פרוטוקול למידה לאנליזת RNAseq

שלב ראשון: תחקיר בטחוני

כל אנליזת RNAseq תתנהל בצורה קצת שונה, בהתאם לניסוי שנערך ולמה בדיוק נבדק בו. השלב הראשון של כל אנליזה הוא ישיבה עם הביולוג שערך את הניסוי ולהבין מה נעשה בניסוי, באילו שיטות ומהם הפרמטרים ביניהם רוצים להשוות.

שאלות לדוגמה:

מה נעשה בניסוי? מה נחקר בו?

בין אילו קבוצות צריך להשוות?

האם הריצוף נעשה paired-end או מכיוון אחד? באיזו מכונה הוא נעשה, ובאיזו חברה? באילו אדפטורים השתמשנו? באיזה קיט הרצה נעשה שימוש?

*הערה: אם הריצוף היה paired-end, שם כל קובץ יופיע פעמיים: פעם אחת עם סיומת 001_ ופעם אחת עם סיומת 002_.

איזה סוג ריצוף נעשה? (של גנים מסוימים, של mRNA, של כל סוגי הRNA וכד'.)

האם מדובר ב-RNA שהופק מרקמה? מדגימת צואה? In vivo, in vitro?

האם כל דגימות הRNA הופקו באותו זמן? (אם לא- יש צורך בהסרה של batch effects).

לפי אילו פרמטרים צריך לנרמל? (כל דוגמה לקונטרול שלה, לדוגמה).

את כל הפרטים הללו יש לכתוב בצורה מסודרת בפרוטוקול האנליזה, בצורה כזו שאם מישהו יעבור על האנליזה בעוד תקופה הוא יוכל לשחזר את התוצאות.

שלב שני: הקדמה לעבודה עם לינוקס

עבודה עם לינוקס שונה מעבודה בשפות תכנות רגילות, משום שההרצה נעשית בכל פעם לפי שורה ואי אפשר לדבג. כשאנחנו רוצים להריץ שורה אחת נוכל לראות בקלות אם היא עובדת או לא, אבל בלולאות זה קצת יותר מסובך. לכן, תמיד לפני שנריץ לולאה נוודא שהיא תעבוד.

דוגמה ללולאה שנרצה להריץ:

```
for sample in $(ls N3*.sorted); do
    echo "sample: "$sample;
    htseq-count -f bam -s no --idattr=gene_name $sample
$GENOME_DIR > ${sample%%.*}_htseq_ens.txt;
done
```

את הבדיקה נעשה בכמה שלבים:

א. הלולאה עובדת על רשימה של קבצים בתוך תיקייה. דרך הגישה שלנו לקבצים תהיה עם פקודת ls על הנתיב אליהם או על שמם (אם הם ממוקמים בתיקייה הנוכחית). בשני המקרים אנחנו עושים שימוש בביטוי רגולרי, ונרצה לוודא שפקודת ls אכן מוצאת לנו את הקבצים הרצויים. בביטוי הרגולרי נרצה שהביטוי יתן לנו את כל הקבצים הרצויים ורק אותם, ובדרך כלל נעשה שימוש ב-*. פירוש הכוכבית הוא שאנחנו מוכנים לקבל כל דבר שיבוא אחריה.

בדוגמה לעיל השתמשנו בביטוי "N3*.sorted", שפירושו כל קובץ שמתחיל בN3, אחריו יכול להיות כל דבר שהוא, אבל שיסתיים ב-"sorted." .
לצורך הבדיקה נריץ את פקודת ls לבד, ונוודא שאכן מודפסים לנו הקבצים אליהם התכוונו.
בדוגמה:

```
ls N3*.sorted
```

