιέχουν τμήματα που αντιστοιχίζονται ακριβώς με τα ανάλογα τμήματα ενός δοθέντος μοτίβου. Ο παράγων $\binom{q}{q-k} \log n$ είναι για τη δυαδική αναζήτηση στη λίστα X_j, για κάθε $0 \leq j < \binom{q}{q-k}$.

Συνεπώς, συνολικά, ο αλγόριθμος απαιτεί χρόνο $\mathcal{O}(\binom{q}{q-k}(n + rv \log n))$, ο οποίος είναι $\mathcal{O}(\binom{q}{q-k}(n + r \log n))$, στην πράξη, εφόσον $v \ll n$.

Η λίστα X_j, για κάθε $0 \leq j < \binom{q}{q-k}$, αποτελείται ακριβώς από $n - m + 1$ πλειάδες. Συνεπώς ο χώρος ο οποίος απαιτείται είναι $\Theta(\binom{q}{q-k}n)$ για την αποθήκευση των λιστών στη μνήμη.

3.2 Πειραματικά αποτελέσματα

Το REAL υλοποιήθηκε από τους ερευνητές σε γλώσσα προγραμματισμού C++, και αναπτύχθηκε στο λειτουργικό σύστημα GNU/Linux. Το πρόγραμμα υλοποιήθηκε με τέτοιο τρόπο ώστε να μην φορτώνει αναγκαστικά όλη την ακολουθία αναφοράς στην κύρια μνήμη. Αντ΄ αυτού, φορτώνει τμήματα της ακολουθίας αναφοράς ανάλογα με το μέγεθος της φυσικής μνήμης της μηχανής. Όσον αφορά στην ευρετηριοποίηση από άποψη χώρου, δεν απαιτείται επιπλέον χώρος στο σκληρό δίσκο, καθώς το πρόγραμμα δεν αποθηκεύει στη δευτερεύουσα μνήμη το ευρετήριο της ακολουθίας αναφοράς. Το πρόγραμμα λαμβάνει ως ορίσματα εισόδου ένα αρχείο με την ακολουθία αναφοράς σε μορφή FASTA, και ένα αρχείο με τα short reads είτε σε μορφή FASTA είτε σε μορφή FASTQ, και στη συνέχεια παράγει ένα αρχείο με τις επιτυχημένες ευθυγραμμίσεις ως έξοδο.

Οι επιτυχείς ευθυγραμμίσεις αξιολογούνται χρησιμοποιώντας μια βαθμολογία (log odds scores)—(βλ. [13]) η οποία βασίζεται στη συχνότητα με την οποία οι εμφανιζόμενες αναντιστοιχίες παρατηρούνται στη φύση(β λ. [45]). Υπολογίζεται ένας πίνακας αξιολόγησης (scoring matrix) του προγράμματος με βάση το επίπεδο ομοιότητας των ακολουθιών, την απόκλιση σύνθεσης των ακολουθιών, την απόκλιση τύπου μετάλλαξης, και τη διαφορά ικανότητας των `G` και `C` να μεταλλάσσονται σε σύγκριση με τα `A` και `T` (transitions vs. transversions). Η βαθμολογία ορίζεται ως $S_{ij} = \log(P_{ij}/P_iP_j)$. P_{ij} είναι η στοχευμένη/παρατηρούμενη συχνότητα μετατροπής της βάσης i σε βάση j, και υπολογίζεται από τις δεδομένες αποκλίσεις μετάλλαξης. Ο παρονομαστής P_iP_j είναι η πιθανότητα να συμβεί το γεγονός κατά τύχη, με βάση το ιστορικό συχνότητας των δύο εμπλεκόμενων βάσεων, και εξαρτάται από την δεδομένη βασική σύνθεση της ακολουθίας. Αυτή η βαθμολογία συνδυάζεται με τη βαθμολογία ποιότητας της κάθε βάσης, όπως προτάθηκε στο [17].

Τα πειράματα διεξήχθησαν σε έναν επιτραπέζιο υπολογιστή,χ ρησιμοποιώντας έναν ενιαίο πυρήνα 2.40GHz Intel Xeon E7340 CPU και8 GB κύριας

μνήμης, με λειτουργικό σύστημα GNU/Linux. Το REAL διανέμεται υπό την άδεια GNU General Public License (GPL). Η εφαρμογή είναι διαθέσιμη στον διαδικτυακό τόπο [41], ο οποίος έχει δημιουργηθεί για τη διατήρηση του πηγαίου κώδικα και του οδηγού χρήσης.

Για την αξιολόγηση της επίδοσης του REAL, συγκρίναμε την επίδοση του με την αντίστοιχη επίδοση του SOAP2 (v2.20) και του Bowtie (v0.2.17), τα οποία είναι, μέχρι σήμερα, δύο από τα πιο δημοφιλή προγράμματα ευθυγράμμισης short reads. Σε κάθε περίπτωση, έγινε προσπάθεια τα προγράμματα να λειτουργούν με όσο το δυνατόν πιο παρόμοιο τρόπο, έτσι ώστε η σύγκριση απόδοσης, ευαισθησίας και ακρίβειας να είναι δίκαιη. Έτσι, στα SOAP2 και Bowtie δόθηκε πάντοτε ο τροποποιητής `-l <INT>`, για να προσαρμόζει το μήκος του seed ώστε να είναι ίσο με εκείνο του REAL. Τα προγράμματα ρυθμίστηκαν ώστε να αναφέρουν μόνο τις καλύτερες–μη επαναλαμβανόμενες–επιτυχείς ευθυγραμμίσεις, διαφορετικά τα αναφερόμενα αποτελέσματα του SOAP2 θα επιλέγονταν τυχαία μεταξύ ίσων αποτελεσμάτων. Στο SOAP2, αυτό επιτεύχθηκε με τη χρήση των τροποποιητών `-M 4 -r 0`, και στο Bowtie με τη χρήση του τροποποιητή –best. Επιπλέον, δεδομένου ότι το Bowtie κάνει χρήση της αντίστοιχης βαθμολογίας ποιότητας της κάθε βάσης(FASTQ), ενώ το SOAP2 όχι(FASTA), δύο εκδόσεις του REAL χρησιμοποιήθηκαν: μια με τον τροποποιητή-q 0, που αγνοεί τη βαθμολογία ποιότητας, και μια με τον τροποποιητή -q 1, που τη χρησιμοποιεί.

