EpiRegioDB Supplemental Material

Release 1.0

Nina Baumgarten, Dennis Hecker, Sivarajan Karunanithi, and Mar

Apr 15, 2020

Contents

CHAPTER 1

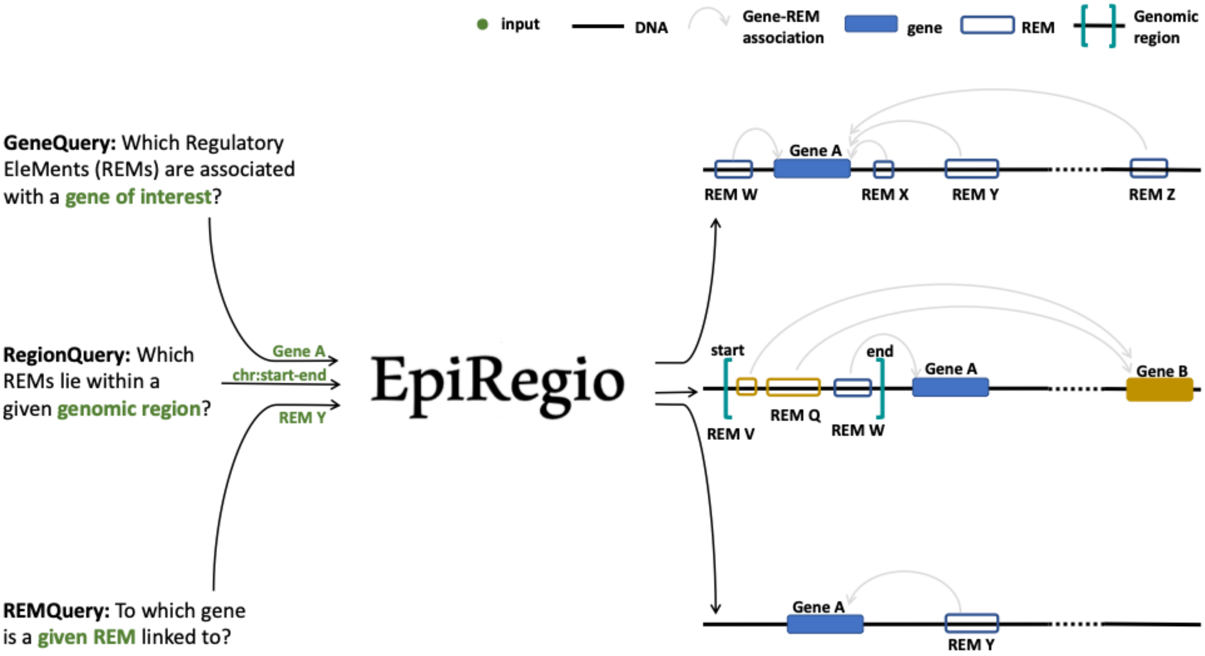
About EpiRegio Web Server

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 by remodelling the accessibility of chromatin. They are able to support 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 get a clear readout of their sequence. 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 its 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


● input
DNA
Gene-REM association
gene
REM
Genomic region
GeneQuery: Which Regulatory EleMents (REMs) are associated with a gene of interest?
RegionQuery: Which REMs lie within a given genomic region?
REMQuery: To which gene is a given REM linked to?
Gene A
chr:start-end
REM Y
EpiRegio
Gene A
REM W
REM X
REM Y
REM Z
start
end
Gene A
Gene B
REM V
REM Q
REM W
Gene A
REM Y

