EpiRegioDB Supplemental Material

Release 1.0

Nina Baumgarten, Dennis Hecker, Sivarajan Karunanithi, and Mar

Aug 31, 2020

Contents

CHAPTER 1

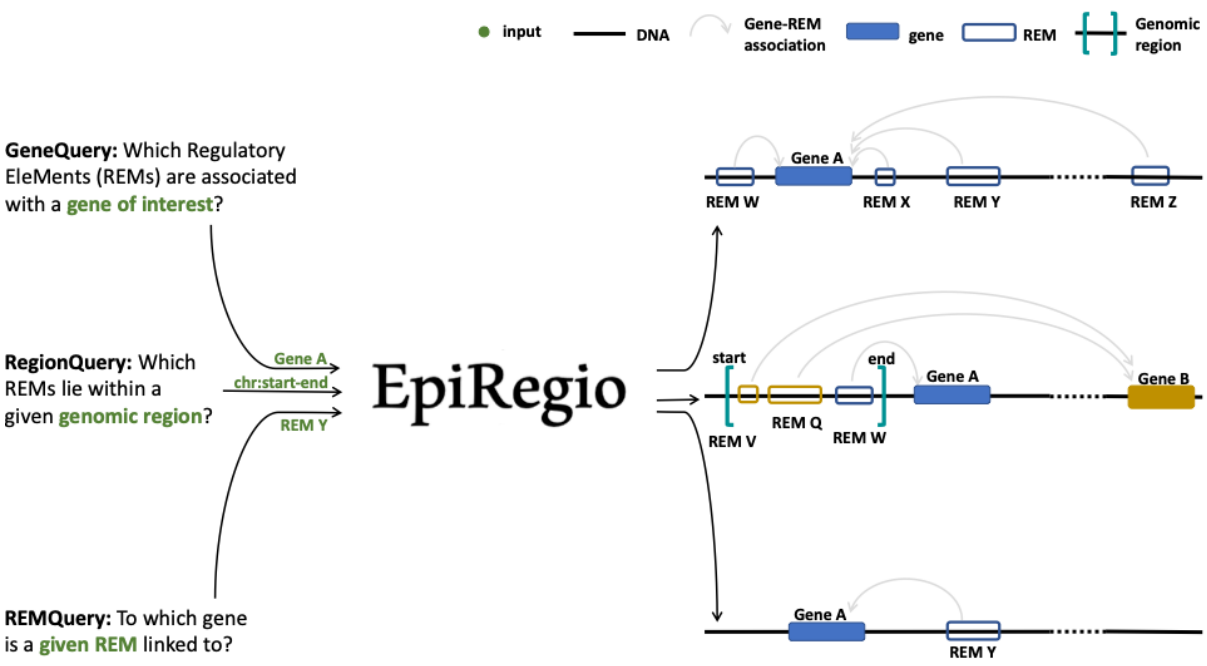
About EpiRegio Web Server

The field of research on gene regulation has considerably grown during the last years and the acknowledgement of its importance in orchestrating the genetic landscape has expanded. One of the key players are non-coding DNA regions, which regulate gene expression. They are able to enhance or repress the expression of their associated genes. These Regulatory EleMents (REMs) can be located far away from their associated genes. Identifying REMs is difficult, as there is no method yet to determine them with abolute certainty. Different computational approaches are being used, combining various kinds of genomics data to annotate REMs. An even more challenging task is to link the putative REMs to their associated gene.

Here we present the EpiRegio web server, a resource of REMs, providing information about their associated gene, their relevance for their gene's expression and their activity in different cell types and tissues. With EpiRegio users are enabled to look into regions of interest, analyze the genomic locations that impact the expression of specific genes and access details about the regulatory elements.

CHAPTER 2

Overview of possible queries



CHAPTER 3

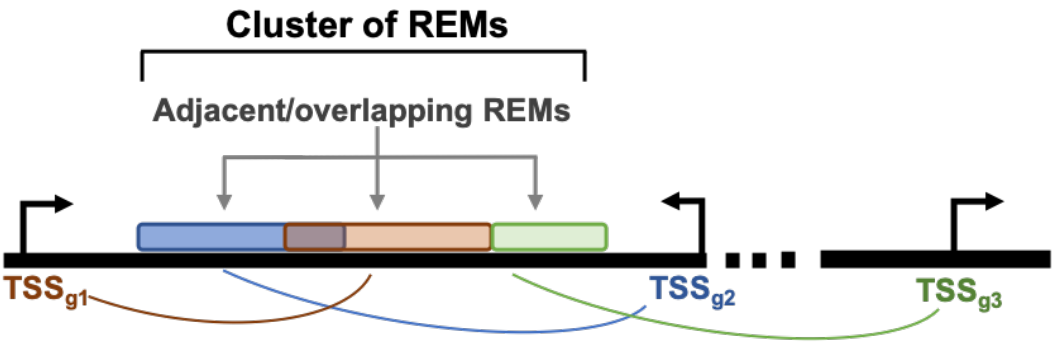
Learning of Regulatory EleMents (REMs)

EpiRegio is based on *STITCHIT*, a method which was previously developed in our group. It is a peak-calling free approach to identify gene-specific REMs by analyzing epigenetic signal of diverse human cell-types with regard to gene expression of a certain gene. In order to identify REMs, a large genomic area around a gene of interest is partitioned into distinct regions, which show variation in their epigenetic profile correlating with changes in gene expression. *STITCHIT* is applied to large collections of paired, uniformly processed DNase1-seq and RNA-seq samples from Roadmap, ENCODE and Blueprint. *STITCHIT* was shown to outperform peak based approaches e.g. GeneEnhancer and UnifiedPeaks regarding the accuracy and resolution. Furthermore, we validated results from *STITCHIT* with external data such as ChIA-PET and Promoter-Capture Hi-C data. To show the functional advantage of *STITCHIT*, various analyses were performed, like the rediscovery of known enhancers and the partitioning of larger regulatory elements into smaller regions. Additionally, CRISPR-Cas9 experiments were done to illustrate the reliability of *STITCHIT* [2].

For more information, a detailed explanation of the computational method and the evaluation of the results, please have a look at our bioRxiv preprint.

CHAPTER 4

Cluster of regulatory elements



The way STITCHIT identifies REMs results in REMs that are mapped to one gene. Genomic locations are not exclusive to REMs, hence REMs associated to different genes can overlap which each other. Consequently the overlapping region is linked to more than one gene. To account for these overlapping REMs, we introduce the term Cluster of Regulatory EleMents (CREM). One CREM consists of all REMs that overlap with each other or that are adjacent to each other without any break in between (see the schema above). A CREM ends when there is no neighbouring REM to either side of it. Each CREM is composed of a minimum of two REMs and is assigned to a unique ID. In other words, a CREM can be considered as one coherent regulatory region that is potentially associated to multiple genes, where it is known which part links to which gene.

CHAPTER 5

Data preprocessing for the EpiRegio webserver

The data hosted by the web server EpiRegio was generated with *STITCHIT*. *STITCHIT* was applied to human paired DNase1-seq and RNA-seq data, namely *110* samples from the Roadmap consortium and *56* samples from the Blueprint consortium. The considered samples comprise of *46* different tissues and cell types. While the Blueprint data set consists of various primary cell types and disease related samples associated to the haematopoietic system, Roadmap data provides a broader diversity of cell and tissue types. All data sets have been uniformly preprocessed. DNase1-seq was adjusted to sequencing depth and gene expression is quantified in transcripts per million. For every gene, *STITCHIT* inspects a user-defined region around the gene to determine putative associated REMs. For the data provided in EpiRegio, we consider a window of *100,000* bp upstream of a gene's transcription start site, the entire gene body and the window of *100,000* bp downstream of a gene's transcription termination site. Hence, even distant REMs are taken into account. In total *EpiRegio* contains *2,404,861* REMs associated to *35,379* protein-coding and non-protein coding genes. Together, they form *365,286* distinct CREMs. In the following table quantitative characteristics of REMs and CREMs are summarized.

	mean	min	max
REM length [bp]	228.9	4	1999
CREM length [bp]	533.6	11	8752
# REMs per CREM	3.5	2	122
# associations to different genes in a CREM	2.8	2	31

CHAPTER 6

Future releases

We will continuously update and expand EpiRegio. Besides of adding more functionalities and analyses, we will also update the underlying dataset if we can make improvements by including new datasets or by tweaking processes of STITCHIT. Right now, version 1 is available. Every file you export contains the current day and the version number. All dataset versions are available at our Zenodo repository, so that you can still reproduce all your analyses even after a version upgrade. We also upload the source code of every release on Zenodo.