Ως αναφορά χρησιμοποιήθηκε το ανθρώπινο χρωμόσωμα 6 (166, 880, 988 bp), homo sapiens (άνθρωπος), δομή 37.2, που λήφθηκε από την NCBI [36]. Προσομοιώσαμε 25.000.000 short reads μήκους 64 bp από την ίδια ακολουθία. Όπως φαίνεται από τα αποτελέσματα του Πίνακα 3.2, το REAL ήταν σε θέση να ολοκληρώσει την εργασία πολύ πιο γρήγορα από ότι το SOAP2 και το Bowtie. Το REAL `-q 0` τελείωσε σε 27 λεπτά και 42 δευτερόλεπτα, το SOAP2 σε 34 λεπτά 36 δευτερόλεπτα, το REAL `-q 1` σε 32 λεπτά 55 δευτερόλεπτα, και το Bowtie σε 57 λεπτά 48 δευτερόλεπτα. Όσον αφορά την ευαισθησία, με αυτό το σύνολο δεδομένων, το SOAP2 και το REAL ήταν πιο ευαίσθητα από το Bowtie, με το SOAP2 να είναι λίγο πιο ευαίσθητο από ότι το REAL.

Ως επιπρόσθετο πείραμα, συγκρίναμε την επίδοση του REAL με τις αντίστοιχες επιδόσεις των SOAP2 και Bowtie, για την ευθυγράμμιση 24.543.488 προσομειωμένων short reads μήκους 70 bp από την ακολουθία Drosophila melanogaster 3L (24, 543, 557 bp), δομή 5.30, που λάβαμε από την NCBI. Όπως φαίνεται από τα αποτελέσματα στον Πίνακα 3.3, το REAL ήταν σε θέση να ολοκληρώσει την εργασία πολύ πιο γρήγορα από ότι το SOAP2 και το Bowtie. Το REAL `-q 0` τελείωσε σε 11 λεπτά και 43 δευτερόλεπτα, το SOAP2 σε 17 λεπτά και 46 δευτερόλεπτα, το REAL `-q 1` σε 16 λεπτά και 41 δευτερόλεπτα, και το Bowtie σε 42 λεπτά και 26 δευτερόλεπτα. Όσον αφορά στην ευαισθησία, με αυτό το σύνολο δεδομένων, το SOAP2 και το REAL ήταν πιο ευαίσθητα

Πίνακας3.2 Ευθυγράμμιση 25.000.000 short reads μήκους 64 bp στο ανθρώπινο χρωμόσωμα 6 (166, 880, 988 bp)

Πρόγραμμα	Συνολικός χρόνος		Επιτυχείς ευθυγραμμίσεις
	Ευρετηριοποίηση	Ευθυγράμμιση	
SOAP2	5m11s	29m25s	22,699,605
REAL `-q 0`	0m00s	27m42s	22,509,708
Bowtie	7m37s	50m11s	21,594,916
REAL `-q 1`	0m00s	32m55s	22,519,739

Σε όλα τα προγράμματα χρησιμοποιήθηκε seed μήκους 48 bp, με το πολύ 2 αναντιστοιχίες στο seed, και αναφέρθηκαν μόνο οι καλύτερες επιτυχείς ευθυγραμμίσεις.

Πίνακας3.3 Ευθυγράμμιση 24.543.488 short reads μήκους 70 bp στην ακολουθία Drosophila melanogaster 3L (24, 543, 557 bp)

Πρόγραμμα	Συνολικός χρόνος		Επιτυχείς ευθυγραμμίσεις	Ακρίβεια
	Ευρετηριοποίηση	Ευθυγράμμιση		
SOAP2	0m44s	17m02s	21,126,303	99,98%
REAL `-q 0`	0m00s	11m43s	21,134,692	99,98%
Bowtie	0m58s	41m28s	18,920,716	96,09%
REAL `-q 1`	0m00s	16m41s	21,134,699	99,98%

Σε όλα τα προγράμματα χρησιμοποιήθηκε seed μήκους 48 bp, με το πολύ 2 αναντιστοιχίες στο seed, και αναφέρθηκαν μόνο οι καλύτερες επιτυχείς ευθυγραμμίσεις.

από το Bowtie, με το REAL να είναι ελαφρώς πιο ευαίσθητο από το SOAP2. Λόγω του γεγονότος ότι τα δεδομένα ήταν προσομοιωμένα, και ως εκ τούτου, ήμασταν σε θέση να γνωρίζουμε τις ακριβείς θέσεις από τις οποίες προήλθαν, μετρήσαμε επίσης την ακρίβεια του κάθε προγράμματος, ελέγχοντας αν τα δεδομένα είχαν ευθυγραμμιστεί στις ίδιες ακριβώς θέσεις. Όπως αποδεικνύεται από τα αποτελέσματα του Πίνακα 3.3, με αυτό το σύνολο δεδομένων, το REAL και το SOAP2 είχαν ακρίβεια 99,98%, ενώ το Bowtie 96,09%.

Στο επόμενο πείραμα, συγκρίναμε την επίδοση του REAL με τις αντίστοιχες επιδόσεις του SOAP2 και του Bowtie, για την ευθυγράμμιση 24, 163, 065 πραγματικών short reads μήκους 76 bp σε ολόκληρη την ακολουθία του ανθρώπινου γονιδιώματος homo sapiens, δομή 37.2, που λάβαμε από την NCBI. Όπως φαίνεται από τα αποτελέσματα του Πίνακα 3.4, το SOAP2 ήταν σε θέση να ολοκληρώσει την εργασία ελαφρώς ταχύτερα από ότι το REAL, ενώ το

Πίνακας3.4 Ευθυγράμμιση 24, 163, 065 short reads μήκους 76 bp σε ολόκληρη την ακολουθία του ανθρώπινου γονιδιώματος

Πρόγραμμα	Συνολικός χρόνος		Επιτυχείς ευθυγραμμίσεις
	Ευρετηριοποίηση	Ευθυγράμμιση	
SOAP2	1h59m07s	1h52m20s	12,664,760
REAL `-q 0`	0m00s	4h07m48s	11,813,271
Bowtie	3h30m00s	1h57m40s	10,789,260
REAL `-q 1`	0m00s	4h21m38s	11,738,732

Σε όλα τα προγράμματα χρησιμοποιήθηκε seed μήκους 48 bp, με το πολύ 2 αναντιστοιχίες στο seed, και αναφέρθηκαν μόνο οι καλύτερες επιτυχείς ευθυγραμμίσεις.

Bowtie ήταν πολύ πιο αργό. Το SOAP2 τελείωσε σε 3 ώρες 51 λεπτά και 27 δευτερόλεπτα, το REAL `-q 0` σε 4 ώρες 7 λεπτά και 48 δευτερόλεπτα, το REAL `-q 1` σε 4 ώρες 21 λεπτά και 38 δευτερόλεπτα, και το Bowtie σε 5 ώρες 27 λεπτά και 40 δευτερόλεπτα. Όσον αφορά την ευαισθησία, με αυτό το σύνολο δεδομένων, το SOAP2 ήταν πιο ευαίσθητο από ότι το REAL, ενώ το Bowtie ήταν σημαντικά λιγότερο ευαίσθητο. Ωστόσο, στην περίπτωση αυτή, δεν ήμασταν σε θέση να μετρήσουμε την ακρίβεια του κάθε προγράμματος, λόγω του γεγονότος ότι δεν ήμασταν ενήμεροι για τις θέσεις από τις οποίες αντλήθηκαν τα short reads.