CHAPTER 3

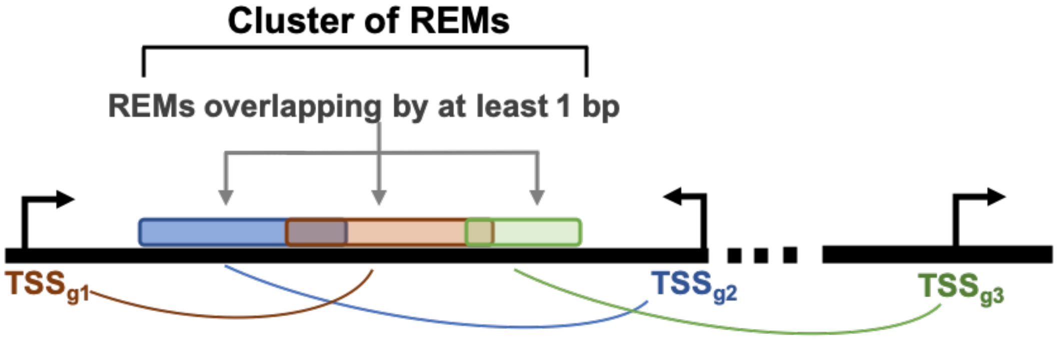
Learning of Regulatory EleMents (REMs)

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. This does not mean that we assume each separate regulatory region in the genome to be associated to just one gene. Regions are not exclusive to REMs, other REMs can overlap with them. 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 without any break in between (see the schema above). A CREM ends when there is no overlapping 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 and we know which part of it links to which gene.

CHAPTER 5

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.

CHAPTER 6

Cite Us

If you use this webserver, please cite the following:

1. Baumgarten et. al., Analysis and retrieval of regulatory elements linked to genes with EpiRegio
2. Schmidt et. al., Integrative analysis of epigenetics data identifies gene-specific regulatory elements

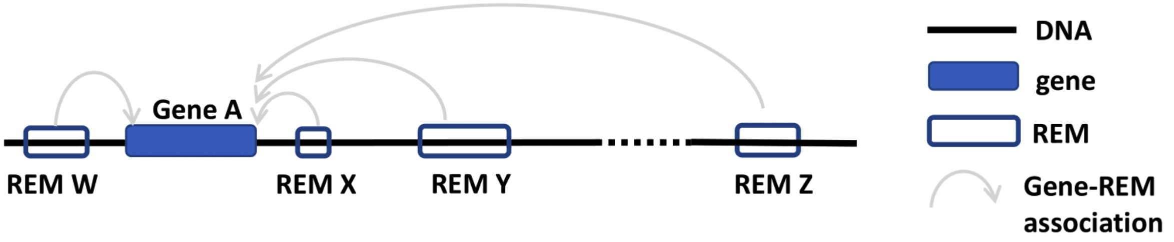
CHAPTER 7

Query Guide

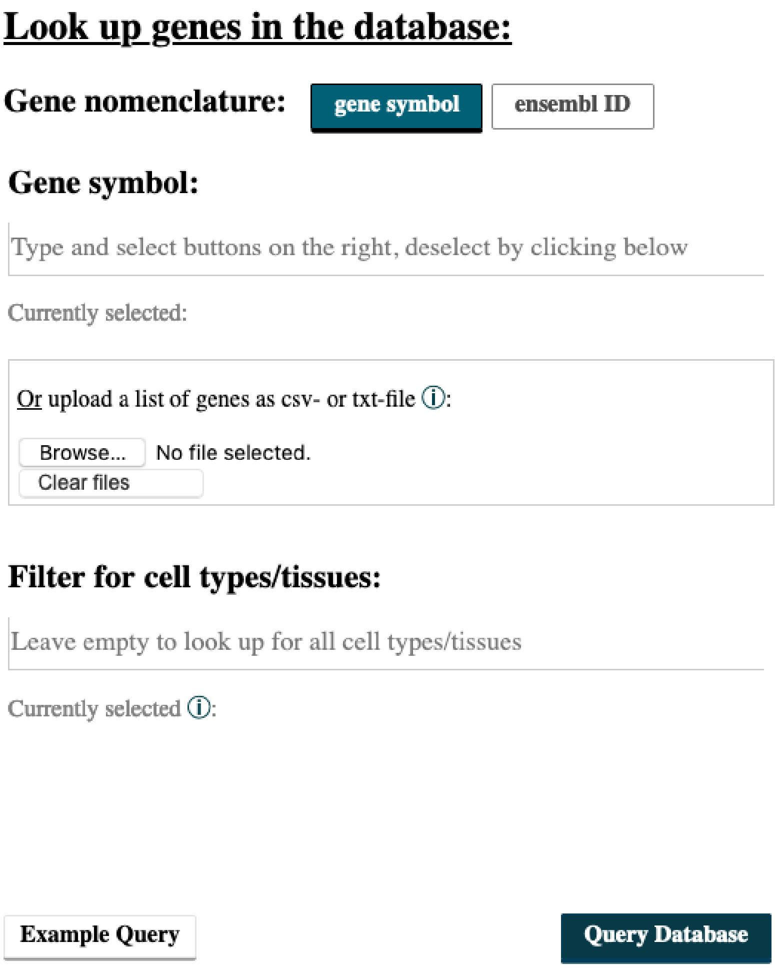
Here we provide a step-by-step guide for every query available, including an explanation of the output. Every query has an *Example Query* button at the bottom of the page. Try it out to see how a valid query would look like. Have a look at *Results in detail* to get an explanation of all the output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

