GMQL - Biological examples

This section of GMQL documentation collects several examples where GMQL is used to answer practical questions/tasks of biological and clinical interest. For each example, after an initial textual statement describing the question/task to be answered, the GMQL query that answers it is reported together with a detailed commented description of the query and its results.

1. Find distal bindings in transcription regulatory regions

"*Find all enriched regions (peaks) of CTCF transcription factor (TF) in ENCODE ChIP-seq narrow peak samples from GM12878 lymphoblastoid human cell line which are the nearest regions farther than 100 kb from a transcription start site (TSS). For the same cell line, find also all peaks of the H3K4me1 histone modification (HM) which are also the nearest regions farther than 100 kb from a TSS. Then, out of the TF and HM peaks found in the same cell line, return all TF peaks that overlap with at least a HM peak and known enhancer (EN) region.*"

```
TF = SELECT(assay == "ChIP-seq" AND output_type == "peaks" AND
        experiment_target == "CTCF-human" AND
        biosample_term_name == "GM12878") HG19_ENCODE_NARROW_NOV_2017;
HM = SELECT(assay == "ChIP-seq" AND output_type == "peaks" AND
        experiment_target == "H3K4me1-human" AND
        biosample_term_name == "GM12878") HG19_ENCODE_NARROW_NOV_2017;
TSS = SELECT(annotation_type == "TSS" AND provider == "UCSC") HG19_BED_ANNOTATION;
EN = SELECT(annotation_type == "enhancer" AND provider == "UCSC")
        HG19_BED_ANNOTATION;

TF1 = JOIN(DISTANCE > 100000, MINDISTANCE(1); output: RIGHT_DISTINCT) TSS TF;
HM1 = JOIN(DISTANCE > 100000, MINDISTANCE(1); output: RIGHT_DISTINCT) TSS HM;
HM2 = JOIN(DISTANCE < 0; output: INT) EN HM1;
HM3 = MERGE() HM2;
TF_RES = JOIN(DISTANCE < 0; output: RIGHT_DISTINCT) HM3 TF1;

MATERIALIZE TF_RES INTO TF_RES;
```

This example, whose context is illustrated in Figure 1, shows that GMQL is a powerful expressive language to answer frontier epigenomics questions on entire genomic datasets. Out of all 10,342 ENCODE ChIP-seq narrow peak samples available on November 2017 (containing a total of 1,604,183,681 enriched regions), initially the GMQL query selects 2,136,849 TF regions from 10 samples and 328,949 HM regions from 3 samples, in both cases of the *GM12878* lymphoblastoid human cell line. It also selects 131,780 TSSs and 1,339 enhancer regions from the *HG19_BED_ANNOTATION* dataset, which are provided by the UCSC database (https://genome.ucsc.edu/cgi-bin/hgTables) from SwitchGear Genomics (http://switchgeargenomics.com/) and Vista Enhancer (http://enhancer.lbl.gov/), respectively.

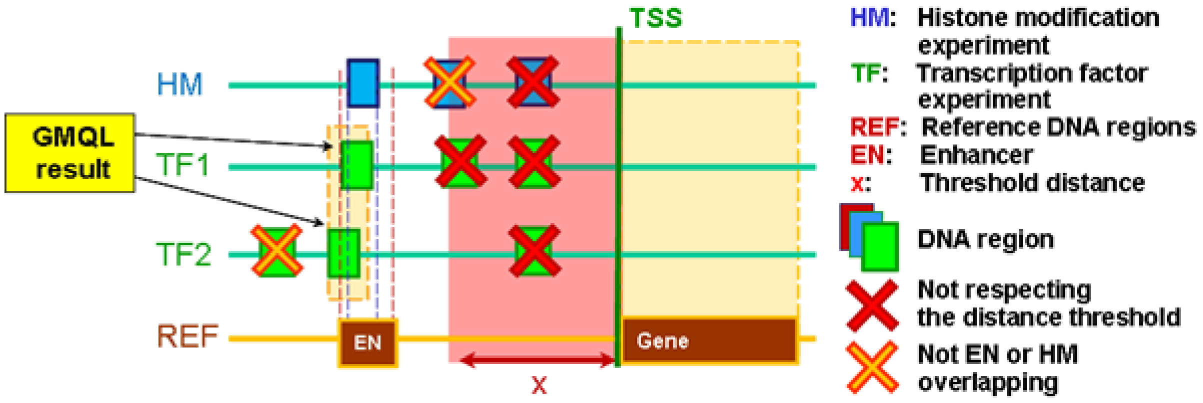
TSS
HM
GMQL result
TF1
TF2
REF
EN
Gene
x
HM: Histone modification experiment
TF: Transcription factor experiment
REF: Reference DNA regions
EN: Enhancer
x: Threshold distance
DNA region
Not respecting the distance threshold
Not EN or HM overlapping

Figure 1. The histone modification (HM) and transcription factor (TF) binding site enriched regions ('peaks'), known reference DNA regions and their distance relationships involved in Example 1.