Είναι γνωστό ότι το SOAP2 έχει τεχνικές δυσκολίες για να χειριστεί short reads μικρότερου μήκους από 35bp, με αποτέλεσμα την κακή ακρίβεια [25] και τη φτωχή ευαισθησία [14]. Για να το επιβεβαιώσουμε αυτό, ευθυγραμμίσαμε 31.116.663 πραγματικά short reads μήκους 25 bp, τα οποία λάβαμε από πειράματα RNA-Seq [34], και 3.619.970 πραγματικά short reads μήκους 35 bp, τα οποία λάβαμε επίσης από πειράματα RNA-Seq [19], σε ολόκληρη την ακολουθία του γονιδιώματος του ποντικού, mus musculus, δομή 37.2, που λάβαμε από το NCBI. Από τα αποτελέσματα που καταδεικνύονται στον Πίνακα 3.5, είναι φανερό ότι το SOAP2 δεν αποδίδει καλά, από άποψης ευαισθησίας, σε short reads μήκους μικρότερου από 35bp. Ωστόσο, δεδομένου ότι τα δεδομένα είναι πραγματικά, πιστεύουμε ότι τα προγράμματα ευθυγράμμισης πρέπει να μπορούν να τα χειριστούν, ανεξάρτητα από το γεγονός ότι το μήκος των short reads, που παράγονται από τις τεχνολογίες επόμενης γενιάς, τείνει να αυξάνεται αντί να μειώνεται.

Πίνακας3.5 Ευθυγράμμιση31.116.663 short reads μήκους 25 bp και3.619.970 short reads μήκους 35 bp σε ολόκληρη την ακολουθία του γονιδιώματος του ποντικού

Πρόγραμμα	Επιτυχείς ευθυγραμμίσεις (25 bp)	Επιτυχείς ευθυγραμμίσεις (35 bp)
SOAP2	11,326,042	1,766,474
REAL `-q 0`	14,219,094	1,732,507

Και στα δύο προγράμματα χρησιμοποιήθηκε seed μήκους 25 bp για τα short reads μήκους 25 bp, seed μήκους 32 bp για τα short reads μήκους 35 bp με το πολύ 2 αναντιστοιχίες στο seed, και αναφέρθηκαν μόνο οι καλύτερες επιτυχείς ευθυγραμμίσεις.

Κεφάλαιο 4

Ευθυγράμμιση γενετικών δεδομένων νέας γενιάς σε ένα γονιδίωμα με κυκλική δομή

Το υλικό που παρουσιάζουμε σε αυτό το κεφάλαιο έχει δημοσιευτεί στο [12].

Tomáš Flouri, Costas S. Iliopoulos, Solon P. Pissis, and German Tischler. Mapping short reads to a genomic sequence with circular structure. *International Journal of Systems Biology and Biomedical Technologies*, 1(1):26–34, 2012

Σε αντίθεση με το γραμμικό DNA των σπονδυλωτών, σε στελέχη ή είδη βακτηρίων η κυκλική οργάνωση χρωμοσωμάτων ή πλασμιδίων, είναι η πιο κοινή. Μέχρι το τέλος της δεκαετίας του 1980, όταν η τεχνολογία για την εξέταση χρωμοσωμάτων και πλασμιδίων βελτιώθηκε, όλα τα βακτήρια πιστευόταν ότι έχουν ένα μόνο κυκλικό χρωμόσωμα [6]. Στην πραγματικότητα, δεν έχουν όλα τα βακτήρια ένα μόνο κυκλικό χρωμόσωμα. Κάποια βακτήρια έχουν πολλαπλά κυκλικά χρωμοσώματα [50, 51, 52, 53], και πολλά βακτήρια έχουν γραμμικά χρωμοσώματα και γραμμικά πλασμίδια [55]. Τα βακτηριακά γονιδιώματα κυμαίνονται σε μέγεθος από περίπου 160,000 bp έως 12,200,000 bp, ανάλογα με τον τύπο [35].

Σε αυτό το κεφάλαιο, καθώς τα περισσότερα από τα βακτηριακά χρωμοσώματα περιέχουν ένα κυκλικό μακρομόριο του DNA, παρουσιάζουμε το cREAL, μια απλή επέκταση του REAL (βλ. Κεφάλαιο 3), που έχει σχεδιαστεί ειδικά για την ευθυγράμμιση εκατομμυρίων μικρών καταγραφών σε ένα γονιδίωμα με κυκλική δομή. Συγκεκριμένα, ορίζουμε το πρόβλημα ως πρόβλημα κυκλικής αντιστοίχισης μοτίβων, το υποβαθμίζουμε σε ένα πρόβλημα αντιστοίχισης κλασικών μοτίβων, και κάνουμε χρήση του REAL, για να ευθυγραμμίσουμε αποτελεσματικά τα δεδομένα σε μια ακολουθία αναφοράς.

Συμβολίζουμε τα παραγόμενα short reads με το σύνολο $\{p_1, p_2, \ldots, p_r\}$, συνήθως $r > 10^6$, και τα αποκαλούμε μοτίβα. Το μήκος m κάθε μοτίβου, που παράγεται από το σύστημα Solexa της Illumina, έχει επί του παρόντος συνήθως μήκος μεταξύ 25 και 150 bp. Ορίζουμε την κυκλική ακολουθία αναφοράς μήκους n, συνήθως $n > 10^6$, με $C(t)$, και το αποκαλούμε κυκλικό κείμενο.

Ορίζουμε το πρόβλημα της ευθυγράμμιση εκατομμυρίων short reads σε ένα γονιδίωμα αναφοράς με κυκλική δομή, ως εξής.

Πρόβλημα Κυκλική αντιστοίχιση short reads)
Βρείτε κατά πόσο το μοτίβο $p_i = p_i[0\,..\,m-1]$, για κάθε $1 \leq i \leq r$, όπου $p_i \in \Sigma^+$, $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, εμφανίζεται, με το πολύ, k-αναντιστοιχίες, στο κυκλικό κείμενο $C(t)$ μήκους n, όπου $C(t) \in \Sigma^+$.

4.1 Αλγόριθμος

Η περιγραφή του προτεινόμενου αλγορίθμου για την επίλυση του Προβλήματος 2 είναι η ακόλουθη:

(ι) Ευθυγραμμιση του κυκλικου κειμενου

Η βασική ιδέα είναι να βρούμε πρώτα έναν τρόπο για να μετατρέψουμε το κυκλικό κείμενο $C(t)$ σε ένα(γραμμικό) κείμενο t', το οποίο είναι ισοδύναμο με το κυκλικό, υποβαθμίζοντας έτσι το πρόβλημα σε ένα κλασικό πρόβλημα αντιστοίχισης μοτίβων, όπου πρέπει να βρούμε όλες τις εμφανίσεις ενός δεδομένου μοτίβου p στο κείμενο t', με το πολύ k-αναντιστοιχίες.