CHAPTER 7

Cite Us

If you use this webserver, please cite the following:

1. Nina Baumgarten, Dennis Hecker, Sivarajan Karunanithi, Florian Schmidt, Markus List, Marcel H Schulz, EpiRegio: analysis and retrieval of regulatory elements linked to genes, Nucleic Acids Research, , gkaa382, https://doi.org/10.1093/nar/gkaa382
2. Schmidt et. al., Integrative analysis of epigenetics data identifies gene-specific regulatory elements

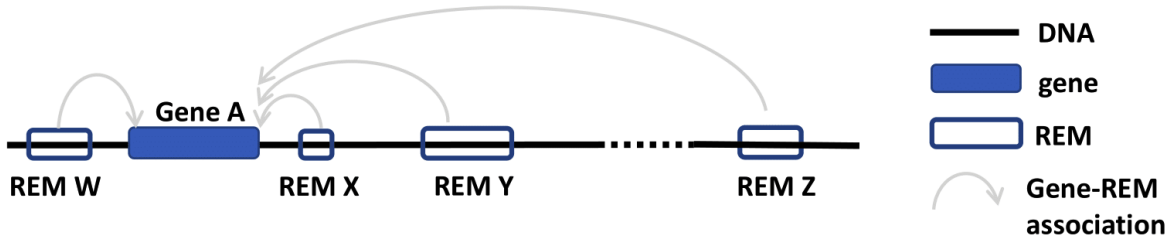
CHAPTER 8

Query Guide

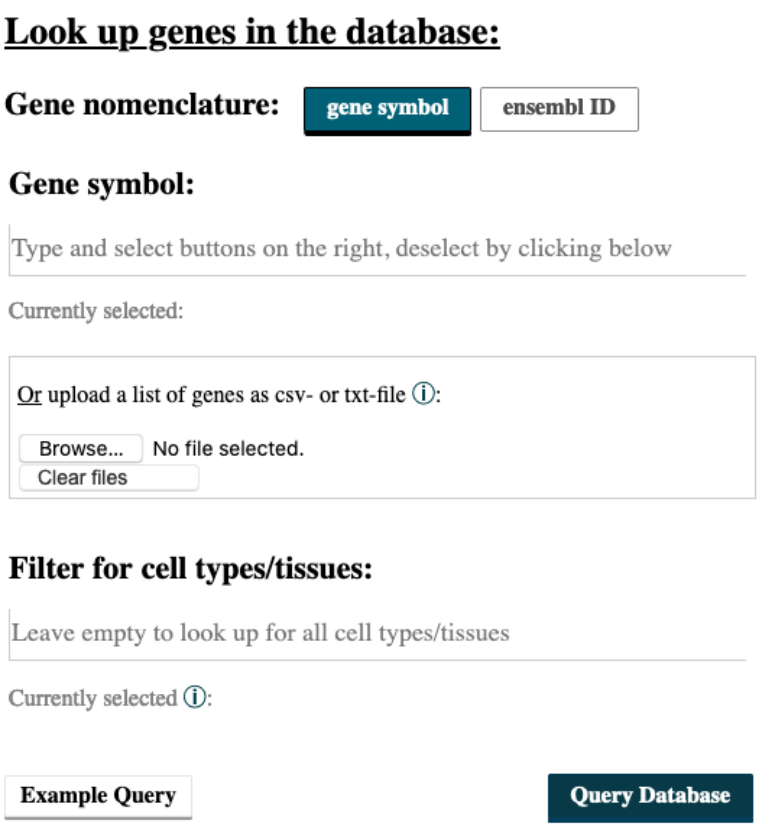
Here we provide a step-by-step guide for every query available, including an explanation of the output. Every query has an *Example Query* button at the bottom of the page. Try it out to see how a valid query would look like. Have a look at *Results in detail* to get an explanation of all the output parameters. In addition, we provide an example file for each possible file upload, see *Examples file upload*.

8.1 Gene Query

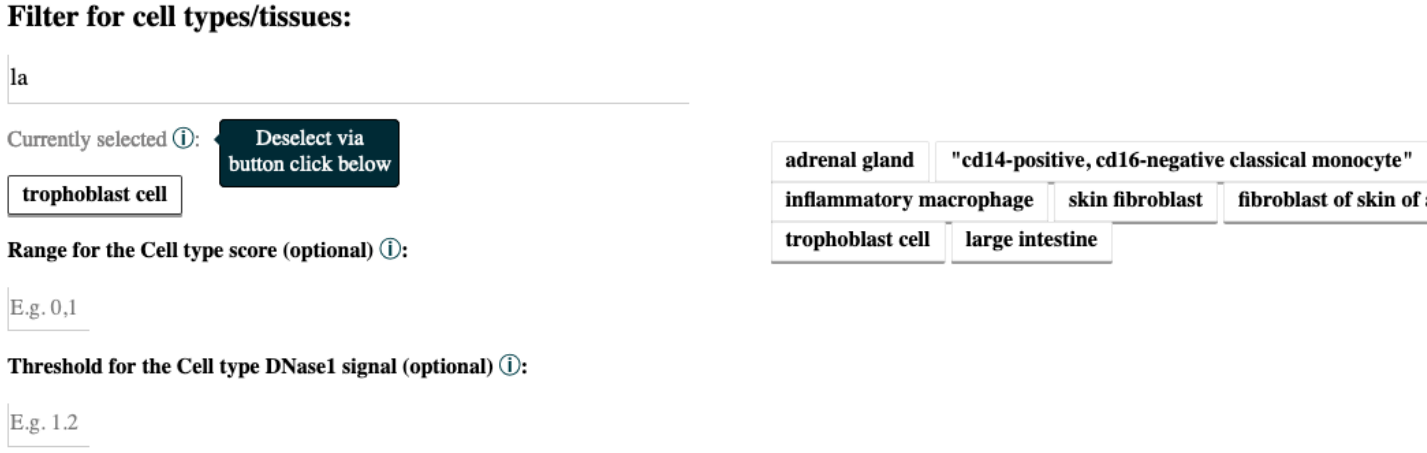
Do you wish to search for Regulatory Elements (REMs) related to a specific gene?



1. Go to the *Gene Query* tab.
2. You can choose to search either with *gene symbol* or *ensembl ID*. The version number of Ensembl IDs is not required. When entering gene symbols, you can add them from the suggestions appearing on the right by clicking on the buttons. Selected buttons will be listed underneath *Currently selected:*. Deselect your choices by reclicking on those buttons. We use the human genome assembly version GRCh38.p10.



3. When you have multiple IDs or symbols to search, separate them by comma in the input field or create a csv- or txt-file and upload it. All of the commonly used separators are being recognized. A combination of both, the input field and the uploaded file, is not implemented.
4. Choosing cell types/tissues: Start typing in the field *Filter for cell types/tissues:* the cell types of your interest, and suggesstions of available cell types matching your query will appear. To select a cell type, click on the button on the right. Cell types you write but do not select via a button click will not be considered for the query. To deselect click again on the button below *Currently selected:*. If your cell type does not appear, have a look at the *Available cell and tissue types* section and see whether you can find it there.