Then, for each sample, the GMQL query computes in *TF1* the TF regions that are at minimal distance from a TSS, provided that such distance is greater than 100,000 bases, and in *HM1* the HM regions that also are at minimal distance from a TSS, with the same constraint. Note that, in addition to the distal condition, the JOIN parameter also indicates that the result must include only the distinct matching regions of the right sample, *TF* and *HM* respectively.

Next, in *HM2* for each sample the query computes the intersection between those *HM1* enriched regions and the enhancer regions in *EN*. Then, it merges in a single sample in *HM3* all regions in all *HM2* samples. Finally, it computes in *TF_res* the enriched regions in each *TF1* sample that intersect with at least a region in *HM3*, producing in the result the matching regions of the right *TF1* sample.

At the time of writing, the query extracted 10 TF samples containing a total of 49 enriched binding regions. Its execution required only 238 seconds, a short time considering the complexity of the task and the multiple implicit iterations on the many samples and regions initially considered.

The join operation with *distance* and *mindistance* functions highlights the power of GMQL in performing genometric evaluations in batch on multiple samples at a time; they are normally performed by executing data manipulation scripts, developed by individual researchers in different programming languages.

2. Find exons with somatic mutations

"Consider all public somatic mutation data samples of TCGA Kidney Renal Clear Cell Carcinoma patients. For each sample, count the mutations occurring in each exon and select the exons with at least one mutation. Return such samples together with the number of such exons and the maximum number of mutations in a single exon."

```
MUT = SELECT(manually_curated__cases__disease_type == " Kidney Renal Clear Cell
        Carcinoma") GRCh38_TCGA_somatic_mutation_masked;
EXON = SELECT(annotation_type == "exon" AND release_version == "22")
        GRCh38_ANNOTATION_GENCODE;

EXON1 = MAP() EXON MUT;
EXON2 = SELECT(region: count_EXON_MUT >= 1) EXON1;
EXON_RES = EXTEND(exon_count AS COUNT(),
        max_mut AS MAX(count_EXON_MUT)) EXON2;
MATERIALIZE EXON_RES INTO EXON_RES;
```

This example shows that GMQL is very effective at counting (in batch and on multiple samples) genomic elements, in this case mutations, that are mapped upon known genomic regions (in this case all exons), extracting regions having more mutations than a given threshold and then counting the number of such regions in each sample and the maximum number of mutations in such regions of each sample. Known human protein-coding and non-protein-coding exon regions of the GENCODE annotation release 22, originally provided by (https://www.gencodegenes.org/releases/22.html), are selected from the *GRCh38_ANNOTATION_GENCODE* dataset. Count of data sample items is performed by the MAP and EXTEND operations. MAP counts mutations in each sample within each exon while mapping the mutations to the exon regions; SELECT removes those exons in each sample that do not contain mutations; EXTEND counts how many exons remain in each sample and evaluates the maximum number of mutations in an exon, storing the result in the sample metadata as a new attribute–value pair.

Note that, also in this example, the query is applied in batch on multiple data samples, i.e., all those samples selected from the *GRCh38_TCGA_somatic_mutation_masked* collection, which can be very numerous (in the case of the publicly available TCGA data, at the time of writing all the available samples were 10,188 for a total of 10,903,607 somatic mutations). At the time of writing, by applying this GMQL query to the 336 somatic mutation samples publicly available in TCGA from kidney renal cell carcinoma patients, containing a total of 79,321 somatic mutations, and considering all 1,172,082 exon regions of 60,483 human protein-coding and non-protein-coding genes provided by GENCODE annotation release 22, it extracted 336 patient samples with somatic mutations involving 54,399 distinct exons. The execution of the query required only 2 hours and 19 minutes, a quite limited time considering the amount of data it applies on.