Αυτό επιτυγχάνεται με τη διάσπαση, αρχικά, του $C(t)$ σε ένα αυθαίρετο σημείο κατά μήκος της ακολουθίας. Το αποτέλεσμα αυτής της διαδικασίας είναι μια συμβολοσειρά $t = t[0\,..\,n-1]$ με δύο άκρα: η αριστερότερη θέση, που είναι η έναρξη της συμβολοσειράς, και η δεξιότερη θέση, η οποία είναι το τέλος της συμβολοσειράς. Η συμβολοσειρά t που προκύπτει δεν είναι ακόμη ισοδύναμη με την αρχική συμβολοσειρά, δεδομένου ότι υπάρχει μια πιθανότητα να εμφανίζονται μερικά από τα μοτίβα στο σημείο όπου το $C(t)$ χωρίστηκε. Για να κάνουμε τη συμβολοσειρά t ισοδύναμη με το κυκλικό κείμενο $C(t)$, αντιγράφουμε το πρόθεμα μήκους $m-1$ του t, δηλαδή το $t[0\,..\,m-2]$, και το τοποθετούμε ως κατάληξη, με αποτέλεσμα μια νέα συμβολοσειρά $t' = t[0\,..\,n]t[0\,..\,m-2]$, η οποία είναι ισοδύναμη με το $C(t)$. Υποβαθμίσαμε τώρα το πρόβλημα στο να βρούμε αν το μοτίβο p_i, για κάθε $1 \le i \le r$, εμφανίζεται στο $t' = t'[0\,..\,n+m-2]$, με το πολύ k-αναντιστοιχίες.

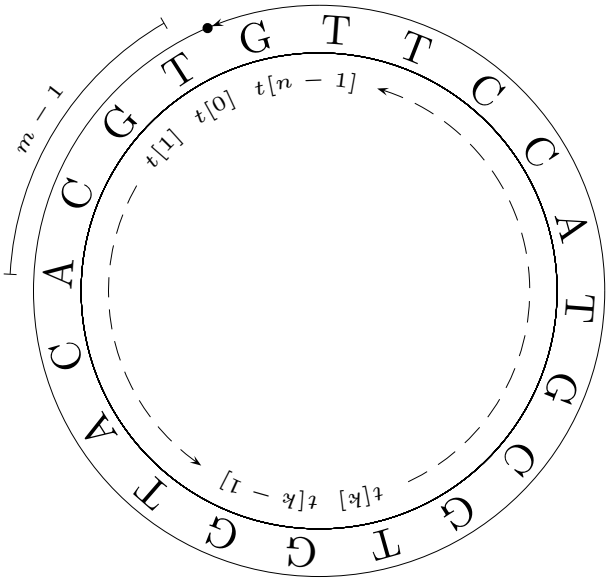
Παράδειγμα

*Το κυκλικό κείμενο $C(t)$, που φαίνεται στην Σχήμα 4.1, απεικονίζει αυτή τη διαδικασία. Η συμβολοσειρά που παράγεται από τη διάσπαση του $C(t)$ στο σημείο που η κεφαλή του βέλους αγγίζει το οβάλ βέλος, είναι $t =$ **TGCACATGGTGCGTACCTTG**. Ας υποθέσουμε ότι έχουμε ένα μοτίβο $p =$ **TGTGC**. Παρατηρούμε ότι υπάρχει μια εμφάνιση του p σε $C(t)$, αλλά δεν υπάρχει εμφάνιση του p στη γραμμική μορφή της $C(t)$, λόγω του ότι το μοτίβο εμφανίζεται κατά το μήκος της διάσπασης. Μια γραμμική συμβολοσειρά, η οποία καλύπτει όλες τις πιθανές υποσυμβολοσειρές με μια δεδομένη διάσπαση, για μοτίβα μήκους $m = 5$, μπορεί να ληφθεί με την αντιγραφή των πρώτων $m-1$ γραμμάτων, δηλαδή του προθέματος $t[0\,..\,3] =$ **TGCA**, και την προσάρτηση του στο τέλος της συμβολοσειράς t. Το αποτέλεσμα αυτής της διαδικασίας είναι μια συμβολοσειρά $t' =$ **TGCACATGGTGCGTACCTTGTGCA**, η οποία καλύπτει κάθε πιθανή θέση εμφάνισης γύρω από το $C(t)$.*

(ιι) Ευθυγραμμιση

Με το REAL (βλ. Κεφάλαιο 3) ψάχνουμε για τα μοτίβα στο t'.

Ανάλυση πολυπλοκότητας Το Βήμα(i) μπορεί να εκτελεστεί σε χρόνο $\Theta(m)$. Το Βήμα(ii) μπορεί να εκτελεστεί σε χρόνο $\mathcal{O}(\binom{q}{q-k}(n + r \log n))$ (βλ. Κεφάλαιο 3).



Σχήμα4.1 Κυκλικό κείμενο $C(t)$. Το σημείο που η κεφαλή του βέλους αγγίζει το οβάλ βέλος είναι το σημείο που διασπάστηκε η συμβολοσειρά. Για να ψάξουμε για ένα μοτίβο p μήκους m σε αυτό το κυκλικό κείμενο, τα πρώτα $m-1$ γράμματα της συμβολοσειράς λαμβάνονται από το σημείο όπου ξεκινά η διάσπαση στο $t[0]$, και προστίθενται στο τέλος $t[n-1]$.

Ως εκ τούτου, συνολικά, ο προτεινόμενος αλγόριθμος χρειάζεται χρόνο $\mathcal{O}(\binom{q}{q-k}(n + r \log n))$.

Ο χώρος που απαιτείται είναι $\Theta(\binom{q}{q-k}n)$.

4.2 Πειραματικά αποτελέσματα

Το cREAL, η απλή επέκταση του REAL, υλοποιήθηκε από τους ερευνητές σε γλώσσα προγραμματισμού Perl, και αναπτύχθηκε σε λειτουργικό σύστημα GNU/Linux.

Τα πειράματα διεξήχθησαν σε έναν επιτραπέζιο υπολογιστή,χ ρησιμοποιώντας έναν ενιαίο πυρήνα 2.40GHz Intel Xeon E7340 CPU και8 GB κύριας μνήμης, με λειτουργικό σύστημα GNU/Linux. Το cREAL διανέμεται υπό την άδεια GNU General Public License (GPL). Η εφαρμογή είναι διαθέσιμη στον διαδικτυακό τόπο [41], ο οποίος έχει δημιουργηθεί για τη διατήρηση του πηγαίου κώδικα και του οδηγού χρήσης.

