1. Programozási projekt: lefedettség számolása

1.1. A projekt rövid leírása

A projekt fő célja, hogy végigvezessen egy szekvenálási adathalmaz illesztésének főbb lépésein, és az illesztések manipulációján. A gyakorlatban egy Java programcsomagot fejlesztünk ki, amely a genomot többszörösen lefedő szekvenálási adatokra a pozíció-specifikus *coverage* értéket kiszámolni és böngészőben megjeleníteni. Egészen pontosan szeretnénk annotálni minden genom-beli i pozícióra, hogy azt hány olvasat (*read*) foglalja magában. Ehhez létező szoftver-implentációkat használunk fel, illetve kifejlesztünk egy saját módszert.

1.2. Feladatok (20 pont)

a. Adatok letöltése (2 pont). A projektben *Drosophila melanogaster* (muslinca) genommal fogunk dolgozni (ami kevesebb, mint 150 Mbp hosszú), a modENCODE projekt keretében generált RNA-Seq szekvenciákkal, és a UCSC genome browserrel.

▶ Töltsd le a D. melanogaster genom szekvenciáját a UCSC genome browser oldaláról! A dm6 assembly egyben megtalálható a ftp://hgdownload.cse.ucsc.edu/goldenPath/dm6/bigZips/dm6.fa.gz címen (erről az információt a http://hgdownload.soe.ucsc.edu/downloads.html oldalon lehet megtalálni, ahova a „Downloads" menüpontból jutunk el.) Bontsd ki a fájlt (`gunzip`) — egyszerűbb lesz dolgozni vele.

▶ Töltsd le az SRR111873 kódú szekvenálási „nyers" adatokat az NCBI Short Read Archiveből! (Keress rá a http://www.ncbi.nlm.nih.gov/sra/ oldalon, és a találati oldalon linkelt FTP könyvtárból töltsd le az `SRR111873.sra` nevű fájlt!)

▶ Installáld az SRA Toolkit szoftver csomagot a számítógépen! (Ld. http://www.ncbi.nlm.nih.gov/sra/). A fastq-dump program segítségével készíts sztenderd FASTQ adatokat a `.sra` kiterjesztésű bináris fájlból: `/fastq-dump -split-files SRR111873.sra`. Ennek az eredménye két fájl (`..._1.fastq` és `..._2.fastq`), melyek a párosított readeket tartalmazzák.

b. Read illesztés (3 pont). A readek RNA-Seq adatokat, azaz mRNS szekvenciákat tartalmaznak. A genom szekvenciához való illesztésükhöz a bwa programot használjuk. Olvasd el a bwa dokumentációját: http://bio-bwa.sourceforge.net/bwa.shtml.

▶ Installáld a bwa programcsomagot a számítógépen! A forráskód letölthető a http://bio-bwa.sourceforge.net oldalról (0.7.12 verzió).

▶ Indexeld a genom szekvenciát: `bwa index dm6.fa` (ennek eredménye 6 új fájl `dm6.fa.*` nevekkel).

▶ Illeszd a FASTQ fájlokban levő readeket:

```
bwa mem -R '@RG\tID:SRR111873\tPL:ILLUMINA' SRR111873_1.fastq SRR111873_2.fastq > ...
```

A standard outputot irányítsd át egy `.sam` kiterjesztésű fájlba. (Az `-R` switch hasznos későbbi lépésekben, az illesztések mellé tárolja a read-ek eredetét.)

c. Coverage számolás read illesztésekből (6 pont). Az illesztett readekből generálunk annotációs sávot a lefedettségről, amit aztán a UCSC Genome Browserben meg lehet vizsgálni. Ehhez kell írni egy programot, ami a SAM fájlból kiolvassa az illesztéseket.

► Olvasd át a SAM specifikációt: http://samtools.github.io/hts-specs/SAMv1.pdf.