3. Find top ChIP-seq samples with highest number of enriched regions in promoters

“In each *ENCODE narrow peak ChIP-seq sample of high quality of the K562 chronic myelogenous leukemia cell line, select ChIP-seq enriched regions that intersect at least a gene promoter and extract the top 3 samples with the highest number of such enriched regions.*”

```
HM_TF = SELECT(assay == "ChIP-seq" AND output_type == "conservative idr thresholded peaks"
        AND biosample_term_name == "K562") HG19_ENCODE_NARROW_NOV_2017;
TRANSC = SELECT(annotation_type == "transcript" AND release_version == "19")
        HG19_ANNOTATION_GENCODE;
PROM = PROJECT(region_update: start AS start - 2000, stop AS start + 1000) TRANSC;

HM_TF_PROM = JOIN(DISTANCE < 0; output: LEFT_DISTINCT) HM_TF PROM;
HM_TF_PROM1 = EXTEND(region_count AS COUNT()) HM_TF_PROM;
RES = ORDER(region_count DESC; meta_top: 3) HM_TF_PROM1;
MATERIALIZE RES INTO RES;
```

This example uses a JOIN operation to extract the enriched regions in each ENCODE ChIP-seq narrow peak sample that intersect with at least a gene promoter (i.e., proximal regulatory region). Then, using an EXTEND operation it counts the number of such regions in each sample. Finally, using an ORDER operation it extracts the top 3 samples with the highest number of such regions.

Out of all 10,342 ENCODE ChIP-seq narrow peak samples available on November 2017 (containing a total of 1,604,183,681 enriched regions), initially the GMQL query selected 328 high quality (*conservative idr thresholded peaks*) ChIP-seq samples of the K562 chronic myelogenous leukemia cell line, including a total of 4,423,009 peaks. It also selected 196,520 gene transcription regions of the GENCODE annotation release 19 from the *HG19_ANNOTATION_GENCODE* dataset, originally provided by (https://www.gencodegenes.org/releases/19.html), around whose first bases it defines (using the typical -2k/+1k bp interval) the gene promoter regions to be considered. Next, a total of 1,925,928 ChIP-seq sample peaks intersecting with promoters were extracted and counted, and the 3 samples with more of such peaks were selected, having a total of 84,011 peaks. Processing required 16 minutes and 8 seconds.

The RES result dataset includes both regions and metadata; the former ones indicate interesting ChIP-seq enriched regions (that can be further inspected using, e.g., genome browsers), the latter ones allow tracing provenance of resulting samples and ease the biomedical interpretation of the results, facilitating also result sample stratification and further evaluations. Table 1 shows 4 metadata attributes of the 3 resulting samples: the *order* of the sample, the *experiment target*, the *biosample term name* (i.e., cell type) considered in the ChIP-seq experiment, and the *count* of enriched regions in the sample.

Table 1. Metadata excerpt of the resulting samples.

ID	Attribute	Value
1	_order	1
1	biosample_term_name	K562
1	experiment_target	RBFOX2-human
1	region_count	36753
2	_order	2
2	biosample_term_name	K562
2	experiment_target	eGFP-VEZF1-human
2	region_count	24219
3	_order	3
3	biosample_term_name	K562
3	experiment_target	L3MBTL2-human
3	region_count	23039

4. Find promoters with highest number of ChIP-seq enriched regions

"*After combining high quality narrow peak ChIP-seq sample replicates for each experiment target of the K562 chronic myelogenous leukemia cell line available in ENCODE and calculating the average enrichment (signal) for each obtained enriched region of each experiment target, extract the gene promoters with more than 50 experiment target enriched regions.*"

```
HM_TF = SELECT(assay == "ChIP-seq" AND output_type == "conservative idr thresholded peaks"
        AND biosample_term_name == "K562") HG19_ENCODE_NARROW_NOV_2017;
TRANSC = SELECT(annotation_type == "transcript" AND release_version == "19")
        HG19_ANNOTATION_GENCODE;
PROM = PROJECT(region_update: start AS start - 2000, stop AS start + 1000) TRANSC;

HM_TF1 = COVER(1, ANY; groupby: experiment_target;
        aggregate: AVG_signal AS AVG(signal)) HM_TF;
HM_TF2 = MERGE() HM_TF1;
PROM_HM_TF = MAP(count_name: region_count) PROM HM_TF2;
```

```
RES = SELECT(region: region_count > 50) PROM_HM_TF;
MATERIALIZE RES INTO RES;
```

This example uses a COVER operation to combine multiple ENCODE ChIP-seq sample replicates, a MERGE operation to collapse all enriched binding regions of such combined replicate samples in a single sample, and a MAP operation to map such regions on each gene promoter region and count how many of them intersect each promoter (i.e., proximal regulatory region). Finally, it selects the promoters that overlaps more than 50 enriched binding regions.