Για την αξιολόγηση της επίδοσης του cREAL, συγκρίναμε την επίδοση του με την αντίστοιχη επίδοσης του SOAP2 (v2.20) και του Bowtie (v0.2.17), τα οποία είναι, μέχρι σήμερα, δύο από τα πιο δημοφιλή προγράμματα ευθυγράμμισης

short reads. Σε κάθε περίπτωση, έγινε προσπάθεια τα προγράμματα να λειτουργούν με όσο το δυνατόν πιο παρόμοιο τρόπο, έτσι ώστε η σύγκριση απόδοσης, ευαισθησίας και ακρίβειας να είναι δίκαιη. Έτσι, στα SOAP2 και Bowtie δόθηκε πάντοτε ο τροποποιητής `-l <INT>`, για να προσαρμόζει το μήκος του seed ώστε να είναι ίσο με εκείνο του cREAL. Τα προγράμματα ρυθμίστηκαν ώστε να αναφέρουν μόνο τις καλύτερες–μη επαναλαμβανόμενες–επιτυχείς ευθυγραμμίσεις, διαφορετικά τα αναφερόμενα αποτελέσματα του SOAP2 θα επιλέγονταν τυχαία μεταξύ ίσων αποτελεσμάτων. Στο SOAP2, αυτό επιτεύχθηκε με τη χρήση των τροποποιητών `-M 4 -r 0`, και στο Bowtie με τη χρήση του τροποποιητή –best. Επιπλέον, δεδομένου ότι το Bowtie κάνει χρήση της αντίστοιχης βαθμολογίας ποιότητας της κάθε βάσης(FASTQ,) ενώ το SOAP2 όχι(FASTA), δύο εκδόσεις του cREAL χρησιμοποιήθηκαν: μια με τον τροποποιητή-q 0, που αγνοεί τη βαθμολογία ποιότητας, και μια με τον τροποποιητή -q 1, που τη χρησιμοποιεί.

Ως γονιδίωμα αναφοράς, χρησιμοποιήσαμε το μοναδικό κυκλικό χρωμόσωμα του Bradyrhizobium Japonicum (9, 105, 828 bp), που λάβαμε από το [21]. Τα short reads λήφθησαν με προσομοίωση 9, 105, 777 short reads μήκους 52 bp από την ίδια ακολουθία. Όπως φαίνεται από τα αποτελέσματα του Πίνακα 4.1, το cREAL και το SOAP2 ήταν σε θέση να ολοκληρώσουν την εργασία πολύ γρηγορότερα από το Bowtie. Το SOAP2 τελείωσε σε 5 λεπτά και 12 δευτερόλεπτα, το cREAL `-q 0` τελείωσε σε 5 λεπτά και 25 δευτερόλεπτα, το cREAL `-q 1` τελείωσε σε 6 λεπτά και 15 δευτερόλεπτα, ενώ το Bowtie σε 10 λεπτά και 15 δευτερόλεπτα. Όσον αφορά στην ευαισθησία, με αυτό το σύνολο δεδομένων, το cREAL ήταν ελαφρώς πιο ευαίσθητο από το SOAP2 και το Bowtie. Λόγω του γεγονότος ότι τα δεδομένα προσομοιώθηκαν, και ως εκ τούτου, ήμασταν σε θέση να γνωρίζουμε τις ακριβείς θέσεις από τις οποίες προήλθαν, μετρήσαμε την ακρίβεια του κάθε προγράμματος, ελέγχοντας αν τα δεδομένα είχαν ευθυγραμμιστεί στις ίδιες ακριβώς θέσεις. Όπως φαίνεται από τα αποτελέσματα του Πίνακα 4.1, με αυτό το σύνολο δεδομένων, το cREAL και το SOAP2 είχαν ακρίβεια 99,99%, ενώ το Bowtie είχε ακρίβεια 98,78%.

Παρόμοια αποτελέσματα ελήφθησαν στους Πίνακες 4.2–4.4. Στον Πίνακα 4.2, για την ευθυγράμμιση 3, 294, 805 προσομοιωμένων short reads μήκους 64 bp στο 16M γονιδίωμα του brucella melitensis, που λήφθηκε από το [18], το οποίο αποτελείται από δύο κυκλικά χρωμοσώματα(2, 124, 241 bp και 1, 162, 204 bp). Στον Πίνακα 4.3, για την ευθυγράμμιση 6, 264, 333 προσομοιωμένων short reads μήκους 72 bp στο μοναδικό κυκλικό χρωμόσωμα του pseudomonas aeruginosa PAO1 (6, 264, 403 bp), που λήφθηκε από [49]. Στον Πίνακα 4.4, για την ευθυγράμμιση 2, 475, 055 προσομοιωμένων δεδομένων ενός αποτελέσματος μήκους 64 bp στο μοναδικό κυκλικό χρωμόσωμα του Xylella Fastidiosa M12 (2, 475, 130 bp), που λήφθηκε από [5].

Ως τελευταίο πείραμα, συγκρίναμε την επίδοση του cREAL με την αντίστοι-

Πίνακας4.1 Ευθυγράμμιση 9,105,777 short reads μήκους 52 bp στο γονιδίωμα Bradyrhizobium Japonicum

Πρόγραμμα	Συνολικός χρόνος		Επιτυχείς ευθυγραμμίσεις	Ακρίβεια
	Ευρετηριοποίηση	Ευθυγράμμιση		
SOAP2	0m21s	4m51s	8,746,116	99,99%
cREAL `-q 0`	0m00s	5m25s	8,747,172	99,99%
Bowtie	0m15s	10m00s	8,248,842	98,78%
cREAL `-q 1`	0m00s	6m15s	8,747,233	99,99%

Σε όλα τα προγράμματα χρησιμοποιήθηκε seed μήκους 32 bp, με το πολύ 2 αναντιστοιχίες στο seed, και αναφέρθηκαν μόνο οι καλύτερες επιτυχείς ευθυγραμμίσεις.

Πίνακας4.2 Ευθυγράμμιση 3,294,805 short reads μήκους 64 bp στο γονιδίωμα Brucella melitensis 16M

Πρόγραμμα	Συνολικός χρόνος		Επιτυχείς ευθυγραμμίσεις	Ακρίβεια
	Ευρετηριοποίηση	Ευθυγράμμιση		
SOAP2	0m17s	1m26s	3,214,557	99,99%
cREAL `-q 0`	0m00s	2m10s	3,214,472	99,99%
Bowtie	0m05s	2m56s	2,962,644	99,26%
cREAL `-q 1`	0m00s	2m31s	3,214,485	99,99%

Σε όλα τα προγράμματα χρησιμοποιήθηκε seed μήκους 32 bp, με το πολύ 2 αναντιστοιχίες στο seed, και αναφέρθηκαν μόνο οι καλύτερες επιτυχείς ευθυγραμμίσεις.