הפלט שיוצא נראה כך:

```
[nissan@matrix old_files]$ ls N3*
N381_S1_L001_R1_001.fastq.gz  N385_S5_L001_R1_001.fastq.gz  N389_S9_L001_R1_001.fastq.gz   N393_S13_L001_R1_001.fastq.gz
N381_S1_L002_R1_001.fastq.gz  N385_S5_L002_R1_001.fastq.gz  N389_S9_L002_R1_001.fastq.gz   N393_S13_L002_R1_001.fastq.gz
N381_S1_L003_R1_001.fastq.gz  N385_S5_L003_R1_001.fastq.gz  N389_S9_L003_R1_001.fastq.gz   N393_S13_L003_R1_001.fastq.gz
N381_S1_L004_R1_001.fastq.gz  N385_S5_L004_R1_001.fastq.gz  N389_S9_L004_R1_001.fastq.gz   N393_S13_L004_R1_001.fastq.gz
N382_S2_L001_R1_001.fastq.gz  N386_S6_L001_R1_001.fastq.gz  N390_S10_L001_R1_001.fastq.gz  N394_S14_L001_R1_001.fastq.gz
N382_S2_L002_R1_001.fastq.gz  N386_S6_L002_R1_001.fastq.gz  N390_S10_L002_R1_001.fastq.gz  N394_S14_L002_R1_001.fastq.gz
N382_S2_L003_R1_001.fastq.gz  N386_S6_L003_R1_001.fastq.gz  N390_S10_L003_R1_001.fastq.gz  N394_S14_L003_R1_001.fastq.gz
N382_S2_L004_R1_001.fastq.gz  N386_S6_L004_R1_001.fastq.gz  N390_S10_L004_R1_001.fastq.gz  N394_S14_L004_R1_001.fastq.gz
N383_S3_L001_R1_001.fastq.gz  N387_S7_L001_R1_001.fastq.gz  N391_S11_L001_R1_001.fastq.gz  N395_S15_L001_R1_001.fastq.gz
N383_S3_L002_R1_001.fastq.gz  N387_S7_L002_R1_001.fastq.gz  N391_S11_L002_R1_001.fastq.gz  N395_S15_L002_R1_001.fastq.gz
N383_S3_L003_R1_001.fastq.gz  N387_S7_L003_R1_001.fastq.gz  N391_S11_L003_R1_001.fastq.gz  N395_S15_L003_R1_001.fastq.gz
N383_S3_L004_R1_001.fastq.gz  N387_S7_L004_R1_001.fastq.gz  N391_S11_L004_R1_001.fastq.gz  N395_S15_L004_R1_001.fastq.gz
N384_S4_L001_R1_001.fastq.gz  N388_S8_L001_R1_001.fastq.gz  N392_S12_L001_R1_001.fastq.gz  N396_S16_L001_R1_001.fastq.gz
N384_S4_L002_R1_001.fastq.gz  N388_S8_L002_R1_001.fastq.gz  N392_S12_L002_R1_001.fastq.gz  N396_S16_L002_R1_001.fastq.gz
N384_S4_L003_R1_001.fastq.gz  N388_S8_L003_R1_001.fastq.gz  N392_S12_L003_R1_001.fastq.gz  N396_S16_L003_R1_001.fastq.gz
N384_S4_L004_R1_001.fastq.gz  N388_S8_L004_R1_001.fastq.gz  N392_S12_L004_R1_001.fastq.gz  N396_S16_L004_R1_001.fastq.gz
```

*אלו שמות הקבצים איתם נעבוד בפרוטוקול, אז אם משהו בצורת הכתיבה של הקבצים לא מובנת- שווה לחזור לכאן לראות איך הם קרויים.

ב. בכל לולאה יהיה חלק שיעשה echo- ידפיס משהו למסך. בדרך כלל זה יהיה הקובץ עליו הלולאה עוברת באותו שלב. נרצה לוודא שהלולאה עצמה עובדת כמו שצריך ולכן נריץ אותה רק עם ההדפסה, ונראה שאכן הקבצים שרצינו לעבור עליהם מודפסים למסך בלי בעיה.
בדוגמה:

```
for sample in $(ls N3*.sorted); do
    echo "sample: "$sample;
done
```

ג. השלב הקודם נתן לנו עוד יתרון: כעת למשתנה "sample" יש ערך- הקובץ האחרון עליו רצינו לעבור. כלומר, כעת יש לנו יכולת להריץ את הפקודה על sample ולבצע אותה פעם אחת בלבד, רק על מנת לראות שהיא אכן עובדת כמו שצריך ומוציאה את התוצאות הרצויות.
בדוגמה:

```
htseq-count -f bam -s no --idattr=gene_name $sample $GENOME_DIR >
${sample%%.*}_htseq_ens.txt;
```

עכשיו, לאחר שווידאנו שהכול עובד כמו שצריך- נוכל להריץ את הלולאה בראש שקט וללכת לאכול צהריים. בתאבון!