Once you selected a cell type, two new input fields will appear, which give the option to choose thresholds. The thresholds refer to the *Cell type score* and the *Cell type DNase1 signal* of the REMs in the cell types/tissues. Only

REMs that exceed the thresholds in **ALL** of the cell types you selected will be shown in the output table. The threshold for the *Cell type score* requires a range (e.g. 0-1). You can filter for the absolute *Cell type scores* if you write it like **|0.5,1|**. It is possible to set just one or both thresholds. Leave the fields empty to get back all REMs independent of their score and DNase1 signal.

5. The result page shows the information based on your query settings. All the REMs associated to your queried genes are listed with their location, their *Predicted function*, the *Model score*, the REM cluster they are belonging to and their activity in the cell types you selected. The *Model score* [0,1] indicates how important a REM is for its associated gene over all cell types. The closer the value is to 1, the more important the REM is. The next column *Cluster of REMs (CREM) ID* contains the ID of the cluster this REM is contained in. A cluster of REMs consists of all the REMs that are directly adjacent with no base pair in between or that overlap with each other. Click on a CREM ID to get to a table with all REMs of this CREM. We provide a more detailed description of CREMs here. If you selected cell types in your query, the *Cell type score* and the *Cell type DNase1 signal* of the REMs in these cell types will be shown as average over all the samples n in the database (for each cell type separately, not averaged over all cell types). The *Cell type score* [-1,1] is the normalized product of the regression coefficient and the standardized DNase1 activity, indicating the relative contribution of a REM to its target gene's expression in this cell type. The higher the value, the higher is the REM region expected to have an activating effect on its gene's expression in this cell type. This does not necessarily mean that the REM is an activator. The REM could also be a repressor but the chromatin is closed at its location. For more detailed informations about the *Cell type score* see here. *Cell type Dnase1 signal* is the DNase1 signal, indicating the chromatin accessibility in the REM region. If you need some more information on the genes themselves, click on the *Gene ID* to get to the respective Ensembl web page. By clicking on the *Gene symbol* you will receive a table with all REMs that are associated to the clicked gene. To see the REM region in the UCSC Genome Browser click on the chromosome entry. Another option is to use the 'Functional enrichment analysis' button to perform an analysis of all genes in the table with g:Profiler on default settings. The limit for genes that can be included in the g:Profiler link is 90. You will get a notification if this limit is exceeded. The link will still work, but contain only the first 90 unique genes. You can export the table as xls- or csv-file. The downloaded file's name is adapted to your query and contains the date as well as the current version of the website.

Home | Gene Query | Region Query | REM Query | REST API | Download | Documentat

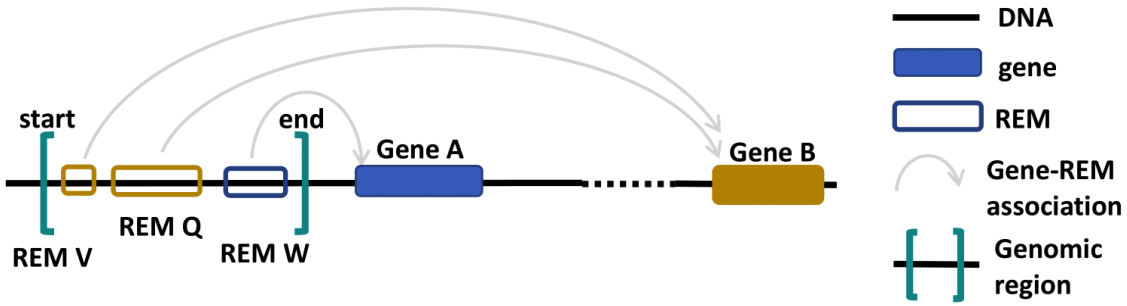
Functional enrichment analysis g:Profiler | Download Excel | Download CSV | Search:

Results based on your query for the genes: SSTR1, DBR1, RP11-77K12.9

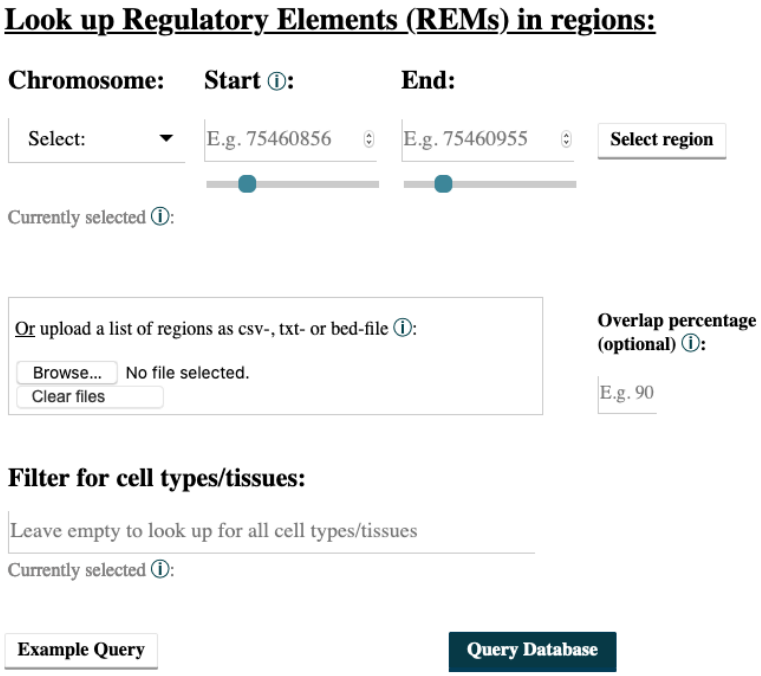
Gene ID	Gene symbol	REM ID	Chr	REM start	REM end	Predicted function	Model score	Cluster of REMs (CREM) ID	Number of REMs in the CREM	Heart score (n=2)	Heart DNa signal (n=2
ENSG00000138231	DBR1	REM1643186	chr3	138061012	138061951	repressing	0.5038420	CREM0258490	2	-0.0459012	1.1149245
ENSG00000138231	DBR1	REM1643189	chr3	138062152	138062301	activating	0.5298390	CREM0258491	2	0.0290645	2.353615
ENSG00000138231	DBR1	REM1643191	chr3	138062802	138062901	activating	0.5553150	No CREM	-	0.0930812	7.35882
ENSG00000138231	DBR1	REM1643194	chr3	138063702	138064401	repressing	0.1541190	CREM0258492	2	0.0019201	0.1939475

8.2 Region Query

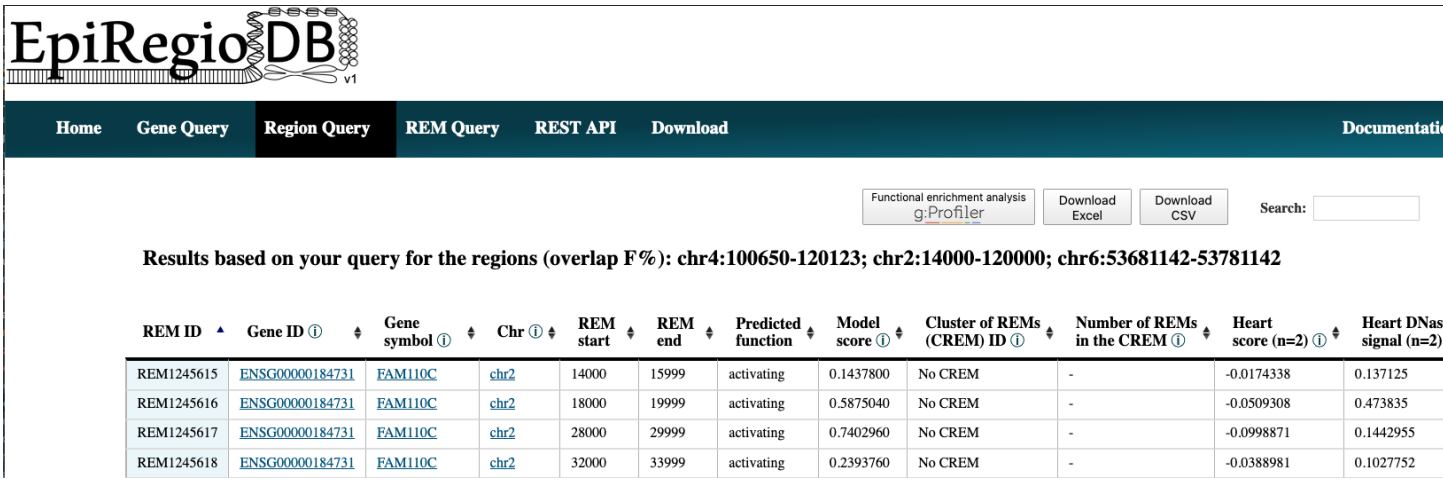
Do you wish to search for Regulatory Elements (REMs) being located in a specific genomic region?



