Preparatory Course - Bioinformatics at the DKFZ

Coding Exercises Day 1

Exercise content:
- Data formats: GENCODE, BED
- Programming languages: Shell/Bash, Python, R
- Tools: bedtools, qsub, qstat
- GUIs: UCSC Genome Browser

Research question:

Which transcription factor binding sites are better conserved in CpG island promoters?

Protocol

Step 1: Download GENCODE V19 annotation. To which genome assembly does this annotation match? _____

Step 2: Generate a BED file that contains the coordinates of each core promoter of the GENCODE annotation. The name field should be the Hugo Gene Symbol and the Ensembel ID connected by underscore (e.g. *TP53_ENSG00000141510*). The core promoter is here defined as 500 bp upstream to 100 bp downstream from the TSS. How do you account for the strand each gene is located on?

_____

_____

_____

Step 3: Classify the promoters into two groups by overlapping them with the UCSC CpG island annotation using *bedtools intersect*. How often do you had to compute the overlap? _____

Which file have you chosen as -a file and which as -b file and why?

Bonus task: Write a python program with the same functionality. Does it have an advantage ? _____

Step 4: Sort both promoter databases into files by their chromosomes and output these to the subfolders CgiProm and NonCgiProm.

Step 5: Download the Conserved TFBS sites track from the UCSC GenomeBrowser and reformat it into a BED file by using the shell command *cut*. Note the resulting command: _____

Step 6: Write a shell script that takes the path of a BED file and the name of an outputfile as input. The script should execute an *bedtools intersect* of the inputfile and the Conserved TFBS sites track and write for each overlapping promoter/TFBS pair into the output file. How have you names the variables that stored the input parameters?
Variable 1: _____
Variable 2: _____

Step 7: Write down the *qsub* command to test the script on one chromosome file:

Which chromosome have you selected and why? _____

Step 8: Create a subdirectory called *logs* and and use the -e and -o option to write all error messages to this path. Execute the resulting *qsub* command twice and introduce a mistake in the input path in the second execution. What result do you get?

Step 9: Write a script either in Shell or Python that takes a directory path as input and writes for each file in this directory an according *qsub* command to *stdout*.

Step 10: Collect all qsub commands into the runscript run.sh by using the redirect '>'and execute the resulting script.

Step 11: Join the resulting files for each promoter type using *cat*.

Step 12: Load results into R using *read.table*. Analyze data and answer research question!

Resources:

GENCODE V19 annotation:
https://www.gencodegenes.org/releases/19.html

CpG island annotation:
http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/cpgIslandExt.txt.gz

Conserved TFBS sites:
http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/tfbsConsSites.txt.gz