שלב שלישי: עבודה עם FASTQ

בשלב זה נעביר את קבצי הריצוף שקיבלנו תהליכים של:

1. בדיקת איכות ב-FASTQC והצגה ויזואלית של התוצאות ב-MULTIQC
2. ביצוע trimming ב-Trimmomatic
3. ביצוע Alignment לגנום ב-STAR
4. ספירת גנים ב-HTSEQ

אנחנו מתחילים את התהליך עם קבצי הFASTQ שקיבלנו מהריצוף באילומינה ומסיימים אותו עם טבלת CSV שמכילה את מספר הקריאות שקיבלנו מכל גן. בשלב זה נשתמש בתוכנות שונות, ומומלץ לקרוא על התהליכים שנעשים בכל אחת מהן לפני שניגשים לעבודה עצמה, על מנת להבין מה נעשה בכל שלב.

בשלב זה נעבוד עם הקבצים בלינוקס, בתוכנה בשם MobaXterm.

לצורך הפרוטוקול ניתנת כאן דוגמה לData של ריצוף ספציפי של RNA, אך השתדלתי שהקוד עצמו יהיה כמה שיותר גנרי על מנת שיהיה אפשר להשתמש בו כמעט במלואו. חלקים שיהיו פחות גנריים הם בדרך כלל המשתנים עליהם הלולאות רצות (שמכילים בפועל את שם הקבצים), והפרמטרים שהפונקציות קיבלו, שברובם נשארים קבועים.

האנליזה שמובאת כאן היא דוגמה עם הסברים. את הסקריפטים המוכנים שניתן להריץ ניתן למצוא בסוף השלב הזה.

בסופו של דבר עבור כל אנליזה ייווצרו קבצים זהים עברו כל אחד מהשלבים. החלק היחיד שישתנה יהיה הנתיב עד לאנליזה הספציפית הזו, ואותו התוכנה תקבל כקלט. כשנעשה שימוש במשתנה INPUT- זהו הנתיב לתיקייה שמכילה את האנליזה הספציפית שלנו.

Unix_commands_dictionary.pdf

מדריך פקודות בלינוקס באדיבות דוד גורליק:

כמובן שלא כל הפקודות מופיעות בו, וניתן להיעזר במדריכים מקוונים אחרים.

ראשית יש להוריד את קבצי הריצוף מהאתר של אילומינה ולשמור אותם בדרייב.

התחברות ללינוקס דרך Mobaxterm:

בשרת SSH: matrix.lnx.biu.ac.il

שם משתמש: [redacted]

סיסמה: [redacted]

מגדירים את הSHELL ואת הקלט:

```
#set the shell
bash
INPUT="$1/"
```

יוצרים תיקייה חדשה בנתיב הרצוי וגוררים אליה את כל קבצי הFASTQ שקיבלנו מהריצוף.

בדיקת ואלידציה:

עושים בדיקת ואלידציה על כל אחד מהקבצים באמצעות FASTQC:

(השורות הלבנות הן הסברים על חלקים בקוד)

```
`Validation Check`
#fastqc
WANTED_DIR="${INPUT}fastq/";
OUTPUT_DIR="${INPUT}fastqc_files/";
mkdir -p $WANTED_DIR;
mkdir -p $OUTPUT_DIR;
cd $WANTED_DIR;
#get into the directory with the fastq files
for sample in $(ls -R *.fastq.* ); do
    echo $sample;
    fastqc $sample -o $OUTPUT_DIR;
    echo -e "Finished\n"
done &>${OUTPUT_DIR}fastqc.txt &
#run multiqc on all the files
multiqc $OUTPUT_DIR &>multiqc.txt&
```

בסיום הלולאה מופיע הסימון &, שמסמן לשרת לא להדפיס את התקדמות התהליך. הוספנו את הסימון <, שאומר לשרת לכתוב את התקדמות התהליך לקובץ במקום זה, ובו נוכל לבדוק איפה אנחנו נמצאים בכל התהליך. הבעיה שזה יוצר היא שלא נוכל להרוג את התהליך עם ctrl+C כמו בדרך כלל, ואם נרצה לעצור את הלולאה באמצע נצטרך את מספר התהליך. זה המספר שקיבלנו כפלט מהשרת כשהוא התחיל את הלולאה. אם נרצה לבדוק אם התהליך רץ נוכל להשתמש ב-jobs או ב-top, ואם נרצה להרוג אותו נשתמש ב- pkill -u number .