7.1 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 version hg38.



Look up genes in the database:

Gene nomenclature:

Gene symbol:

Type and select buttons on the right, deselect by clicking below

Currently selected:

Or upload a list of genes as csv- or txt-file ⓘ:

No file selected.

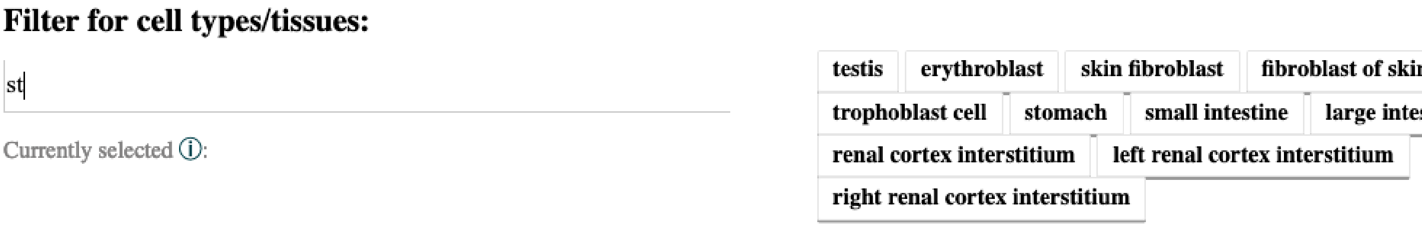
Filter for cell types/tissues:

Leave empty to look up for all cell types/tissues

Currently selected ⓘ:

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, a new input field will appear, which gives the option to choose an activity threshold. This threshold refers to the DNase activity of the REMs in the cell types/tissues. Only REMs that exceed the threshold in **ALL** of the cell types you selected will be shown in the output table. Leave the field empty to get back all REMs independent of their activity.

