**GMQL for the analysis of (epi)genomic data features**

Ex. 1 – Find the genomic regions of human genome GRCh38 covered by signal peaks of both H3K4me1 and H3K27ac histone marks, which are potential active enhancers, in the HCT116 human cell line. Use ENCODE *narrowpeak* data (dataset of November 2017) regarding *experiment target* H3K4me1 and H3K27ac.

Then, count the number of regions in the input data and in the result.

Report obtained sample and region numbers, and discuss the results.

Version 1

#1 Loading the data

H3K4me1 = SELECT(biosample\_term\_name == 'HCT116' AND experiment\_target == 'H3K4me1-human') GRCh38\_ENCODE\_NARROW\_NOV\_2017;

H3K27ac = SELECT(biosample\_term\_name == 'HCT116' AND experiment\_target == 'H3K27ac-human') GRCh38\_ENCODE\_NARROW\_NOV\_2017;

#2 Combining multiple replicates in a single sample

H3K4me1\_1 = COVER(1, ANY) H3K4me1;

H3K27ac\_1 = COVER(1, ANY) H3K27ac;

#3 Generating single input dataset

K4me1\_K27ac = UNION() H3K4me1\_1 H3K27ac\_1;

#4 Extracting the portion of intersecting regions

RES\_0 = COVER(2, ANY) K4me1\_K27ac;

#5 Counting output regions and saving the result

RES = EXTEND(\_RegionCountOutput AS COUNT()) RES\_0;

MATERIALIZE RES INTO res;

#6 Counting and saving input regions

HM\_0 = UNION() H3K4me1 H3K27ac;

HM = EXTEND(\_RegionCountInput AS COUNT()) HM\_0;

MATERIALIZE HM INTO HM;

**Execution time**: 00:03:53

RES samples: 1

RES regions: 93 562

HM samples: 16

HM regions: 1 490 710

Version 2

#1 Loading the data

H3K4me1 = SELECT(biosample\_term\_name == 'HCT116' AND experiment\_target == 'H3K4me1-human') GRCh38\_ENCODE\_NARROW\_NOV\_2017;

H3K27ac = SELECT(biosample\_term\_name == 'HCT116' AND experiment\_target == 'H3K27ac-human') GRCh38\_ENCODE\_NARROW\_NOV\_2017;

#2 Combining multiple replicates in a single sample

H3K4me1\_1 = COVER(1, ANY) H3K4me1;

H3K27ac\_1 = COVER(1, ANY) H3K27ac;

#3 Extracting the portion of intersecting regions

RES\_0 = JOIN(distance < 0; output: INT) H3K4me1\_1 H3K27ac\_1;

#4 Counting output regions and saving the result

RES = EXTEND(\_RegionCountOutput AS COUNT()) RES\_0;

MATERIALIZE RES INTO res;

#5 Counting and saving input regions

HM\_0 = UNION() H3K4me1 H3K27ac;

HM = EXTEND(\_RegionCountInput AS COUNT()) HM\_0;

MATERIALIZE HM INTO HM;

**Execution time**: 00:03:50 (00:04:33 without initial covers)

RES samples: 1 (64 without initial covers)

RES regions: 93 562 (3 326 076 without initial covers)

HM samples: 16

HM regions: 1 490 710

Ex. 2 – Identify HG19 DNA regions bound by the JUN transcription factor that overlap binding sites of FOS transcription factor, which could indicate co-binding events and provide information for finding functionally related factors. Use ENCODE *narrowpeak* data (dataset of November 2017) regarding *experiment target* JUN and FOS in the HeLa-S3 cell line; particularly, use only the single sample data produced by the *lab* '*ENCODE Consortium Analysis Working Group*'.

Then, count the number of regions in the input data and in the result.

Report obtained sample and region numbers, and discuss the results.

#1 Loading the data

FOS = SELECT(biosample\_term\_name == 'HeLa-S3' AND experiment\_target == 'FOS-human' AND lab == '*ENCODE Consortium Analysis Working Group*') HG19\_ENCODE\_NARROW\_NOV\_2017;

JUN = SELECT(biosample\_term\_name == 'HeLa-S3' AND experiment\_target == 'JUN-human' AND lab == '*ENCODE Consortium Analysis Working Group*') HG19\_ENCODE\_NARROW\_NOV\_2017;

#2 Identifying the JUN binding regions overlapping with FOS binding regions

RES\_0 = JOIN(distance < 0; output: LEFT\_DISTINCT) FOS JUN;

#3 Counting output regions and saving the result

RES = EXTEND(\_RegionCountOutput AS COUNT()) RES\_0;

MATERIALIZE RES INTO res;

#4 Counting and saving input regions

TF\_0 = UNION() FOS JUN;

TF = EXTEND(\_RegionCountInput AS COUNT()) TF\_0;

MATERIALIZE TF INTO TF;

**Execution time**: 00:02:06

RES samples: 1

RES regions: 7692

TF samples: 2

TF regions: 31 228

Ex. 3 - Identify DNA promotorial regions bound by the MYC transcription factor that present somatic mutations (if any) in breast cancer patients. Use narrowpeak data from the ChIP-seq *assay* of MYC binding sites for cell line MCF-7 from ENCODE (HG19 dataset of November 2016), DNA-seq data from TCGA breast cancer (BRCA *tumor tag*) patients with clinical *followup reason* *'additional new tumor event*', and genomic region annotations data (promoter, TSS, gene) from RefSeq (dataset HG19\_BED\_ANNOTATION).