Out of all 10,342 ENCODE ChIP-seq narrow peak samples available on November 2017 (containing a total of 1,604,183,681 enriched regions), initially the GMQL query selected 328 high quality (*conservative idr thresholded peaks*) ChIP-seq samples of the K562 chronic myelogenous leukemia cell line, including a total of 4,423,009 peaks. It also selected 196,520 gene transcription regions of the GENCODE annotation release 19 from the *HG19_ANNOTATION_GENCODE* dataset, originally provided by (https://www.gencodegenes.org/releases/19.html), around whose first bases it defines (using the typical -2k/+1k bp interval) the gene promoter regions to be considered. Next, ChIP-seq replicate samples of the same experiment target are combined in a single sample, where every set of originally overlapping or contiguous peaks is combined in a single region representing all genomic bases covered by the original peaks in the set, and the average signal of all combined peaks in the set is calculated and associated with the new single region. In this way, a total of 259 samples and 3,359,704 regions was obtained. Then, all such regions in all combined ChIP-seq samples are merged in a single sample (where they remain distinct from each other) and are mapped to all considered promoters while counting how many of such regions overlap each promoter. Finally, only 61,245 promoters overlapping more than 50 of such regions are selected and stored. At the time of writing, processing required 4 minutes and 3 seconds.

5. Combining ChIP-seq and DNase-seq data in different formats and sources

"From ENCODE ChIP-seq experiment samples extract narrow peaks of enriched binding sites that intersect DNase-seq hotspot broad open chromatin regions with a false discovery rate (fdr) threshold of at least 0.01 from Roadmap Epigenomics in normal H1 embryonic stem cells."

```
CHIPSEQ = SELECT(assay == "ChIP-seq" AND output_type == "peaks" AND
        biosample_term_name == "H1-hESC") HG19_ENCODE_NARROW_NOV_2017;
DNASESEQ = SELECT(manually_curated__data_type == "DNase-seq" AND
        manually_curated__peak_caller == "HOTSPOT" AND manually_curated__region_type ==
        "broad" AND manually_curated__fdr_threshold == "0.01" AND
        epi__standardized_epigenome_name == "H1 Cells")
        HG19_ROADMAP_EPIGENOMICS_BED;
CHIPSEQ_IN_DNASESEQ = JOIN(DISTANCE < 0; output: RIGHT_DISTINCT) DNASESEQ
      CHIPSEQ;
MATERIALIZE CHIPSEQ_IN_DNASESEQ INTO CHIPSEQ_IN_DNASESEQ;
```

Combining data in different formats and from different sources, this example shows how to improve the quality of ChiP-seq called peaks by filtering out those peaks that are not in open chromatin regions, where they should be to reflect biological constraints. For the embryonic stem cell H1-hESC cell line, all ChIP-seq narrow peak samples available from the ENCODE data collection on November 2017 and DNase-seq open chromatin region samples from the Roadmap Epigenomics Project are selected. Since the HG19_ROADMAP_EPIGENOMICS_BED contains a single consensus (consolidated) sample for each epigenome, there is no need to use a COVER operation to combine multiple DNase-seq replicate samples into a single sample, which includes all identified open chromatin regions. A JOIN operation with the ChIP-seq peaks produces only the peaks that at least partially overlap any of these open chromatin regions, which are finally materialized. The join is performed for each of the selected ChIP-seq samples individually, so that each resulting sample is an originally selected ENCODE ChIP-seq sample, but including only the peaks that intersect an open chromatin region.

At the time of writing, this query was executed on 10,342 samples from the *HG19_ENCODE_NARROW_NOV_2017* dataset containing a total of 1,604,183,681 narrow peaks and 156 samples from the *HG19_ROADMAP_EPIGENOMICS_BED* dataset containing 24,574,576 regions in total; it initially selected 115 ChiP-seq narrow peak samples, regarding 40 antibody targets, and 1 DNase-seq sample, including a total of 31,220,157 peaks and 155,235 regions, respectively. As result, the query produces 115 samples with a total of 6,836,567 ChiP-seq peaks, regarding 40 different ChiP-seq experiment targets. Processing lasted 3 minutes and 28 seconds.