בסיום התהליך נקבל קובץ אחד שבו יש פרמטרים שונים לבדיקת איכות עבור קבצי הריצוף שלנו.

הקובץ הזה אינטראקטיבי (יש לפתוח בפורמט Microsoft Edge), וניתן לראות בו עבור כל פרמטר מה הוא אומר. בלשונית ה-Help יש הסבר על כל פרמטר, ומה נצפה לקבל בו. החלקים עליהם מסתכלים בגדול הם בדיקת האיכות (רצוי- כל הרצפים באזור הירוק), כמות הרידים (לשים לב שאין דוגמה עם כמות רידים נמוכה משמעותית מהיתר) כמות GC (רצוי- התפלגות נורמלית סביב 50), כמות N (=בסיסים לא מזוהים, רצוי- כל הרצפים באזור הירוק), רצפים מיוצגים יתר על המידה (אם הם קיימים אלה בד"כ יהיו אדפטורים).

*התוכנה מגדירה רצף חוזרני כרצף בעל אחוז דמיון גבוה לרצף אחר, ולא כשני רצפים שזהים לחלוטין זה לזה. כלומר, טבעי שיהיו הרבה כאלה ואין לכך משמעות.

Trimming:

אם לאחר בדיקת האיכות נרצה "לגזום" חלק מהרידים, נשתמש בתוכנה בשם Trimmomatic.
התוכנה עושה:

הסרת אדפטורים, הסרת בסיסים בעלי ציון ודאות נמוך, והסרה של רידים קצרים.

```
`Trimmomatic`
IN_DIR="${INPUT}fastq/"
OUTPUT_DIR_FASTQC="${INPUT}fastqc_trimmed/"
OUTPUT_DIR_TRIM="${INPUT}trimmed_files/"
mkdir -p $OUTPUT_DIR_FASTQC
mkdir -p $OUTPUT_DIR_TRIM
cd $IN_DIR
for sample in $(ls -R *.fastq.gz); do
    echo $sample;
    java -jar /private/software/packages/Trimmomatic-
0.39/trimmomatic-0.39.jar SE -threads 3 -phred33 $sample
"$OUTPUT_DIR_TRIM${sample%%.*}_trimmed.fastq.gz"
ILLUMINACLIP:/home/stu/nissan/Tamar/imported_data/TruSeq-
All_Bili_adaptors.fa:2:30:10:2 LEADING:3 TRAILING:3
MINLEN:20
done &> trim.txt&
```

לאחר מכן נרצה לבצע שוב FASTQC על מנת לבדוק שהאיכות שלנו אכן עלתה.

```
#re-run fastqc
cd $OUTPUT_DIR_TRIM
for sample in $(ls -R *.fastq.* ); do
    echo $sample;
    fastqc $sample -o $OUTPUT_DIR_FASTQC;
    echo -e "Finished\n"
done &>${OUTPUT_DIR_FASTQC}fastqc.txt &
#run multiqc on all the files
multiqc $OUTPUT_DIR
```

לאחר מכן נוכל להשוות בין שני קבצי ה-MultiQC שקיבלנו ולהחליט האם אנחנו מעדיפים להמשיך את העבודה עם הקבצים המקוריים או אלה שעברו טרימינג.

נשים לב שאף על פי שטרימינג מעלה את איכות הריצוף שלנו, הוא גם גורם לרידים בעלי אורך לא אחיד, ועל כן אם איכות הריצוף ההתחלתית היתה טובה ייתכן שנחליט להמשיך את האנליזה עם הקבצים המקוריים.