Then, for each selected patient count the number of MYC bound promoters with patent’s mutations.

Report obtained sample and region numbers, and discuss the results.

#1 Loading data

MYC = SELECT(assay == 'ChIP-seq' AND biosample\_term\_name == 'MCF-7' AND

experiment\_target == 'MYC-human') HG19\_ENCODE\_NARROW\_NOV\_2016;

BRCA = SELECT(manually\_curated\_\_tumor\_tag == 'brca' AND clinical\_follow\_up\_\_followup\_reason == 'additional new tumor event') HG19\_TCGA\_dnaseq;

# PROM = SELECT(annotation\_type == 'promoter') HG19\_BED\_ANNOTATION;

GENE = SELECT(provider == 'RefSeq' AND annotation\_type == 'gene') HG19\_BED\_ANNOTATION;

# PROM = PROJECT(region\_update: start AS start - 2000, stop AS start + 1000) GENE;

#MATERIALIZE GENE INTO GENE;

# In the input GENE dataset for the same gene there can be multiple regions with

# different gene IDs.

# So, calculating directly the promoter regions I can obtain multiple replicate

# regions for the same promoter.

# To avoid it, I can perform the following operations

TSS = PROJECT(region\_update: stop AS start + 1) GENE;

TSS\_1 = COVER(1, ANY; aggregate: BAG(name)) TSS;

PROM = PROJECT(region\_update: start AS start - 2000, stop AS start + 1000) TSS\_1;

#2 Combining MYC replicates

MYC\_1 = COVER(1, ANY) MYC;

#3 Extracting MYC-binding promoters

# PROM\_MYC = JOIN(distance < 0; output: left\_distinct) PROM MYC\_1;

PROM\_MYC\_0 = MAP() PROM MYC\_1;

PROM\_MYC = SELECT(region: count\_PROM\_MYC\_1 > 0) PROM\_MYC\_0;

#MATERIALIZE PROM\_MYC INTO PROM\_MYC;

#4 Extracting BRCA-mutated MYC-binding promoters

# PROM\_MYC= MERGE() PROM\_MYC\_0;

# Using here MERGE instead of COVER before is not the proper solution since it

# maintains several replicated regions in the MERGE output coming from the

# different MERGE input samples

PROM\_MYC\_BRCA = JOIN(distance < 0; output: left\_distinct) PROM\_MYC BRCA;

#PROM\_MYC\_BRCA\_0 = MAP() PROM\_MYC BRCA;

#PROM\_MYC\_BRCA = SELECT(region: count\_PROM\_MYC\_BRCA > 0) PROM\_MYC\_BRCA\_0;

#5 Counting BRCA-mutated MYC-binding promoters for each patient

RES = EXTEND(\_prom\_count AS COUNT()) PROM\_MYC\_BRCA;

MATERIALIZE RES INTO RES;

**Execution time**:00:03:23

RES samples: 3

RES regions: 6

Ex. 4 – Differential binding search. Answer the following questions using HG19 ENCODE *narrowpeak* data with *output type* ‘*optimal idr thresholded peaks*’regarding JUN *experiment target* from the ChIP-seq *assay* (dataset of November 2017) for cell line k562 (chronic myelogenous leukemia), with 1 control and 2 cases treated using interferon alpha or gamma, respectively, each of the latter ones with two replicas. Comment required GMQL statement parameters not specified in the exercise text.

Report obtained sample and region numbers, and discuss the results.

Question 1: Combining replicate samples (if existing) by considering only regions in at least two replicas and calculating the minimum pvalue of the combined regions, find those JUN binding regions in each treated cases where there are no JUN binding regions in the control case.

# Select the control

JUN = SELECT(assay == 'ChIP-seq' AND experiment\_target == 'JUN-human' AND

biosample\_term\_name == 'K562' AND

output\_type == 'optimal idr thresholded peaks' AND

NOT (biosample\_treatments == '\*')) HG19\_ENCODE\_NARROW\_NOV\_2017;

# Only one control sample is selected with the used selection parameters

#JUN = COVER(2, ANY; aggregate: pValue AS MIN(pvalue)) JUN\_2;

# Select treated samples

JUN\_IFNa\_2 = SELECT(assay == 'ChIP-seq' AND experiment\_target == 'JUN-human' AND biosample\_term\_name == 'K562'

AND output\_type == 'optimal idr thresholded peaks'

AND biosample\_treatments == 'interferon alpha') HG19\_ENCODE\_NARROW\_NOV\_2017;