6. Counting distinct DNA mutations in patient groups

“Considering all TCGA public data on somatic mutations, group patients by tumor type and ethnicity, and count the distinct DNA somatic mutations in each group.”

```
MUTATION = SELECT() GRCh38_TCGA_somatic_mutation_masked;
MUTATION_BY_RACE = COVER(1, ANY;
        groupby: manually_curated__cases__disease_type, clinical__clin_shared__race;
        aggregate: overlap_count AS COUNT(), barcodes AS BAG(tumor_sample_barcode))
        MUTATION;
MUTATION_COUNT = EXTEND(mutation_count AS COUNT()) MUTATION_BY_RACE;
MATERIALIZE MUTATION_COUNT INTO MUTATION_COUNT;
```

This example of GMQL query takes into account all public DNA-seq data of TCGA patients, groups samples by their tumor type and patient ethnicity, and for each ethnic group of every tumor type it extracts and counts its distinct DNA somatic mutations, counting for each of them the overlaps among the different samples (each sample is identified by its TCGA barcode). It is worth noting that, the COVER operator permits to extract the genomic regions with certain features (e.g., DNA mutations) in the considered samples, and for each extracted region the BAG operator collects the barcodes of the samples with genomic features in that region.

Conversely, the EXTEND operator counts (through its COUNT() aggregate function) the number of distinct mutations in each resulting sample (one for each tumor type and patient race) and stores it in the sample metadata; finally, the MATERIALIZE operator returns the obtained result. In particular, the COVER operator extracts a sample for each tumor type and kind of patient race; the regions in the result samples are non-overlapping and are formed as contiguous intersections of at least one and at most any number of regions (i.e., somatic mutations) in the grouped input samples. For each result region, the COUNT aggregate function in the COVER operator computes the number of feature regions (i.e., mutations) that contribute to create the result region, and the BAG aggregate function gathers the TCGA barcode (identifier) of the sample of each contributing region to keep track of them. The metadata of each final resulting sample are the union of the metadata of the samples in the input data set that regard the same tumor type and patient race, and are enhanced with the number of distinct mutations computed for the tumor type and patient race the sample is referring to.

At the time of writing, the query applied on the entire *GRCh38_TCGA_somatic_mutation_masked* TCGA dataset, containing a total of 10,188 DNA-seq data samples and 10,903,607 mutations for the 33 different tumors listed in TCGA, extracted a total of 3,327,223 mutations within 141 samples. For example, at the time of writing the number of TCGA DNA-seq data samples regarding the Kidney Renal Clear Cell Carcinoma (KIRC) was 336, and the result data set included a sample for each of the ethnic group represented in the KIRC TCGA data, i.e., Asian, black or African American, and white; the total numbers of overall DNA somatic mutations in the input samples for the three ethnic groups were 874, 14,353, and 62,601, respectively, and the overall numbers of samples for the three groups were 6, 52, and 272, respectively, whereas the corresponding numbers of distinct somatic mutations in the result samples were 320, 4,846, and 22,241, respectively. The entire processing, for all the 33 tumor types required only 3 minutes and 21 seconds.

7. Combining different data types: DNA copy number variation and microRNA data

“Match DNA copy number variation (CNV) and microRNA (miRNA) data samples regarding the same biospecimen and extract the CNVs occurring in expressed miRNA genes in the paired samples of TCGA Thyroid Carcinoma (THCA) patients.”

```
CNV = SELECT(manually_curated__data_type == "Copy Number Segment" AND
        manually_curated__cases__disease_type == "Thyroid Carcinoma")
        GRCh38_TCGA_copy_number;
MIRNA_GENE = SELECT(manually_curated__data_type == "miRNA Expression Quantification"
        AND manually_curated__cases__disease_type == "Thyroid Carcinoma"; region:
        reads_per_million_mirna_mapped > 1) GRCh38_TCGA_miRNA_expression;
CNV_GENE_0 = MAP(mirna_genes AS BAG(mirna_id), mirna_gene_symbols AS
        BAG(gene_symbol), mirna_gene_IDs AS BAG(entrez_gene_id); count_name: gene_count;
        joinby: biospecimen__bio__bcr_sample_barcode) CNV MIRNA_GENE;
CNV_GENE = SELECT(region: gene_count > 0) CNV_GENE_0;
MATERIALIZE CNV_GENE INTO CNV_GENE;
```

This example GMQL query searches and combines pairs of TCGA samples of Copy Number Variation (CNV) and miRNA-seq data types that regard the same biospecimen, and returns the DNA copy number variations in each CNV sample which are within microRNA (miRNA) genes that are expressed (reads_per_million_mirna_mapped > 1) in the paired miRNA-seq sample.