► A genom minden egyes pozíciójában számoljuk az oda illeszkedő readeket. Ehhez a CIGAR oszlopot kell megnézni, és a CIGAR match pozícióinak megfelelő (M, =, X) pozíciókat szűrni. Az alábbi SAM formátumú sorból pl. a chr3R kromoszóma 25256242..25256293 intervallumán 1-gyel növekedik a lefedettség.

```
SRR111873.19    147     chr3R   25256242        60      52M24S ...
```

A pozíciónkénti lefedettséget legegyszerűbb egy `int[]` tömbben tárolni (mivel a genom olyan kicsi). Ehhez tervezni kell egy dinamikus adatstruktúrát, ami az ismeretlen nevű kromoszómákat (RNAME oszlopokból), és az előre nem ismert hosszakat (POS oszlopokból) kezelni tudja: a SAM inputot csak egyszer szabad olvasni.

d. Annotációs track output (6 pont). A SAM fájl olvasása után az outputot bedGraph formátumban kell kiírni: http://genome.ucsc.edu/goldenPath/help/bedgraph.html. (Vigyázz a 0/1 kezdetű koordinátákra: SAM-ban 1-gyel kezdődik, az annotációban pedig 0-val.) A szoftvernek meg kell vizsgálnia a FLAG oszlopot is, hogy csak érvényes illesztéseket számoljon. Az annotációs trackben csak a 0-tól különböző lefedettségi értékeket kell jelenteni, és az egymás utáni azonos lefedettségű pozíciókat egy szegmensben jelenteni (egy sor az outputban).

```
% java readCoverage SRR111873.sam > cov.bg
% cat cov.bg
track type=bedGraph name="SRR111873" description="Read alignment coverage"
chr2L 47807 47816 1
chr2L 47816 47883 2
chr2L 47883 47892 1
chr2L 47901 47955 1
chr2L 47955 47977 2
chr2L 47977 47986 1
chr2L 47986 47998 2
chr2L 47998 48001 3
chr2L 48001 48005 4
chr2L 48005 48031 5
chr2L 48031 48052 4
chr2L 48052 48062 5
...
```

A bedgraph formátumú fájlt fel lehet tölteni a UCSC Genome browser szerverre (Custom tracks): nézd meg, hogy hogyan változik különböző gének lefedettsége.

e. Összehasonlítás más implementációval (2 pont). Az elkészített programot összehasonlítjuk a bedtools programmal, ami sorba rendezett illesztéseken különösen hatékony.

▶ Töltsd le és installáld a samtools csomagot: http://www.htslib.org. Olvasd el a dokumentációt: http://www.htslib.org/doc/samtools-1.2.html.

▶ Konvertáld át a SAM fájlt BAM formátumra (`samtools view -b`) és rendezd sorba koordináták szerint (`samtools sort`)!

▶ Installáld a bedtools csomagot (https://github.com/arq5x/bedtools2), és használd a genomeCoverageBed programot a lefedettségi annotációs track írására: `genomeCoverageBed -ibam SRR111873-sorted.bam -bg`. Hasonlítsd össze a futási időket a nem sorba rendezett és sorba rendezett BAM fájlokon, illetve a saját implementációt a nem sorba rendezett SAM fájlon.

f. Extrapoláció (1 pont). Gondolj bele, hogy jól skálázható-a kifejlesztett implementáció emberi genomra. Mekkora lenne a memóriaigény a lefedettséget tároló adatstruktúrára?

g. Online implementáció (+5 pont bónusz). Tervezz meg, illetve implementálj egy olyan megoldást, ami sorba rendezett SAM input fájlon memóriatakarékosan tud lefedettséget számítani. (Egy új pozícióban kezdődő illesztésnél már ki lehet írni az azelőtti pozíciók lefedettségét, így azokat már nem kell tovább tárolni.)

1.3. Beadandók

A következőket kell beadni:

- ⋆ PDF dokumentumot, ami leírja részletesen a lépéseket a parancssorban kiadott utasításokkal, és a keletkező fájlok méretével, valamint tartalmazza a szöveges válaszokat (e,f,g).
- ⋆ Az implementáció Java forráskódját.
- ⋆ Az implementáció bedgraph formátumú outputját.