JUN\_IFNg\_2 = SELECT(assay == 'ChIP-seq' AND experiment\_target == 'JUN-human' AND biosample\_term\_name == 'K562'

AND output\_type == 'optimal idr thresholded peaks'

AND biosample\_treatments == 'interferon gamma') HG19\_ENCODE\_NARROW\_NOV\_2017;

# Extract the treated regions in at least two replicas and their minimum p-value

JUN\_IFNa\_3 = COVER(2, ANY; aggregate: min\_pValue AS MIN(pvalue)) JUN\_IFNa\_2;

JUN\_IFNg\_3 = COVER(2, ANY; aggregate: min\_pValue AS MIN(pvalue)) JUN\_IFNg\_2;

# Extract treated regions which do not overlap with control ones (Question 1)

#JUN\_IFNa = JOIN(DGE(0), MD(100); output: left\_distinct) JUN\_IFNa\_3 JUN;

#JUN\_IFNg = JOIN(DGE(0), MD(100); output: left\_distinct) JUN\_IFNg\_3 JUN;

# MD() condition is needed to have a well-formed genometric clause

# Subjective value 100 can influence final results

# The JOIN(DGE(0), MD(100); ...) solution is wrong since for each reference region

# the extracted regions could overlap other reference regions ....

JUN\_IFNa = DIFFERENCE() JUN\_IFNa\_3 JUN;

JUN\_IFNg = DIFFERENCE() JUN\_IFNg\_3 JUN;

MATERIALIZE JUN INTO JUN;

MATERIALIZE JUN\_IFNa INTO JUN\_IFNa;

MATERIALIZE JUN\_IFNg INTO JUN\_IFNg;

**Execution time**:00:02:54

JUN samples: 1

JUN regions: 9 848

JUN\_IFNa samples: 1

JUN\_IFNa regions: 454 (2 979 with JOIN(DGE(0), MD(100); ...))

JUN\_IFNg samples: 1

JUN\_IFNg regions: 1 121 (4 002 with JOIN(DGE(0), MD(100); ...))

Question 2: Find the JUN binding regions that are differentially present between treated cases (e.g., in alpha and not in gamma interferon treated samples) and not in control.

# All treated samples: union of all treated regions non-overlapping with control ones

JUN\_IFNag = UNION() JUN\_IFNa JUN\_IFNg;

# Regions that are present in treatment alpha and not in treatment

# gamma (Question 2)

# A possible solution:

#JUN\_IFNag\_ONLYa\_0 = MAP() JUN\_IFNa JUN\_IFNg;

#JUN\_IFNag\_ONLYa = SELECT(region: count\_ JUN\_IFNa\_ JUN\_IFNg == 0) JUN\_IFNag\_ONLYa\_0;

# or better:

JUN\_IFNag\_ONLYa = DIFFERENCE() JUN\_IFNa JUN\_IFNg;

MATERIALIZE JUN\_IFNag\_ONLYa INTO JUN\_IFNag\_ONLYa;

# Regions that are present in treatment gamma and not in treatment

# alpha (Question 2)

JUN\_IFNag\_ONLYg = DIFFERENCE() JUN\_IFNg JUN\_IFNa;

MATERIALIZE JUN\_IFNag\_ONLYg INTO JUN\_IFNag\_ONLYg;

**Execution time**:00:02:43 (cumulated)

JUN\_IFNag\_ONLYa samples: 1

JUN\_IFNg\_ONLYa regions: 258 (729 with JOIN(DGE(0), MD(100); ...))

JUN\_IFNag\_ONLYg samples: 1

JUN\_IFNg\_ONLYg regions: 925 (1 755 with JOIN(DGE(0), MD(100); ...))

Question 3: Find the RefSeq genes (dataset HG19\_BED\_ANNOTATION) that overlap with JUN binding regions either in control or treated samples.

# Extract all genes

GENE = SELECT(annotation\_type == 'gene' AND provider == 'RefSeq') HG19\_BED\_ANNOTATION;

# All samples: union of regions in controls or in treated samples

JUN\_ALL\_0 = UNION() JUN JUN\_IFNag;

# Combine all JUN control or treated regions in a single sample

JUN\_ALL = COVER(1, ANY) JUN\_ALL\_0;

# Extract genes intersecting with regions in control or treated samples (Question 3)

#GENE\_JUN\_ALL = JOIN(distance < 0; output: left\_distinct) GENE JUN\_ALL;

GENE\_JUN\_ALL\_0 = MAP() GENE JUN\_ALL;

GENE\_JUN\_ALL = SELECT(count\_GENE\_JUN\_ALL > 0) GENE\_JUN\_ALL\_0;

MATERIALIZE GENE\_JUN\_ALL INTO GENE\_JUN\_ALL;

**Execution time**:00:02:44

GENE\_JUN\_ALL samples: 0

GENE\_JUN\_ALL regions: 0

Question 4: In each interferon alpha and gamma treated and control condition, find the overall average JUN ChIP-seq signal in gene promoters (areas surrounding the TSS) intersecting JUN binding regions either in control or treated samples.