In particular, the MAP operator on CNV and miRNA-seq datasets first joins samples based on the equivalence of their metadata *biospecimen_bio__bcr_sample_barcode* attribute (the identifier for TCGA biospecimens); then, in each pair of samples the COUNT aggregate function calculates the number of miRNA genes overlapping each DNA copy number variation, and a few BAG aggregate functions are used to collect the miRBase (http://www.mirbase.org/) IDs, the symbols, and the Entrez Gene IDs of such genes. Finally, the SELECT operator selects only those copy number variations of the paired samples that overlap at least one expressed miRNA gene, and the MATERIALIZE operator returns the result.

The resulting dataset contains only those CNV samples, with their metadata, that have a matching miRNA-seq sample, and containing only their DNA copy number variations (at least one) that occur within an expressed miRNA gene in the matched miRNA-seq sample. At the time of writing the TCGA CNV and miRNA-seq data samples were a total of 22,374 and 10,947 samples, respectively. Out of them, those of Thyroid Carcinoma (THCA) patients were 492 and 573, respectively. The pairs of samples found regarding the same biospecimen were 567; all of them contained DNA copy number variations within expressed miRNA genes of the same sample, with an average number of 88 copy number variations per sample. At the time of writing, the entire processing time required 1 minute and 43 seconds, a limited time considering the amount of processed data.

8. Combining and comprehensive processing of patients' heterogeneous omics data

"In TCGA data of BRCA patients, find the DNA somatic mutations within the first 2000 bp outside of the genes that are both expressed with FPKM > 3 and have at least a methylation in the same patient biospecimen, and extract these mutations of the top 5% patients with the highest number of such mutations."

```
EXPRESSED_GENE = SELECT(manually_curated__cases__disease_type == "Breast Invasive
        Carcinoma"; region: fpkm > 3.0) GRCh38_TCGA_gene_expression;
METHYLATION = SELECT(manually_curated__cases__disease_type == "Breast Invasive
        Carcinoma") GRCh38_TCGA_methylation;
MUTATION = SELECT(manually_curated__cases__disease_type == "Breast Invasive
        Carcinoma") GRCh38_TCGA_somatic_mutation_masked;

GENE_METHYL = JOIN(DISTANCE < 0; output: LEFT_DISTINCT; joinby:
        biospecimen__bio__bcr_sample_barcode) EXPRESSED_GENE METHYLATION;
MUTATION_GENE = JOIN(DISTANCE <= 2000, DISTANCE >= 0; output: LEFT_DISTINCT;
        joinby: biospecimen__bio__bcr_sample_barcode) MUTATION GENE_METHYL;

MUTATION_GENE_count = EXTEND(mutation_count AS COUNT()) MUTATION_GENE;
MUTATION_GENE_top = ORDER(mutation_count DESC; meta_topp: 5)
        MUTATION_GENE_count;
MATERIALIZE MUTATION_GENE_top INTO MUTATION_GENE_top;
```

The query is divided into three sections. Using the SELECT operator, the first one extracts relevant samples from three TCGA datasets (gene expressions, DNA methylations, somatic mutations); the second one combines the extracted samples and metrically evaluates the localization of their genomic regions, by means of two JOIN operations, to produce the relevant mutations searched; the third one counts them and selects those of the most mutated patients.

Specifically, the first JOIN operator applies on expressed gene and DNA-methylation datasets. It first combines samples based on the equivalence of their metadata *biospecimen__bio__bcr_sample_barcode* attribute (the TCGA biospecimen identifier); then, from every pair of samples of each biospecimen, it extracts the expressed gene regions that overlap at least a methylation site in the paired DNA methylation sample. Similarly, the second JOIN operator applies on the extracted expressed and methylated genes in each sample and on the entire BRCA mutation dataset of TCGA; in each sample of the latter one, it finds the DNA somatic mutations occurring within the first 2,000 bp upstream or downstream of any of the expressed methylated genes extracted in the paired sample of the same biospecimen. Then, the EXTEND operator uses the COUNT() aggregate function to determine the number of these mutations in each sample, the ORDER operator ranks the samples according to such number and extracts the top 5% samples with the highest number of these somatic mutations, and finally the MATERIALIZE operator returns the result. Note that this complex query is simply expressed through a few GMQL statements, also thanks to the GMQL implicit iteration over all the samples even matched through their metadata.