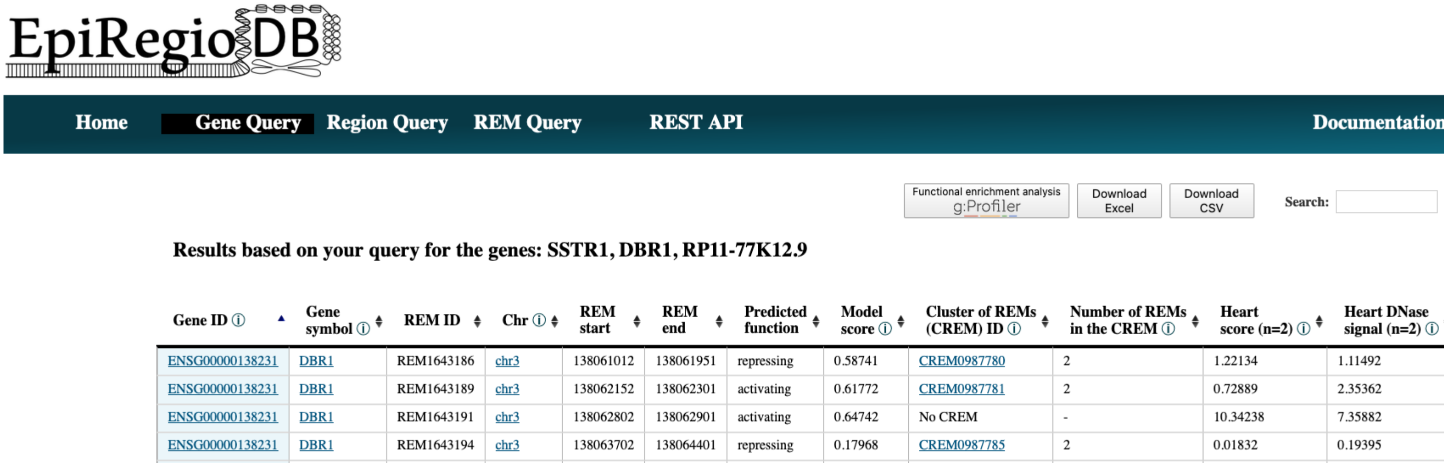


Filter for cell types/tissues:

st

Currently selected ⓘ:

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* indicates how important a REM is for its associated gene over all cell types. The higher the value, 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 overlap by at least 1 bp. 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 activity* 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* is the absolute product of the regression coefficient and the DNase activity, indicating how important a REM is in this cell type. The higher the value, the higher the REMs expected contribution to its gene's expression in this cell type. *Cell type activity* is the DNase signal alone, 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. 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.

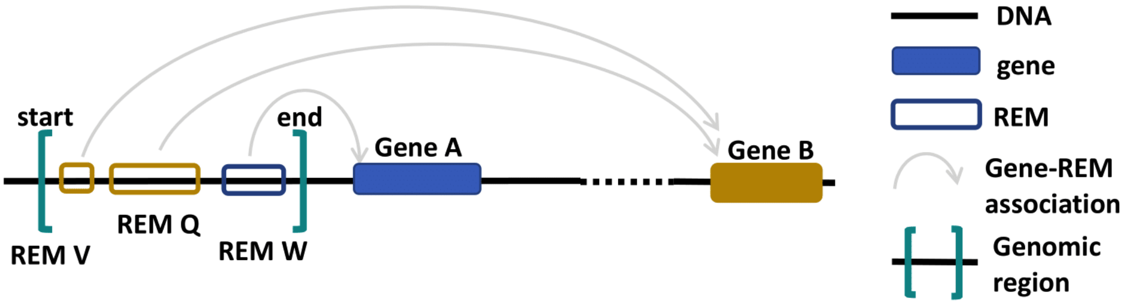


Results based on your query for the genes: SSTR1, DBR1, RP11-77K12.9

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 DNase signal (n=2) ⓘ
ENSG00000138231	DBR1	REM1643186	chr3	138061012	138061951	repressing	0.58741	CREM0987780	2	1.22134	1.11492
ENSG00000138231	DBR1	REM1643189	chr3	138062152	138062301	activating	0.61772	CREM0987781	2	0.72889	2.35362
ENSG00000138231	DBR1	REM1643191	chr3	138062802	138062901	activating	0.64742	No CREM	-	10.34238	7.35882
ENSG00000138231	DBR1	REM1643194	chr3	138063702	138064401	repressing	0.17968	CREM0987785	2	0.01832	0.19395