<u>הסרה ידנית של קבצים:</u>

אם לאחר הטרימינג עדיין יש קבצים שרוצים להסיר, נעשה זאת כך:

```
`Delete Lines Or Samples`
#if there is a bad sample or line we should remove it from the other
files
cd WANTED_DIR;
```

```
mkdir -p ../invalid_files
BAD_FILES="N397*.fastq.*"
for sample in $(ls -R ${WANTED_DIR}/$BAD_FILES); do
    echo $sample;
    mv $sample ../invalid_files
done
```

זה לא מצב שאמור לקרות בעיקרון, אבל עדיין טוב שיש את הקוד בצד למקרה הצורך.

Alignment:

לאחר שנותרנו רק עם הקבצים המשמעותיים, נרצה לעשות להם alignment לגנום על מנת למצוא את שמות הגנים שלהם. לשם כך נשתמש ב-STAR.

```
`STAR`
#star
WANTED_DIR="${INPUT}STAR/";
IN_DIR="${INPUT}fastq";
mkdir -p $WANTED_DIR
cd $IN_DIR;
genomeDir="/home/stu/nissan/software/STAR/STARgenomes/E
NSEMBL/mus_musculus/ENSEMBL.mus_musculus.GRCm38.noAnnot
";
for sample in $(ls -R *.fastq.gz); do
    echo $sample;
    echo ${sample%%.*};
    /private/software/bin/STAR --genomeDir $genomeDir -
-runThreadN 6 --readFilesIn <(gunzip -c $sample) --
outFileNamePrefix "$WANTED_DIR/${sample%%.*}_" --
outSAMtype BAM SortedByCoordinate --outSAMunmapped
Within --outSAMattributes Standard;
done &>star.log&
```

ספירת רידים ומיון:

קיבלנו את תוצאות ההתאמה מ-STAR. כעת נרצה לספור את כמות הרידים מכל דגימה שעברו Alignment, ונעשה זאת באמצעות הכלי view שקיים ב-samtools.

```
`READ COUNT`
#read count
IN_DIR="${INPUT}STAR/"
PROCESS_FILE="RNA_Seq_count.txt"
cd $IN_DIR
for count in $(ls *Aligned.sortedByCoord.out.bam); do
    echo "count:"$count;
```

```
    /private/software/bin/samtools view -c -F 4 $count
;
done &> $PROCESS_FILE&
```

נעבור על הקובץ שהתקבל ונבדוק שקיבלנו תוצאות טובות. אם באחת מהדגימות קיבלנו מספר נמוך מאוד אף על פי שבMultyQC קיבלנו כמות רידים טובה- ייתכן שהיתה בעיה בהרצה של STAR על אותה דגימה וננסה להריץ אותו עליה שוב.

נמיין את הגנים שהתקבלו לפי המיקום הכרומחומלי שלהם, באמצעות sort של samtools.

```
`BAM SORT`
#bam sort
IN_DIR="${INPUT}STAR/"
PROCESS_FILE="RNA_Seq_sort.txt"
OUTPUT_DIR="${INPUT}bam_sorted/"
mkdir -p $OUTPUT_DIR;
cd $IN_DIR;
for bamFile in $(ls *Aligned.sortedByCoord.out.bam); do
    echo "sort: "$bamFile;
    /private/software/bin/samtools sort $bamFile -o
${bamFile%%.*}.sorted -@ 4;
done &>$PROCESS_FILE&
mv *.sorted $OUTPUT_DIR
```

<u>ספירת הגנים:</u>

כעת יש לנו קובץ עבור כל דגימה, בו יש רשימה של כל הגנים שמתבטאים באותה הדגימה. נרצה לספור את הגנים בכל דוגמה, כך שכל גן יופיע בקובץ פעם אחת בלבד, עם מספר הפעמים בו הוא הופיע, ונעשה זאת עם HTSEQ.

```
`HTSEQ`
#HTSeq
IN_DIR="${INPUT}bam_sorted/"
OUTPUT_DIR="${INPUT}counts"
mkdir -p $OUTPUT_DIR
GENOME_DIR="/home/stu/nissan/software/htseq_genome/Mus_
musculus.GRCm38.99.gtf"
cd $IN_DIR
for sample in $(ls *.sorted); do
    echo "sample: "$sample;
    htseq-count -f bam -s no --idattr=gene_name $sample
$GENOME_DIR > ${sample%%.*}_htseq_ens.txt;
done &>htseq_ens.log&
mv *_ens.txt $OUTPUT_DIR
```