At the time of writing, the query was executed over all 11,091 gene expression, 12,218 DNA methylation and 10,188 somatic mutation samples publicly available in TCGA, for a total of 56.5 GB, 1.3 TB and 2.3 GB of data respectively. The query initially selects 1,222 samples of expressed gene data, 1,234 samples of DNA methylation data, and 985 samples of DNA somatic mutation data of TCGA BRCA patients, containing a total of 11,847,376 expressed gene regions, 358,803,211 methylation sites, and 363,521 DNA mutations, respectively.

The combination of each biospecimen’s gene expression and DNA methylation data identified 1,208 breast cancer patient samples presenting methylated expressed genes, with an average of 8,573.45 of such genes for each identified biospecimen. Thanks to the TCGA patients’ clinical data reported in the available sample metadata, which GDM seamlessly manages and GMQL carries on during the processing, these patients can be clinically characterized. In particular, they have an *average age at diagnosis* of 58.28 years; 552 of them received *radiation therapy*, whereas 453 did not, and for 203 of them it is unknown; 872 patients are *estrogen receptor positives* and 252 *negatives*; 713 are *progesterone receptor positives* and 349 *negatives*. Then, the query extracts 636 biospecimens having somatic mutations occurring within the first 2000 bp outside of the same biospecimen’s expressed and methylated genes. Finally, these mutations in each biospecimen are counted (their average number per biospecimen is 3.10), and the mutations of the top 5% patient biospecimens with the highest number of such somatic mutations are selected (their average number per biospecimen is 22.06).

The Table 2 below reports an excerpt of the metadata attributes and of their values associated with the selected patients. Notably, the top patient biospecimen has 128 mutations, about three times of the ones of the second top patient, who was first diagnosed with BRCA when was about 20 years younger; all patients but 8 are positives to progesterone and/or estrogen receptor, and 9 of them received radiation therapy whereas 11 did not. The whole execution of this example query on the public GMQL system installation, where the entire public TCGA datasets are available, lasted only 57 minutes; execution time is low when compared to the big amount of samples and genomic regions processed, and to the complexity of the processing.

Table 2. Metadata excerpt of the top 5% patients finally selected.

Order	Mutation count	Age at initial pathologic diagnosis	Radiation therapy	Estrogen receptor	Progesterone receptor
1	128	90	NO	Positive	Negative
2	46	68	NO	Positive	Positive
3	43	63	NO	Positive	Positive
4	28	61	YES	Negative	Negative
5	28	81	YES	Negative	Negative
6	27	83		Positive	Negative
7	27	60	YES	Negative	Negative
8	25	47	YES	Positive	Positive
9	25	55	YES	Negative	Negative
10	23	76	NO	Positive	Negative
11	22	69	NO	Negative	Negative
12	22	50	YES	Positive	Positive
13	21	77	NO	Positive	Positive
14	20	74	NO	Positive	Positive
15	20	77	YES	Positive	
16	16	90	NO	Negative	Negative
17	14	59	NO	Positive	Positive
18	14	41	YES	Positive	Positive
19	14	68	YES	Positive	Positive
20	13	75	NO	Positive	Positive
21	13	64	YES	Positive	Positive
22	12	59	NO	Negative	Negative
23	11	69	YES	Positive	Positive
24	10	69	YES	Positive	Negative
25	10	40	YES	Negative	Negative
26	10	44	YES	Positive	Positive
27	9	63	NO	Positive	Positive
28	9	88	NO	Positive	Positive
29	8	45	NO	Positive	Positive
30	8	61		Positive	Negative
31	8	66		Positive	Positive

The same GMQL query can be directly applied on other types of patients or datasets, just by changing the SELECT operator parameters. Note that the result dataset includes both genomic somatic mutations and clinical metadata of the finally selected patients. The former ones indicate interesting somatic mutations that could be associated with breast cancer (which can be further inspected, e.g., using genome browsers); the latter ones allow tracking the provenance of resulting samples and ease the biomedical interpretation of the results, facilitating also result sample stratification and further evaluations. This association between processed genomic data and their biological/clinical metadata is not supported by other system currently available, and represents one of the relevant aspects of GDM and GMQL.