1. Go to the *Region Query* tab.
2. You can enter a region by choosing a chromosome, the start and the end point and then clicking on the *Select* button. Add as many regions as you like. Deselect your choices by reclicking on the added buttons. Only REMs that are located in your chosen regions will be given as output. You can select the percentage of overlap and by this define how much of your selected region has to be covered by a REM for this REM to be shown in the output. For example, with an overlap of 50% only the REMs that cover at least half of a region's length will be returned. Per default only REMs that are located completely within your regions are reported.



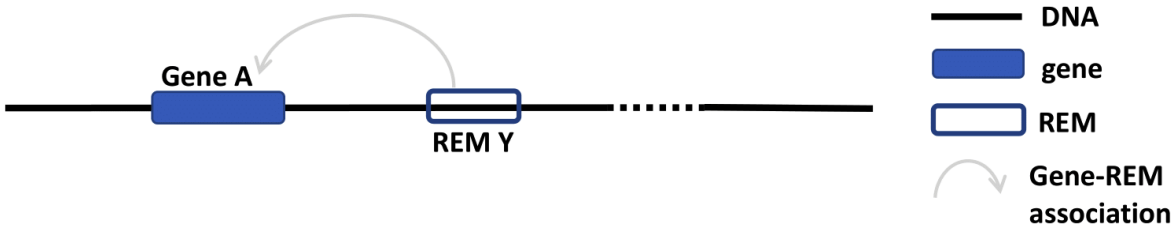
3. You can also upload a csv-, txt- or bed-file with your regions of interest, see *Examples file upload*.
4. Choosing cell types/tissues: The selection of cell types functions in the same way as described above in the *Gene Query* at point 4.
5. The output is very similar for all queries. Have a look at point 5 of the *Gene Query* or at the *Results in detail*. Below you can see how the output of the Region query looks like.



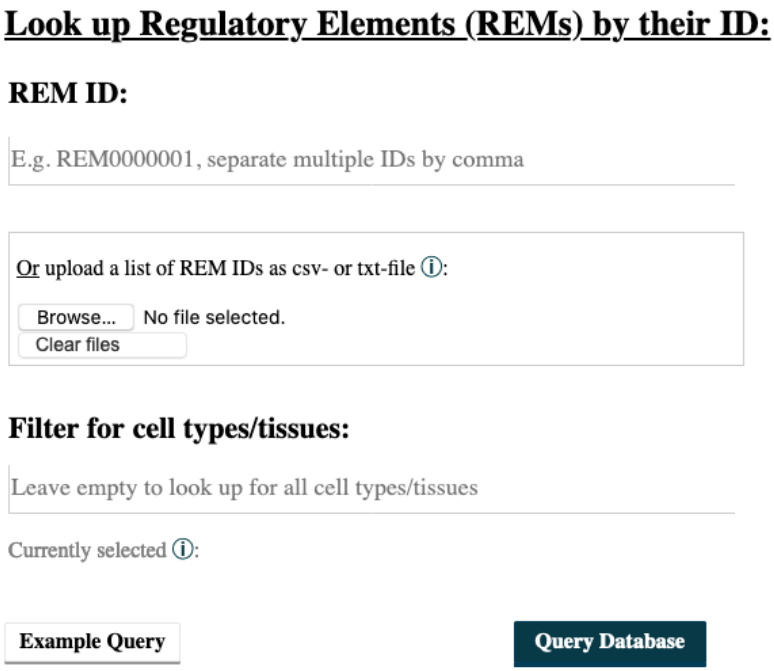
REM ID	Gene ID	Gene symbol	Chr	REM start	REM end	Predicted function	Model score	Cluster of REMs (CREM) ID	Number of REMs in the CREM	Heart score (n=2)	Heart DNas signal (n=2)
REM1245615	ENSG00000184731	FAM110C	chr2	14000	15999	activating	0.1437800	No CREM	-	-0.0174338	0.137125
REM1245616	ENSG00000184731	FAM110C	chr2	18000	19999	activating	0.5875040	No CREM	-	-0.0509308	0.473835
REM1245617	ENSG00000184731	FAM110C	chr2	28000	29999	activating	0.7402960	No CREM	-	-0.0998871	0.1442955
REM1245618	ENSG00000184731	FAM110C	chr2	32000	33999	activating	0.2393760	No CREM	-	-0.0388981	0.1027752

8.3 REM Query

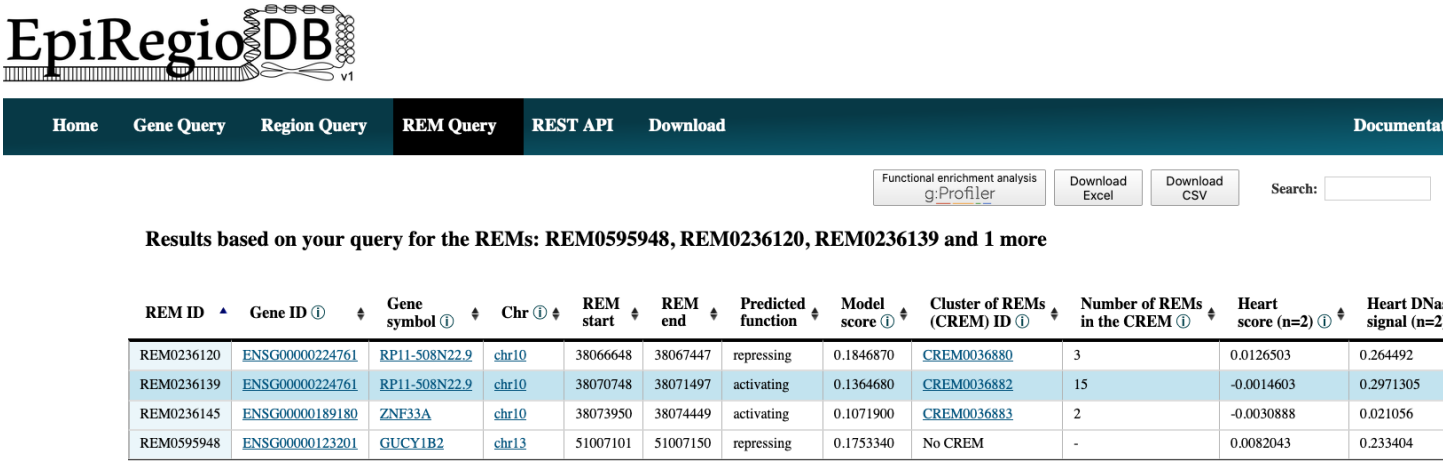
Do you wish to search for Regulatory Elements (REMs) by their ID?



1. Go to the *REM Query* tab.
2. Enter the IDs of your REMs of interest. Sepearte multiple ones by comma. You can upload a csv-file containing REM IDs. A combination of both, input field and uploaded file, is not implemented.