נשנה את שמות הקבצים לשמות שנרצה שיהיו לנו ככותרות בטבלה בסופו של דבר:

```
#change files names
for sample in $(ls *.txt); do
    echo $sample;
    mv -- "$sample" "${sample%_R1*}.txt";
done
```

(מה שנעשה כאן בעצם הוא מחיקה של כל דבר שמופיע אחרי הביטוי "_R1", כולל הביטוי עצמו.
לאחר פעולה זו נשארנו רק עם השם של כל דגימה + מספר הליין שלה)

ואז נאחד את קבצי ה-HTSEQ לקובץ יחיד:

```
#merge the counts to one file
INPUT_FILE="countdata"
COUNTS_DIR="${INPUT}counts"
SCRIPT_DIR="/home/stu/nissan/Tamar/imported_scripts/hts
eq-merge_all.R"
Rscript $SCRIPT_DIR $COUNTS_DIR $INPUT_FILE
```

<u>המרת הקובץ לטבלה:</u>

נמיר את קובץ הcounts המאוחד לטבלת CSV (כמו Excel):

```
#turn to csv
INPUT_FILE="countdata.txt"
OUTPUT_FILE="countdata.csv"
sed -e 's/[[:space:]]\{1,\}/,/g' $INPUT_FILE >
$OUTPUT_FILE
```

את הקובץ הזה נשמור בדרופבוקס, ואנחנו מוכנים לאנליזת הData ב-R!

מחיאות כפיים!

לאחר הרצת סקריפט זה נוצר קובץ CSV שניתן להוריד למחשב ולהמשיך איתו לאנליזה.

פקודת הרצה לסקריפט:

```
chmod 755 $script_name

$script_name
```

שני הסקריפטים יוצרים עץ תיקיות שנראה כך:

Input

fastq_files

#fastq files

fastqc_files

#fastqc files

trimmed_files

trimmed files

STAR

#aligned files

bam_sorted

#sorted BAM files

counts

#HTseq files

כלומר בפועל לכל אנליזה נוצרות 6 תיקיות שמכילות את כל הקבצים.

קובץ ה-CSV נוצר בתיקיית counts.

שלב רביעי: אנליזה ב-R

מדריך לאנליזה של RNASeq של DESeq2 שאפשר להיעזר בו:
https://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html

גם בחלק זה נעשה אנליזה של RNA על Data מסוים, אבל ננסה לעשות אותו גנרי ככל האפשר.

בחלק זה אנו עושים שימוש בשתי טבלאות:

Count data- הטבלה בה מופיעה רמת הביטוי של הגנים של כל הדוגמאות

Meta data- טבלה שמכילה את המידע על הדגימות השונות בניסוי: ID של כל דגימה, מין, סוג טיפול וכדומה. בטבלה צריכים להופיע כל ההבדלים בין הקבוצות- אם הניסויים נערכו בזמנים שונים, לדוגמה, נרצה לסווג אותם לפי זה כדי לבדוק שלא היתה הטיה לאחד מהניסויים עקב כך.

ספריות:

על מנת להשתמש בחבילה ב-R צריך להוריד אותה למחשב פעם אחת, ואז לייבא אותה בתור ספריה. את פקודת ה- install.packages("package") נעשה תמיד בקונסול ולא בסקריפט, כדי שלא נוריד את החבילה מחדש בכל פעם שנרצה להריץ את הסקריפט.

אם רוצים להשתמש בחבילה ולא בטוחים אם היא כבר קיימת אצלו או לא, ניתן להריץ את הפקודה לייבוא החבילה ("library(package)"), ואם היא לא קיימת על המחשב תתקבל שגיאה.

קוד מסודר לאנליזת RNA-Seq ב-R נמצא בתיקייה הסמוכה, בה יש קוד ספציפי לאנליזה מסוימת, אבל אפשר להמיר אותו בקלות לאנליזות אחרות.