9. Calling cell line-specific active enhancers

"From the entire ENCODE ChIP-seq narrow peak dataset, combine all available H3K4me1 and H3k27ac histone modifications to identify, for each cell line in ENCODE, putative active enhancers, i.e., regions of the genome where both a peak of H3K4me1 and a peak of H3k27ac are present. Next, combine the results for the various cell lines to identify those enhancers that are specifically present in only one of them."

```
me1 = SELECT(assay == "ChIP-seq" AND output_type == "peaks" AND
        experiment_target == "H3K4me1-human") HG19_ENCODE_NARROW_NOV_2017;
me1_c = COVER(1,ANY; groupby: biosample_term_name) me1;
ac = SELECT(assay == "ChIP-seq" AND output_type == "peaks" AND
        experiment_target == "H3K27ac-human") HG19_ENCODE_NARROW_NOV_2017;
ac_c = COVER(1,ANY; groupby: biosample_term_name) ac;

active = JOIN(DISTANCE < 0; output: INT; joinby: biosample_term_name) me1_c ac_c;
labeled = PROJECT(region_update: cell AS META(ac_c.biosample_term_name, STRING)) active;

cell_specific = COVER(1,1; aggregate: cell_line AS BAG(cell)) labeled;
MATERIALIZE cell_specific INTO cell_specific;
```

This example shows the power of GMQL in performing implicit iterations on entire datasets, even matching the dataset samples through their metadata; this enables GMQL to take full advantage of parallel processing and to apply efficiently on big data in order to provide genome-wide answers to fundamental (epi)genomics questions. First, from the entire ENCODE CHiP-seq narrow peak dataset, the GMQL query selects all ChIP-seq experiment samples targeting the H3K4me1 histone modification and, by means of a COVER operation with the "*biosample_term_name*" sample metadata attribute as *groupby* key, it combines their replicas in case available for each cell line. The same operation is then performed for the ChIP-seq experiment samples targeting the H3K27ac histone modification.

Then, the query identifies putative active enhancers as the genomic regions where a H3K4me1 peak overlaps with a H3K27ac peak in the same cell line; in order to do so, first a JOIN operation with the "*biosample_term_name*" sample metadata attribute as *joinby* key is used. The result is a set of samples, one for each cell line, where each sample contains the list of putative active enhancer regions for a cell line. A PROJECT operation copies the value of the sample metadata attribute "*biosample_term_name*", describing the cell line, within a new attribute "*cell*" of every region, so that each region in the output is labeled with the name of the cell line of origin. Finally, a COVER(1,1) operation, with parameters min and max accumulation equal to 1, selects only those putative active enhancer regions of each cell line that do not overlap any other enhancer of any cell line; it outputs a single sample with all the cell line specific putative active enhancers extracted, each one labeled with its cell line name in the new attribute “*cell_line*”.

Out of all 10,342 ENCODE ChIP-seq narrow peak samples available on November 2017 (containing a total of 1,604,183,681 enriched regions), initially the GMQL query selects 31,146,835 H3K4me1 regions from 182 replicate samples and 23,457,101 H3K27ac regions from 231 replicate samples, from 64 and 77 different human cell lines; after combining sample replicates, 13,621,664 H3K4me1 regions from 64 samples and 10,379,105 H3K27ac regions from 77 samples, one sample for each cell line, remain. The putative active enhancer regions extracted (where both H3K4me1 and H3K27ac peaks overlap) are 5,786,721 within 56 samples, one for each cell line with both H3K4me1 and H3K27ac data. Finally, the extracted cell line specific putative active enhancers are 1,406,494, regarding 56 different cell lines, with an average of about 25,116 ones per cell line.

At the time of writing, query execution required only 9 minutes and 23 seconds. An excerpt of the final output of the query follows in Table 3.

Table 3: Excerpt of the cell line specific putative active enhancers extracted.

chr	left	right	strand	cell_line
chr1	162007355	162007518	*	SK-N-MC
chr2	68330641	68330886	*	osteoblast
chr2	189184720	189184799	*	SK-N-SH
chr5	79140318	79140346	*	osteoblast
chr6	3006575	3006901	*	neutrophil
chr6	133014305	133014320	*	thoracic aorta
chr8	19211561	19211765	*	body of pancreas
chr8	66267133	66268538	*	A549
chr8	77650141	77650203	*	SK-N-MC
chr9	14888447	14889044	*	SK-N-MC
chr10	44528641	44528739	*	body of pancreas
chr10	118500245	118500438	*	fibroblast of lung
chr11	59325121	59325258	*	tibial nerve
chr12	96421146	96421605	*	neutrophil
chr18	26308257	26308406	*	osteoblast
chrX	151267578	151268406	*	bipolar neuron