3. Choosing cell types/tissues: The selection of cell types functions in the same way as described above in the *Gene Query* at point 4.
4. The output is very similar for all queries. Have a look at point 5 of the *Gene Query* or at the *Results in detail*. Below you can see how the output of the REM query looks like.



REM ID	Gene ID ⓘ	Gene symbol ⓘ	Chr ⓘ	REM start	REM end	Predicted function	Model score ⓘ	Cluster of REMs (CREM) ID ⓘ	Number of REMs in the CREM ⓘ	Heart score (n=2) ⓘ	Heart DNa signal (n=2
REM0236120	ENSG00000224761	RP11-508N22.9	chr10	38066648	38067447	repressing	0.1846870	CREM0036880	3	0.0126503	0.264492
REM0236139	ENSG00000224761	RP11-508N22.9	chr10	38070748	38071497	activating	0.1364680	CREM0036882	15	-0.0014603	0.2971305
REM0236145	ENSG00000189180	ZNF33A	chr10	38073950	38074449	activating	0.1071900	CREM0036883	2	-0.0030888	0.021056
REM0595948	ENSG00000123201	GUCY1B2	chr13	51007101	51007150	repressing	0.1753340	No CREM	-	0.0082043	0.233404

8.4 Interactive tables

All result tables possess additional functionalities like the possibility to filter for certain values or to sort the table by a selected column. Moreover, there are several links included. Each *Gene ID* in the tables is a link that gets you to the entry of this gene from the Ensembl genome browser from the Ensembl release 91. The original annotation that the model was trained on is GENCODE V27. The entries in *Gene symbol* creates a new table with all the REMs that are associated to the clicked gene. Further, you can click on the chromosome value in a row to view the REM's region inside of the UCSC Genome Browser. The values in the column *Cluster of REMs (CREM) ID* redirect you to a new table with all the REM contained in this cluster. In addition, the button 'Functional enrichment analysis' runs

an analysis on all the genes currently in the table with g:Profiler on default settings. With more than 90 unique genes, the link exceeds the maximum characters possible for URLs. You will get a warning if that happens (see below). You can still use the link, but it will only contain the first 90 unique genes of your query. To do the functional enrichment analysis with all genes, you could download the excel-file, copy the *Gene ID* column and paste it in g:Profiler.

8.5 Examples file upload

It is also possible to upload a csv-, txt- or bed-file for the different queries. A combination of both, input field and uploaded file, is not implemented. All of the commonly used separators are being recognized, however files with empty fields will not be read correctly. For the region query it is important that the order of chromosome, start position and end position is correct. If a bed-file is provided for the this query, the columns have to be in the order chromsome, start position and end position as well. All additional columns beside of those first three ones will be ignored. For the gene query you still need to specifiy the *gene nomenclature*, either *gene name* or *ensembl id* according to the one used in your uploaded file.

In the following you can download an example for each query:

- `Gene query`
- `Region query`
- `REM query`

8.6 Available cell and tissue types

In case you are wondering, whether your cell type or tissue is availale on EpiRegio, we list the available ones here. Every name is written as you would find it in the field where you filter for cell types (without the bullet point of course).

The following cell/tissue types are available from Roadmap. Please note that we list the cell/tissue type (biosample) names as listed in the ENCODE website, which also hosts the Roadmap data. :

- skin fibroblast
- fibroblast of skin of abdomen
- imr-90
- trophoblast cell
- muscle of arm
- stomach
- muscle of back
- small intestine
- muscle of leg
- large intestine
- left lung
- kidney
- right lung
- thymus
- heart
- renal cortex

- adrenal gland
- renal pelvis
- left kidney
- left renal cortex
- left renal pelvis
- right renal pelvis
- spinal cord
- right renal cortex interstitium
- spleen
- psoas muscle
- muscle of trunk
- ovary
- pancreas
- testis
- forelimb muscle
- hindlimb muscle
- h1-hesc

From Blueprint we got the following cell types:

- "cd8-positive, alpha-beta t cell"
- "cd14-positive, cd16-negative classical monocyte"
- acute lymphocytic leukemia
- macrophage
- "cd34-negative, cd41-positive, cd42-positive megakaryocyte cell"
- "cd4-positive, alpha-beta t cell"
- erythroblast
- macrophage
- inflammatory macrophage
- acute myeloid leukemia
- chronic lymphocytic leukemia
- macrophage – b-glucan
- cd14-positive monocyte

8.7 Results in detail

The tables you get from the different queries contain the same columns. Here you can get some more detailed information on each of them.

8.7.1 Gene ID and symbol

For the gene nomenclature we use the hg38 human genome version from the Ensembl Genome Browser. For each gene ID we have one gene symbol available. If a queried gene symbol is called to be invalid, try to use the ENSG ID (e.g. ENSG00000000001), as they are more definite.

8.7.2 REM ID

REM ID is how we define the REMs internally. Each *REM ID* is unique. Also the REMs, which have the exact same genomic region but are associated to different genes (happens rarly), are assigned to different *REM IDs*. We start counting from REM0000001 ascending.

8.7.3 Predicted function

STITCHIT identifies REMs by interpreting differential gene expression, meaning that a REM can be associated with an increase in gene expression as well as with a decrease. This association is represented by the regression coefficient. In case of a positive regression coefficient we assume an activating effect of the REM on its target gene's expression and for a negative regression coefficient a repressing effect.

8.7.4 Model score

The *Model score* is the absolute binary logarithm of the p-value of the regression coefficient for the association between a REM and its target gene. To normalize the model score in the range [0, 1], we diveded the values by the maximal value. Consequently, at least one REM out of the REMs associated to a gene, has the highest value 1. The *Model score* serves as an indicator on how important a REM is for the expression prediction of its target gene in relation to the other REMs associated to that gene. The closer the score is to 1, the more impact the REM is supposed to have. This value is not cell type specific as it is based on the regression coefficient, which is calculated over all cell types. It allows for a comparison in between the REMs but not in between cell types. For a cell type-specific comparison, have a look at the *Cell type score*.

8.7.5 Cluster of REMs (CREM) ID

As STITCHIT determines REMs for each gene seperately, the identified REMs for different genes can overlap. A *REM cluster* is a region of neighbouring REMs that are directly adjacent or that overlap with each other. There has to be a minimum of two neighbouring REMs to be called a CREM. Each *REM cluster* is assigned to a unique *CREM ID*. We start counting from CREM0000001 ascending. By clicking on the *Cluster of REMs (CREM) ID* you get forwareded to a table with all REMs inside of this cluster. We show a schema of a CREM here.

8.7.6 Number of REMs per CREM

This shows how many REMs are contained in the CREM to which the REM belongs to. If the row is empty, then the REM does not have any adjacent or overlapping REMs and therefore is not considered as a cluster.

8.7.7 Cell type score

Cell type score is a normalized quantity in range [1,1], which represents the relative contribution of a REM (r) to its gene's expression in a cell type-specific manner (c). We defined the *Cell type score* as following:

$$Cell\ type\ score(r,c) := \frac{\beta_r \cdot DNase1\text{-}signal_{r,c}}{\sum_{r_i \in R} |\beta_{r_i} \cdot DNase1\text{-}signal_{r_i,c}|}.$$

The regression coefficient (β) describes the association between a REM and its gene's expression. The DNase1 signal is log-transformed and standardized for each REM over all cell types (mean=0, standard deviation=1) and represents how active a REM is in a cell type c. R is defined as the set of all REMs associated to a given gene, thus $R = r_1, ..., r_n$. The *Cell type score* normalizes the contribution of REM r to its gene's expression in this specific cell type as predicted by the linear model of STITCHIT. Notice, that the sum of the absolute value of the *Cell type score* of REMs associated to a gene for one cell type c, adds up to 1. So, we do not expect to observe *Cell type scores* to be 1 or close to 1, as we do for the *Model score*. A positive *Cell type score* indicates an expected increase of the gene's expression in comparison to the other considered cell types and a negative value has a decreasing effect. There are two possible scenarios when observing a positive cell type score:

1) The REM has a positive regression coeffiecient and the cell type's DNase1 signal is higher than the mean over all considered cell types (positive value). This means that the REM is an activator and the chromatin is open, so the REM is likely to enhance the gene's expression in comparisons to cell types, where the chromatin is more closed.