Πίνακας4.3 Ευθυγράμμιση 6,264,333 short reads μήκους 72 bp στο γονιδίωμα Pseudomonas aeruginosa PAO1

Πρόγραμμα	Συνολικός χρόνος		Επιτυχείς ευθυγραμμίσεις	Ακρίβεια
	Ευρετηριοποίηση	Ευθυγράμμιση		
SOAP2	0m19s	3m36s	6,035,526	99,99%
cREAL `-q 0`	0m00s	3m17s	6,037,765	99,99%
Bowtie	0m10s	7m46s	4,896,047	99,24%
cREAL `-q 1`	0m00s	4m17s	6,037,765	99,99%

Σε όλα τα προγράμματα χρησιμοποιήθηκε seed μήκους 32 bp, με το πολύ 2 αναντιστοιχίες στο seed, και αναφέρθηκαν μόνο οι καλύτερες επιτυχείς ευθυγραμμίσεις.

Πίνακας4.4 Ευθυγράμμιση 2,475,055 short reads μήκους 76 bp στο γονιδίωμα Xylella fastidiosa M12

Πρόγραμμα	Συνολικός χρόνος		Επιτυχείς ευθυγραμμίσεις	Ακρίβεια
	Ευρετηριοποίηση	Ευθυγράμμιση		
SOAP2	0m15s	1m14s	2,255,798	99,99%
cREAL `-q 0`	0m00s	1m13s	2,257,124	99,99%
Bowtie	0m04s	2m45s	1,918,988	96,53%
cREAL `-q 1`	0m00s	1m50s	2,257,117	99,99%

Σε όλα τα προγράμματα χρησιμοποιήθηκε seed μήκους 32 bp, με το πολύ 2 αναντιστοιχίες στο seed, και αναφέρθηκαν μόνο οι καλύτερες επιτυχείς ευθυγραμμίσεις.

Πίνακας4.5 Ευθυγράμμιση 5,288,154 short reads μήκους 36 bp στο γονιδίωμα Escherichia Coli

Πρόγραμμα	Συνολικός χρόνος		Επιτυχείς ευθυγραμμίσεις
	Ευρετηριοποίηση	Ευθυγράμμιση	
SOAP2	0m18s	2m14s	3,532,761
cREAL `-q 0`	0m00s	2m30s	3,255,228
Bowtie	0m07s	6m20s	3,646,029
cREAL `-q 1`	0m00s	2m48s	3,255,266

Σε όλα τα προγράμματα χρησιμοποιήθηκε seed μήκους 32 bp, με το πολύ 2 αναντιστοιχίες στο seed, και αναφέρθηκαν μόνο οι καλύτερες επιτυχείς ευθυγραμμίσεις.

χη επίδοση των SOAP2 και Bowtie, για την ευθυγράμμιση 5,288,154 πραγματικών short reads μήκους 36 bp, που λήφθησαν από το European Nucleotide Archive [10], στο μοναδικό κυκλικό χρωμόσωμα του Escherichia Coli str. K-12 substr. MG1655, που λήφθηκε από την GenBank [15]. Στα αποτελέσματα του Πίνακα 4.5, δεν παρατηρήθηκε καμιά διαφορά όσον αφορά στην αποτελεσματικότητα, ενώ με αυτό το σύνολο δεδομένων, το Bowtie ήταν ελαφρώς πιο ευαίσθητο από το cREAL και το SOAP2. Ωστόσο, στην περίπτωση αυτή, δεν ήμασταν σε θέση να μετρήσουμε την ακρίβεια του κάθε προγράμματος, λόγω του γεγονότος ότι δεν ήμασταν ενήμεροι για τις θέσεις από τις οποίες αντλήθηκαν τα short reads.

Κεφάλαιο 5

Συμπεράσματα

Η παραγωγή μεγάλων ποσοτήτων γενετικών δεδομένων, με τη μορφή μικρών ακολουθιών, των short reads, από στενά συσχετιζόμενα είδη ή οργανισμούς του ιδίου είδους, είναι ο κινητήριος μοχλός για την εφαρμογή που είναι γνωστή ως *επαναπροσδιορισμός αλληλουχίας* (re-sequencing).

Σε αυτήν τη Διπλωματική Εργασία έχουμε παρουσιάσει το REAL, ένα αποτελεσματικό, ευαίσθητο και ακριβές πρόγραμμα αντιστοίχισης μικρών ακολουθιών για την ευθυγράμμιση εκατομμυρίων short reads σε ένα γονιδίωμα αναφοράς. Βασίζεται σε ένα νέο, σχετικά, αλγόριθμο, διαφορετικό από όσους χρησιμοποιούνταν μέχρι τώρα.

Μετά από μια σειρά πειραμάτων έχουμε δείξει ότι μπορεί να ανταγωνιστεί, ή ακόμη και να ξεπεράσει, τα τρέχοντα δημοφιλή προγράμματα ευθυγράμμισης short reads, όπως το Bowtie και το SOAP2, όσον αφορά στην αποτελεσματικότητα, την ευαισθησία, και την ακρίβεια. Τα δύο κύρια πλεονεκτήματα του REAL, σε σχέση με τα πιο δημοφιλή προγράμματα, είναι τα εξής: Οι επιδόσεις του—σε ότι αφορά στην ευαισθησία και την ακρίβεια—είναι συνεπείς και ανεξάρτητες από το μήκος των short reads. Επίσης, ενώ μπορεί να ανταγωνιστεί, ή ακόμη και να ξεπεράσει, τρέχοντα δημοφιλή προγράμματα από την άποψη της αποτελεσματικότητας με μεγάλα γονιδιώματα ως είσοδο, είναι πάντα πιο αποτελεσματικό με τα μικρά. Το κύριο μειονέκτημα του REAL είναι ότι δε δημιουργεί ένα μόνιμο ευρετήριο του γονιδιώματος αναφοράς, το οποίο θα μπορούσε να επαναχρησιμοποιηθεί σε διαδοχικές λειτουργίες ευθυγράμμισης—ειδικά για μεγάλα γονιδιώματα.

Επιπλέον, έχουμε παρουσιάσει το cREAL, μια απλή επέκταση του REAL, η οποία έχει σχεδιαστεί ειδικά για την ευθυγράμμιση των εκατομμυρίων short reads σε ένα γονιδίωμα με κυκλική δομή,π. χ. στέλεχος ή είδη βακτηρίων με κυκλική οργάνωση των χρωμοσωμάτων ή πλασμιδίων τους.

Bibliography

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[2] Pavlos Antoniou, Jackie W. Daykin, Costas S. Iliopoulos, Derrick Kourie, Laurent Mouchard, and Solon P. Pissis. Mapping uniquely occurring short sequences derived from high throughput technologies to a reference genome. In E.C. Kyriakou, C.P. Loizou, C.S. Christodoulou, P.D. Bamidis, V. Promponas, Y. Poirazi, C.N. Schizas, and C.S. Pattichis, editors, *Proceedings of the ninth IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB 2009)*, Cyprus, 2009.

[3] Pavlos Antoniou, Costas S. Iliopoulos, Laurent Mouchard, and Solon P. Pissis. A fast and efficient algorithm for mapping short sequences to a reference genome. In Hamid R. Arabnia, editor, *Advances in Computational Biology*, volume 680 of *Advances in Experimental Medicine and Biology*, pages 399–403. Springer, 2011.