2) The REM has a negative regression coefficient and the cell type's DNase1 signal is lower than the mean over all considered cell types (negative value). In other words, the REM is a repressor of the gene, but the chromatin is rather closed, so the REM is most likely not able to regulate the gene's expression. This leads to a higher gene expression in comparison to cell types where the chromatin is more open.

There are also two scenarios to observe a negative cell type score:

1) The REM has a positive regression coefficient and the cell type's DNase1 signal is lower than the mean over all considered cell types (negative value). This means that the REM is interpreted as an activator, but the chromatin is closed. Thus, the REM is most likely not able to regulate the expression of the gene. Consequently, the gene expression is decreased in comparsion to a cell type where the chromatin is more open.

2) The REM has a negative regression coefficient and the cell type's DNase1 signal is higher than the mean over all considered cell types (positive value). Therefore, the REM is a repressor and the chromatin is rather open. This leads to a decreasing gene expression in comparsion to a cell type where the chroamtin is more closed.

The following table summarizes how to interpret the *Cell type score*:

Sign of Regression coefficient	Predicted function of REM based on regression coefficient	Sign standardized DNase1 signal (mean = 0, std = 1)	Chromatin state based on standardized DNase1 signal	Sign of the Cell type score	Predicted effect on the gene's expression
positive	activating	positive	open	positive	increased
negative	repressing	negative	closed	positive	increased
positive	activating	negative	closed	negative	decreased
negative	repressing	positive	open	negative	decreased

The *Cell type score* can be used to rank REMs according to their importance between cell types for the same gene or to rank REMs within one cell type.

8.7.8 Cell type DNase1 signal

Cell type DNase1 signal is the DNase1 signal for the cell type of interest measured in the REM region. It is normalized for sequencing depth and can be used to compare the activity of REMs between samples. As we can have more than one sample for each cell type, we take the average activity of those samples. The activity was obtained from the Roadmap and Blueprint consortia and is no parameter calculated by STITCHIT.

CHAPTER 9

Application scenarios

In this section we provide a step-by-step explanation of application scenarios of EpiRegio. Two scenarios are similar to those in our paper *EpiRegio: Analysis and retrieval of regulatory elements linked to genes* (currently in revision).

9.1 How to use EpiRegio to identify TF's target genes using ChIP-seq peak regions

The application scenario is based on the section *Elucidation of disease pathways directly from a TF-ChIP experiment* from our paper.

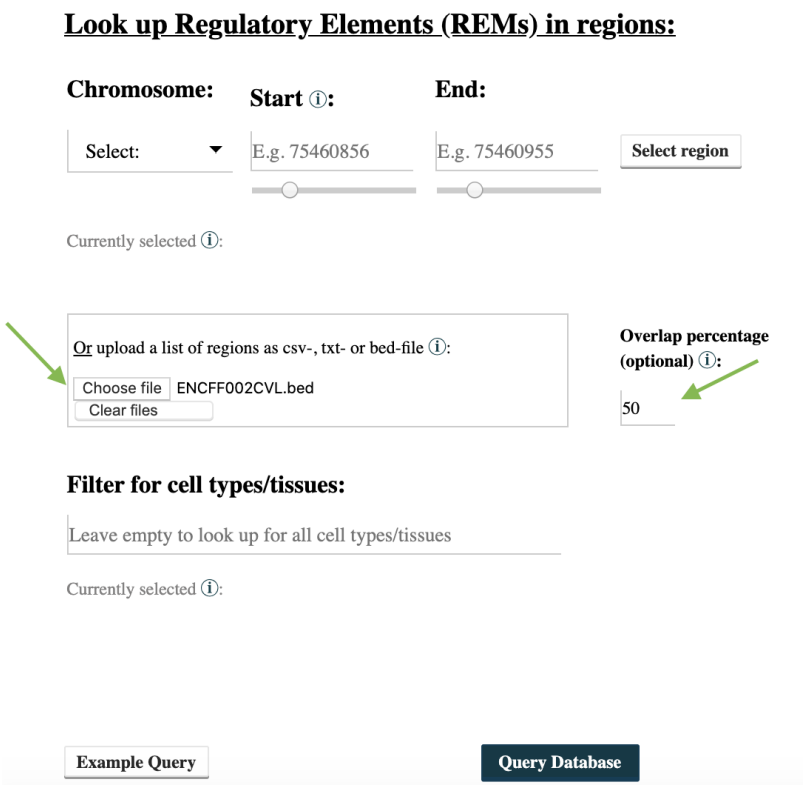
Step 1: Download the binding locations of the TF of interest, for instance from the ENCODE database as a BED file. As an example, we use the ChIP-seq peaks of TF ARID3A with the accession number ENCFF002CVL. Either click here to get the data from the ENCODE webpage or download it via:

```
wget 'https://www.encodeproject.org/files/ENCFF002CVL/@@download/ENCFF002CVL.bed.gz'.
```

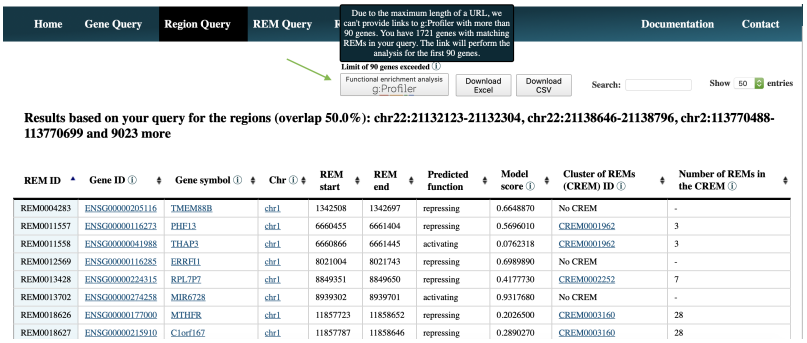
Unzip the file using e.g.:

```
gzip -d ENCFF002CVL.bed.gz
```

Step 2: Use EpiRegio's Region Query to search for REMs overlapping at least 50% with the TF ChIP-seq peaks. Go to https://epiregio.de/regionQuery/, click *choose File* and upload the unzipped ChIP-seq peaks from Step 1. Next to the upload field, you can see an option *Overlap percentage (optional)* to define the percentage that a REM should by minimum overlap with the binding locations (50% of the binding location's length). Since we want a 50% overlap, type 50 in this field and click *Query Database*.



Step 3: Click the bottom *Functional enrichment analysis g:Profiler* in the upper left corner, to perform a GO term enrichment analysis using g:Profiler (default parameters) of the resulting REMs. Notice, if the resulting genes are more than 90, the maximal possible length of a url is exceeded. Therefore, the first *90* genes are considered.



If you want to perform the analysis with all identified genes, please download the result by clicking *Download Excel*, open the excel file and copy the column named *Gene ID*. Go to https://biit.cs.ut.ee/gprofiler/gost and paste the copied gene IDs in the field over the *Run query* button. Then select *Run query*. Duplicate gene IDs will only be considered once.

9.2 How to use EpiRegio to identify enriched TFs of a set of genes of interest

The application scenario is based on the section *Identify enriched transcription factors of differentially expressed genes* from our paper. To perform the analysis python3 and bedtools must be installed on your machine. You also need a current version of a human genome in fasta format, which can, for example, be downloaded on the UCSC webpage.

In addition, we provide a GitHub repository with an example file, the TF binding motifs and the motif enrichment tool PASTAA, which we use in Step 4. To clone the repository use:

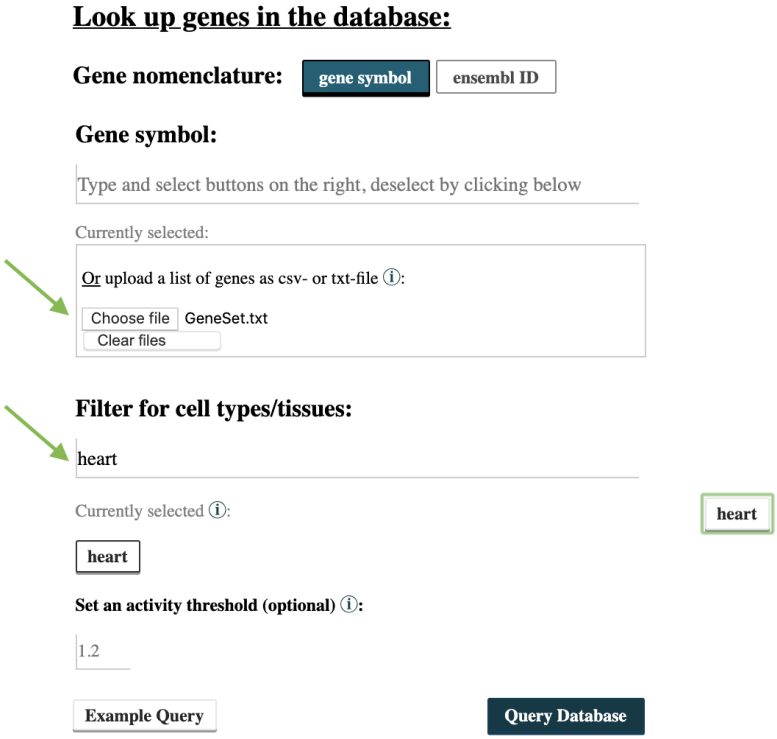
```
git clone https://github.com/TeamRegio/ApplicationScenarioExamples.git
```

In the repository the TRAP version, a script used by PASTAA, is slightly changed. We normalized the resulting TRAP affinities by the TF binding motif length. Next go to the src folder in the cloned repository and compile PASTAA via:

```
cd ApplicationScenarioExamples/src/
make
```

As an example, we consider a set of differential expressed genes based on a single-cell RNAseq data set from Glaser et al. (doi.org/10.1073/pnas.1913481117), where Human Umbilical Endothelial Cells (HUVECs) were treated with TGF-beta to trigger an endothelial-to-mesenchymal transition (EndoMT). However, the analysis works with every set of genes. If you want to perform the example, please have a look at the folder *identifyEnrichedTFs* in our GitHub repository where we provide a file called *GeneSet.txt* containing this set of genes.

Step 1: Use EpiRegio's Gene Query to identify the REMs associated to the genes of interest. Go to https://epiregio.de/geneQuery/, click *choose File* and upload the file from Step 1. Enter *heart* in the field *Filter for cell types/tissues*. We choose heart as tissue as endothelial cells within the heart undergo EndoMT during cardiac development and we expect the regulatory processes to be comparable. If you are using an individual data set, please also choose a cell type or tissue which is most suitable for your data. Next click *Query Database*.



Step 2: Download the resulting table by clicking on the bottom *CSV*.

Results based on your query for the genes: TAGLN, TPM1, IGFBP7 and 301 more

Gene ID	Gene symbol	REM ID	Chr	REM start	REM end	Predicted function	Model score	Cluster of REMs (CREM) ID	Number of REMs in the CREM	Heart score (n=2)	Heart DNase1 signal (n=2)
ENSG00000005022	SLC25A5	REM2381079	chrX	119370960	119370999	repressing	0.5558670	No CREM	-	0.0162728	0.0
ENSG00000005022	SLC25A5	REM2381080	chrX	119371100	119371299	repressing	0.5960380	No CREM	-	0.0060397	0.103216
ENSG00000005022	SLC25A5	REM2381082	chrX	119373010	119373049	activating	0.5971500	No CREM	-	-0.0088053	0.233404
ENSG00000005022	SLC25A5	REM2381102	chrX	119388000	119388049	repressing	0.7620160	No CREM	-	0.0176583	0.2422006
ENSG00000005022	SLC25A5	REM2381107	chrX	119390900	119390949	activating	0.6696590	CREM0362187	2	-0.0006862	0.2023035
ENSG00000005022	SLC25A5	REM2381117	chrX	119397850	119397999	activating	0.9541260	No CREM	-	-0.0334666	0.216979

Step 3: Next, we determine the DNA-sequence of the identified REMs using *bedtools* and run *PASTAA* to perform the motif enrichment analysis. In our GitHub repository we provide a workflow to run the analysis and a set of TF binding motifs downloaded from the JASPAR database (version 2020). To run the workflow the following command can be used:

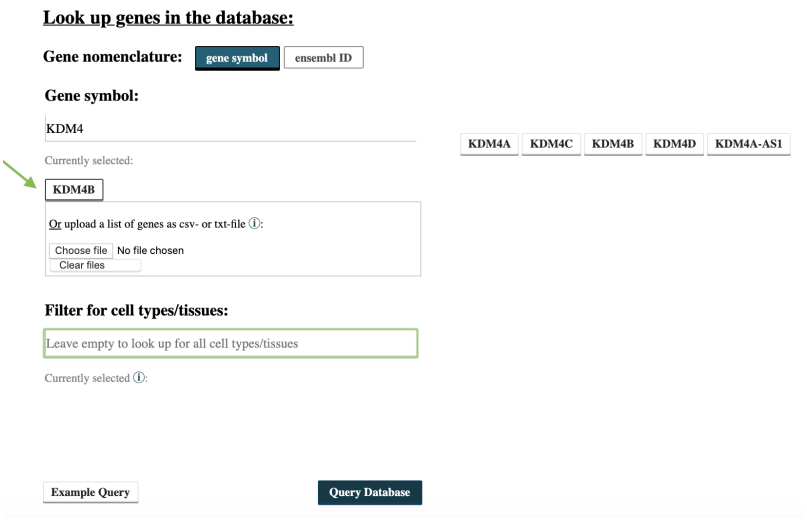
```
bash <pathToClonedRepo>identifyEnrichedTFs/workflow.sh <Motifs> <pathToClonedRepo>
↪<pathToGenome> <REMs> <outputDir> <pvalue>,
```

where <*pathToClonedRepo*> represents the path to the cloned repository and <*Motifs*> the path to the TF motif file. You can either use the motif file we provide in our repository (ApplicationScenarioExamples/identifyEnrichedTFs/JASPAR2020_HUMAN_transfac.txt) or a self-chosen one. The motifs should be in TRANSFAC format. <*pathToGenome*> is the path to the fasta file of the human genome, <*REMs*> the path to the downloaded csv-file, and <*output*> the path to a user-defined output folder. If the Benjamini-Hochberg adjusted p-value from PASTAA is smaller than or equal to the parameter <*pvalue*> the motif is assumed to be significant enriched. For this example, set the <*pvalue*> to 0.05. The resulting significant enriched TF motifs are stored in <outputDir>/PASTAA_result.txt.

9.3 How to use EpiRegio to identify TF-binding sites within REMs of a gene of interest

To perform the analysis bedtools must be installed on your machine. You also need a current version of a human genome in fasta format, which can, for example, be downloaded on the UCSC webpage.

Step 1: Use Epiregio's Gene Query to identify REMs associated to your gene of interest. In this example we want to perform the analysis for the gene KDM4B. Go to https://epiregio.de/geneQuery/, enter KDM4B in the field *Gene symbol*. After typing several letters, gene names containing the entered letters will appear. Click at KDM4B and the gene name is listed under *Currently selected*. Next select *Query Database*.