[4] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical Report SRC-RR-124, Standord Univeristy, 1994.

[5] J. Chen, G. Xie, S. Han, O. Chertkov, D. Sims, and E. L. Civerolo. Whole Genome Sequences of Two Xylella fastidiosa Strains (M12 and M23) Causing Almond Leaf Scorch Disease in California. *Journal of Bacteriology*, 192(17):4534, 2010.

[6] S. T. Colem and I. Saint-Girons. Bacterial genomes - all shapes and sizes. *Organization of the prokaryotic genome*, 291:35–62, 1999.

[7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, 2001.

[8] A. Cox. ELAND: Efficient local alignment of nucleotide data. (unpublished), 2005.

[9] M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on Strings*. Cambridge University Press, USA, 2007.

[10] European Nucleotide Archive (ENA). `http://www.ebi.ac.uk/ena/data/view/`, August 2011.

[11] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In IEEE, editor, *Proceedings of the fourty-first annual Symposium on Foundations of Computer Science (FOCS 2000)*, pages 390–398, USA, 2000. IEEE Computer Society.

[12] Tomáš Flouri, Costas S. Iliopoulos, Solon P. Pissis, and German Tischler. Mapping short reads to a genomic sequence with circular structure. *International Journal of Systems Biology and Biomedical Technologies*, 1(1):26–34, 2012.

[13] Martin C. Frith, Raymond Wan, and Paul Horton. Incorporating sequence quality data into alignment improves DNA read mapping. *Nucleic Acids Research*, 38:1–9, 2010.

[14] Kimon Frousios, Costas S. Iliopoulos, Laurent Mouchard, Solon P. Pissis, and German Tischler. REAL: an efficient REad ALigner for next generation sequencing reads. In Aidong Zhang, Mark Borodovsky, Gultekin Özsoyoglu, and Armin R. Mikler, editors, *Proceedings of the first ACM International Conference on Bioinformatics and Computational Biology (BCB 2011)*, pages 154–159, USA, 2010. ACM.

[15] GenBank. `http://www.ncbi.nlm.nih.gov/genbank/`, August 2011.

[16] Gilbert, Lynn P. Tomsho, Snjezana Rendulic, Michael Packard, Daniela I. Drautz, Andrei Sher, Alexei Tikhonov, Love Dalén, Tatyana Kuznetsova, Pavel Kosintsev, Paula F. Campos, Thomas Higham, Matthew J. Collins, Andrew S. Wilson, Fyodor Shidlovskiy, Bernard Buigues, Per G. P. Ericson, Mietje Germonpré, Anders Götherström, Paola Iacumin, Vladimir Nikolaev, Malgosia Nowak-Kemp, Eske Willerslev, James R. Knight, Gerard P. Irzyk, Clotilde S. Perbost, Karin M. Fredrikson, Timothy T. Harkins, Sharon Sheridan, Webb Miller, and Stephan C. Schuster. Whole-Genome Shotgun Sequencing of Mitochondria from Ancient Hair Shafts. *Science*, 317(5846):1927–1930, 2007.

[17] Warren Gish and Stephen F. Altschul. Improved sensitivity of nucleic acid database searches using application-specific scoring matrices. *Methods*, 3(1):66–70, 1991.

[18] Shirley M. Halling, Brooke D. Peterson-Burch, Betsy J. Bricker, Richard L. Zuerner, Zhang Qing, Ling-Ling Li, Vivek Kapur, David P. Alt, and Steven C. Olsen. Completion of the Genome Sequence of Brucella abortus and Comparison to the Highly Similar Genomes of Brucella melitensis and Brucella suis. *Journal of Bacteriology*, 187(8):2715–2726, 2005.

[19] X. Han, X. Wu, W.-Y. Chung, T. Li, A. Nekrutenko, N. S. Altman, G. Chen, and H. Ma. Transcriptome of embryonic and neonatal mouse cortex by high-throughput RNA sequencing. *Proceedings of the National Academy of Sciences of the United States of America*, 106(31):12741–12746, 2009.

[20] Hui Jiang and Wing Hung Wong. SeqMap: mapping massive amount of oligonucleotides to the genome. *Bioinformatics*, 24(20):2395–2396, 2008.

[21] Takakazu Kaneko, Yasukazu Nakamura, Shusei Sato, Kiwamu Minamisawa, Toshiki Uchiumi, Shigemi Sasamoto, Akiko Watanabe, Kumi Idesawa, Mayumi Iriguchi, Kumiko Kawashima, Mitsuyo Kohara, Midori Matsumoto, Sayaka Shimpo, Hisae Tsuruoka, Tsuyuko Wada, Manabu Yamada, and Satoshi Tabata. Complete Genomic Sequence of Nitrogen-fixing Symbiotic Bacterium Bradyrhizobium japonicum USDA110. *DNA Research*, 9(6):189–197, 2002.

[22] W. James Kent. BLAT - The BLAST-Like Alignment Tool. *Genome Research*, 12(4):656–664, 2002.

[23] T. W. Lam, W. K. Sung, S. L. Tam, C. K. Wong, and S. M. Yiu. Compressed Indexing and Local Alignment of DNA. *Bioinformatics*, 24(6):791–797, 2008.

[24] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology*, 10(3):R25+, 2009.

[25] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[26] Heng Li, Jue Ruan, and Richard Durbin. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research*, 18(11):1851–1858, 2008.

[27] Ruiqiang Li, Yingrui Li, Karsten Kristiansen, and Jun Wang. SOAP: short oligonucleotide alignment program. *Bioinformatics*, 24(5):713–714, 2008.

[28] Ruiqiang Li, Chang Yu, Yingrui Li, Tak-Wah Lam, Siu-Ming Yiu, Karsten Kristiansen, and Jun Wang. SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25(16):1966–1967, 2009.

[29] Ruiqiang Li, Hongmei Zhu, Jue Ruan, Wubin Qian, Xiaodong Fang, Zhongbin Shi, Yingrui Li, Shengting Li, Gao Shan, Karsten Kristiansen, Songgang Li, Huanming Yang, Jian Wang, and Jun Wang. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, 20(2):265–272, 2010.

[30] Ross A. Lippert. Space-efficient whole genome comparisons with Burrows-Wheeler transforms. *Journal of computational biology : a journal of computational molecular cell biology*, 12(4):407–415, 2005.