Step 2: After the query is done, download the table with the resulting REMs by clicking on the bottom *CSV*. Before we can determine the DNA-sequence of the REMs, we need to format the CSV file to a bed file with the following command:

```
awk 'NR!=1{print $4 "\t" $5 "\t" $6}' <yourCSVFile>  >REMs.bed,
```

where *<yourCSVFile>* represents the file you just downloaded from the server. Using bedtools getFasta command, we are able to extract the DNA-sequences of the REMs:

```
<pathToBedtools>/bedtools getfasta -fi <humanGenome> -bed REMs.bed -fo REMs.fa
```

<pathToBedTools> represents the path to your bedtools source folder (if not included in your environment variables) and *<humanGenome>* the path to a file containing the human genome in fasta format.

Step 3: To identify TF-binding sites, we use the tool *Fimo* from MEME suite. *Fimo* requires the DNA-sequences of the REMs from Step 2 and a set of known TF binding motifs. In our GitHub repository we provide human motifs from the JASPAR database (version 2020) in meme format. You can clone the repository using:

```
git clone https://github.com/TeamRegio/ApplicationScenarioExamples.git
```

The TF-binding motif file is located in *ApplicationScenarioExamples/identifyTFBindingSites/JASPAR2020_HUMAN_meme.txt*. Go to http://meme-suite.org/tools/fimo, in the section *Input the motifs* click *choose file* and upload the motifs. Next click at *Ensembl Ab initio Predicted Proteins* in the section *Input the sequences* and select *Upload sequences*. A field where you can upload the DNA-sequences will appear. To do so, select *Choose file* and upload the fasta file from Step 2. Click *Start search*. Note that it can take some minutes until the calculations are done.

CHAPTER 10

REST API

EpiRegioDB provides a user-friendly REST framework based web interface to retrieve information from our database. This browsable interface provides information as a JSON file.

10.1 General Information

The REST API allows 3 different kinds of queries (GeneQuery, RegionQuery and REMQuery), which have similar functionalities as the corresponding queries of the web interface (Gene ID, Genomic region and Regulatory element). Furthermore, there is a query that reports all REMs that belong to a cluster of REMs (CREMQuery) and a query that provides general information of a gene (GeneInfo). All queries follow the same syntax rule:

```
https://epiregio.de/REST_API/<query>/<input>/
```

where *input* represents the input of the current query. For instance, if you are interested in which REMs are linked to the gene ENSG00000223972, then *query* is *GeneQuery* and *input* is *ENSG00000223972*, which results in the following URL:

https://epiregio.de/REST_API/GeneQuery/ENSG00000223972/

In addition, it is possible to request information for multiple inputs within one run. Therefore, the inputs need to be separated by an underscore '_'. This can be done as follows

https://epiregio.de/REST_API/GeneQuery/ENSG00000223972_ENSG00000223974/

and returns all REMs associated to the genes ENSG00000223972, and ENSG00000223974. The following provides more information as well as an example for each of the query types.

10.2 GeneQuery

Providing an ensembl gene ID or gene symbol (or multiple ones) as input, this query returns the associated REMs. In detail, the geneID, geneSymbol, REMID, chr, start, end, regressionCoefficient, p-value, normalized model score, version of the Epiregio database, number of REMs per CREM, CREM ID, and a list of the cell type score and the

DNase1 signal of the cell type used in STITCHIT are displayed. We have a section explaining all of the results in detail.

10.2.1 Example

Please have a look at the *General Information* section above for an example.

10.3 RegionQuery

By providing a genomic region, this query returns all REMs that lie completely within this region. The genomic region must be given as chr:start-end, where start is smaller than or equal to end (e.g. chr16:75423948-75424405). The output has the same format as the *GeneQuery* output. Optionally, you can also hand an overlap value to the URL like this: RegionQuery/50/... which retrieves all REMs that overlap with the regions by at least 50% of the region's length.

10.3.1 Example:

https://epiregio.de/REST_API/RegionQuery/chr16:75423948-75424405/

https://epiregio.de/REST_API/RegionQuery/50/chr16:75423948-75424405/

https://epiregio.de/REST_API/RegionQuery/chr16:75423948-75424405_chr2:1369428-1369900/

10.4 REMQuery:

This query answers the question, which gene is linked to a given REM. Therefore, the input must be a valid REM ID (e.g REM0000006). As it was for the *GeneQuery* and the *RegionQuery* before, multiple inputs are possible, and the output has the same format.

10.4.1 Example:

https://epiregio.de/REST_API/REMQuery/REM0000002/

https://epiregio.de/REST_API/REMQuery/REM0000002_REM0000007_REM0000009/

10.5 CREMQuery

Given a CREM ID (e.g CREM0000007) or multiple CREM IDs (e.g CREM0000002_CREM0000008), this query lists all REMs contained in the CREM(s). The output format is the same as for the *GeneQuery*.

10.5.1 Example:

https://epiregio.de/REST_API/CREMQuery/CREM0000002/

https://epiregio.de/REST_API/CREMQuery/CREM0000002_CREM0000008_CREM0000009/

10.6 GeneInfo

For a given ensembl ID (or multiple ones), the query returns general gene information such as chr, start, end, gene symbol, alternative gene id, strand, and annotation version.

10.6.1 Example:

https://epiregio.de/REST_API/GeneInfo/ENSG00000223972/

https://epiregio.de/REST_API/GeneInfo/ENSG00000223972_ENSG00000223978/

10.7 Programmatic access via Python

If you wish to call the REST API outside of your browser, for example if you need to get data regularly and want to include it into one of your scripts, you need a program that is capable of doing HTTP requests. One easy-to-use tool is the Python package Requests. Let's go through an example: You have a Python list with genomic regions and you want to know which REMs cover at least 50% of your region's length. In the end you want to have a new list containing the REM IDs, their location as well as their cell type score for the left kidney. So here is what we need to get going:

```
import requests

important_regions = [['chr16', 75423948, 75424405], ['chr2', 1369428, 1369900], ['chr1
↪', 8000, 25999]]
overlap = 50
important_results = []  # Let's already define our output
```

Requests is straightforward to use, pass an URL to the requests.get() function and proceed with it as you need it. In our case this could look like this:

```
for region in important_regions:
        our_url = 'https://epiregio.de/REST_API/RegionQuery/'+str(overlap)+'/
↪'+region[0]+':'+str(region[1])+'-'+str(region[2])+'/'
        api_call = requests.get(our_url)
        if api_call.status_code != 200:  # In case the page does not work properly.
                print("Page Error")
        for hit in api_call.json():
                important_results.append([hit['REMID'], hit['chr'], hit['start'], hit[
↪'end'], hit['cellTypeScore']['left kidney']])
```

CHAPTER 11

Known Issues

Here we discuss some of the known issues, and what you can do to rectify it. All issues we discuss here are what we have learned from our testing on the browsers/OS listed below.

OS/Browser	Chrome	Edge	Firefox	Safari
Linux	x		x	
MacOS	x		x	
Windows	x	x	x	

11.1 Layout issues

If you find elements on the website overlapping with each other, you could try clearing the cache of your browser or try it in private mode. If it still overlaps, try updating to a newer version of your browser. We developed our website in the newest browser version in private mode.

11.2 Server Error (500)

Issue: One of the parameters you have set is wrong!

Solution: If you are using Chrome, please try to clear the cache in your browser, and try again. Still the issue persists? Please check all your inputs, and the options you selected.

We are aware of issues causing a Server Error 500 if the input list is too large. We are working on solving this issue. In the meantime, unfortunately, you might have to try to provide your input in smaller chunks.

11.3 The issue with the gene name – ensembl id mapping

In general, we advise to use the ensembl ids of the genes of interest instead of the gene name, since EpiRegio does not store all gene name aliases. Especially, if you are interested in the gene Y_RNA, please stick to the ensembl id

to avoid any confusions. EpiRegio stores REMs for roughly 300 genes with the gene name Y_RNA (according the gencode.v26.annotation.gtf file). However, all of them have a different ensembl id to identify them uniquely.

If you face other problems, please let us know through GitHub issues!