[31] Marcel Margulies, Michael Egholm, William E. Altman, Said Attiya, Joel S. Bader, Lisa A. Bemben, Jan Berka, Michael S. Braverman, Yi-Ju Chen, Zhoutao Chen, Scott B. Dewell, Lei Du, Joseph M. Fierro, Xavier V. Gomes, Brian C. Godwin, Wen He, Scott Helgesen, Chun H. Ho, Gerard P. Irzyk, Szilveszter C. Jando, Maria L. I. Alenquer, Thomas P. Jarvie, Kshama B. Jirage, Jong-Bum Kim, James R. Knight, Janna R. Lanza, John H. Leamon, Steven M. Lefkowitz, Ming Lei, Jing Li, Kenton L. Lohman, Hong Lu, Vinod B. Makhijani, Keith E. McDade, Michael P. McKenna, Eugene W. Myers, Elizabeth Nickerson, John R. Nobile, Ramona Plant, Bernard P. Puc, Michael T. Ronan, George T. Roth, Gary J. Sarkis, Jan F. Simons, John W. Simpson, Maithreyan Srinivasan, Karrie R. Tartaro, Alexander Tomasz, Kari A. Vogt, Greg A. Volkmer, Shally H. Wang, Yong Wang, Michael P. Weiner, Pengguang Yu, Richard F. Begley, and Jonathan M. Rothberg. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, 2005.

[32] A. M. Maxam and W. Gilbert. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America*, 74(2):560–564, 1977.

[33] Jason R Miller, Sergey Koren, and Granger Sutton. Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315–327, 2010.

[34] Ali Mortazavi, Brian A A. Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by *RNA-Seq*. *Nature methods*, 5(7):621–628, 2008.

[35] Atsushi Nakabachi, Atsushi Yamashita, Hidehiro Toh, Hajime Ishikawa, Helen E. Dunbar, Nancy A. Moran, and Masahira Hattori. The 160-Kilobase Genome of the Bacterial Endosymbiont Carsonella. *Science*, 314(5797):267, 2006.

[36] National Center for Biotechnology Information (NCBI). `http://www.ncbi.nlm.nih.gov/`, August 2011.

[37] Gonzalo Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[38] Sarah B. Ng, Kati J. Buckingham, Choli Lee, Abigail W. Bigham, Holly K. Tabor, Karin M. Dent, Chad D. Huff, Paul T. Shannon, Ethylin W. Jabs, Deborah A. Nickerson, Jay Shendure, and Michael J. Bamshad. Exome sequencing identifies the cause of a mendelian disorder. *Nature Genetics*, 42(1):30–35, 2010.

[39] Pia Ostergaard, Michael A Simpson, Glen Brice, Sahar Mansour, Fiona C Connell, Alexandros Onoufriadis, Anne H Child, Jae Hwang, Kamini Kalidas, Peter S Mortimer, Richard Trembath, and Steve Jeffery. Rapid identification of mutations in GJC2 in primary lymphoedema using whole exome sequencing combined with linkage analysis with delineation of the phenotype. *Journal of Medical Genetics*, 48(4):251–255, 2010.

[40] William R. Pearson and David J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Science of the United States of America*, 85(8):2444–2448, 1988.

[41] REad ALigner (`REAL`). `http://www.inf.kcl.ac.uk/pg/real/`, August 2011.

[42] F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, C. A. Fiddes, C. A. Hutchison, P. M. Slocombe, and M. Smith. Nucleotide sequence of bacteriophage phi X174 DNA. *Nature*, 265(5596):687–695, 1977.

[43] F. Sanger and A. R. Coulson. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *Journal of Molecular Biology*, 94(3):441–448, 1975.

[44] F. Sanger, S. Nicklen, and A. R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Science of the United States of America*, 74(12):5463–5467, 1977.

[45] Julie A. Schneider, Manish S. Pungliya, Julie Y. Choi, Ruhong Jiang, Xiao Jenny Sun, Benjamin A. Salisbury, and J. Claiborne Stephens. DNA variability of human genes. *Mechanisms of Ageing and Development*, 124(1):17–25, 2003.

[46] Jay Shendure, Gregory J. Porreca, Nikos B. Reppas, Xiaoxia Lin, John P. McCutcheon, Abraham M. Rosenbaum, Michael D. Wang, Kun Zhang, Robi D. Mitra, and George M. Church. Accurate Multiplex Polony Sequencing of an Evolved Bacterial Genome. *Science*, 309(5741):1728–1732, 2005.

[47] Jared T. Simpson, Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J. M. Jones, and İnanç Birol. ABySS: A parallel assembler for short read sequence data. *Genome Research*, 19(6):1117–1123, 2009.

[48] Michael A. Simpson, Melita D. Irving, Esra Asilmaz, Mary J. Gray, Dimitra Dafou, Frances V. Elmslie, Sahar Mansour, Sue E. Holder, Caroline E. Brain, Barbara K. Burton, Katherine H. Kim, Richard M. Pauli, Salim Aftimos, Helen Stewart, Chong Ae Kim, Muriel Holder-Espinasse, Stephen P. Robertson, William M. Drake, and Richard C. Trembath. Mutations in NOTCH2 cause Hajdu-Cheney syndrome, a disorder of severe and progressive bone loss. *Nature Genetics*, 43(4):303–305, 2011.

[49] C. K. Stover, X. Q. Pham, A. L. Erwin, S. D. Mizoguchi, P. Warrener, M. J. Hickey, F. S. Brinkman, W. O. Hufnagle, D. J. Kowalik, M. Lagrou, R. L. Garber, L. Goltry, E. Tolentino, S. Westbrock-Wadman, Y. Yuan, L. L. Brody, S. N. Coulter, K. R. Folger, A. Kas, K. Larbig, R. Lim, K. Smith, D. Spencer, G. K. Wong, Z. Wu, I. T. Paulsen, J. Reizer, M. H. Saier, R. E. Hancock, S. Lory, and M. V. Olson. Complete genome sequence of Pseudomonas aeruginosa PAO1, an opportunistic pathogen. *Nature*, 406(6799):959–964, 2000.

[50] A Suwanto and S Kaplan. Physical and genetic mapping of the Rhodobacter sphaeroides 2.4.1 genome: genome size, fragment identification, and gene localization. *Journal of Bacteriology*, 171(11):5840–5849, 1989.

[51] A Suwanto and S Kaplan. Physical and genetic mapping of the Rhodobacter sphaeroides 2.4.1 genome: presence of two unique circular chromosomes. *Journal of Bacteriology*, 171(11):5850–5859, 1989.

[52] A Suwanto and S Kaplan. A self-transmissible, narrow-host-range endogenous plasmid of Rhodobacter sphaeroides 2.4.1: physical structure, incompatibility determinants, origin of replication, and transfer functions. *Journal of Bacteriology*, 174(4):1124–1134, 1992.

[53] A Suwanto and S Kaplan. Chromosome transfer in Rhodobacter sphaeroides: Hfr formation and genetic evidence for two unique circular chromosomes. *Journal of Bacteriology*, 174(4):1135–1145, 1992.

[54] John R. ten Bosch and Wayne W. Grody. Keeping up with the next generation : Massively parallel sequencing in clinical diagnostics. *Journal of Molecular Diagnostics*, 10(6):484–492, 2008.

[55] Jean-Nicolas Volff and Josef Altenbuchner. A new beginning with new ends: linearisation of circular chromosomes during bacterial evolution. *FEMS Microbiology Letters*, 186(2):143–150, 2000.