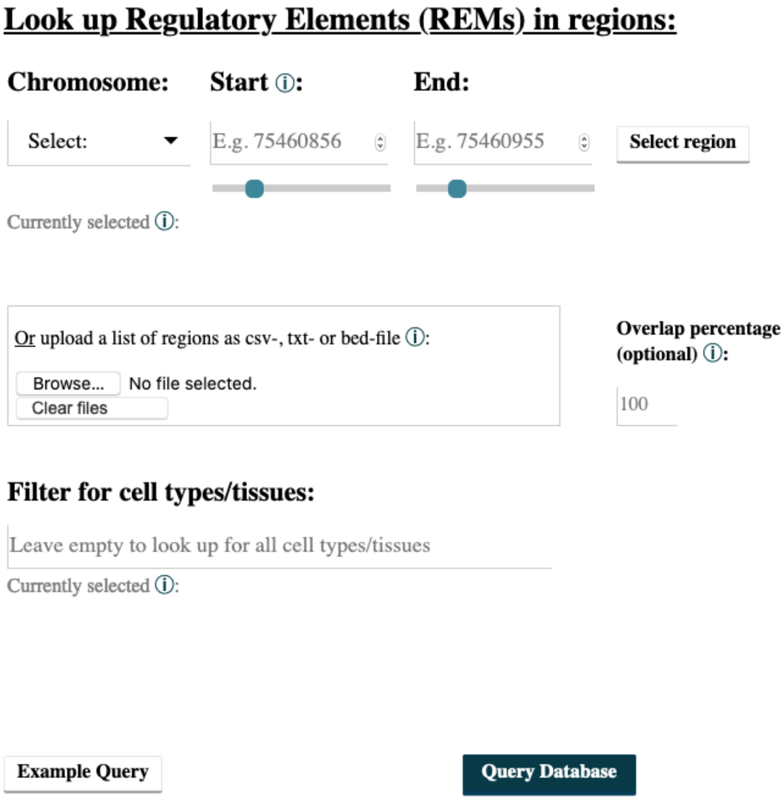
7.2 Region Query

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

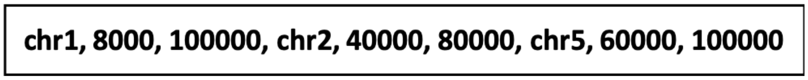


1. Go to the *Region Query* tab.
2. You can enter a region by choosing a chromosome, the start and the end point and then clicking on the *Select* button. Add as many regions as you like. Deselect your choices by reclicking on the added buttons. Only REMs

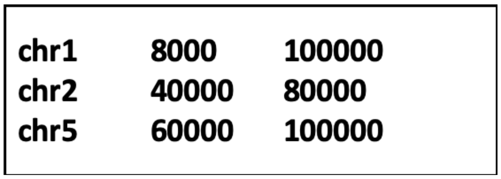
that are located in your chosen regions will be given as output. You can select the percentage of overlap and by this define how much of the REM has to overlap with your selected regions to be shown in the output. For example, with an overlap of 50% only the REMs that overlap by at least half of their length with your selected regions will be returned.


Look up Regulatory Elements (REMs) in regions:
Chromosome:
Start ⓘ:
End:
Select:
E.g. 75460856
E.g. 75460955
Select region
Currently selected ⓘ:
Or upload a list of regions as csv-, txt- or bed-file ⓘ:
Browse...
No file selected.
Clear files
Overlap percentage (optional) ⓘ:
100
Filter for cell types/tissues:
Leave empty to look up for all cell types/tissues
Currently selected ⓘ:
Example Query
Query Database

3. You can also upload a csv-, txt- or bed-file with your regions of interest in which the first value has to be the chromosome, followed by the start and the end position. A combination of both, input field and uploaded file, is not implemented. You can see the format of exemplary upload files below (comma-separated and tab-separated). All of the commonly used separators are being recognized, as long as the order of chromosome, start position and end position is correct. For the bed-files, 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. Files with empty fields will not be read correctly.



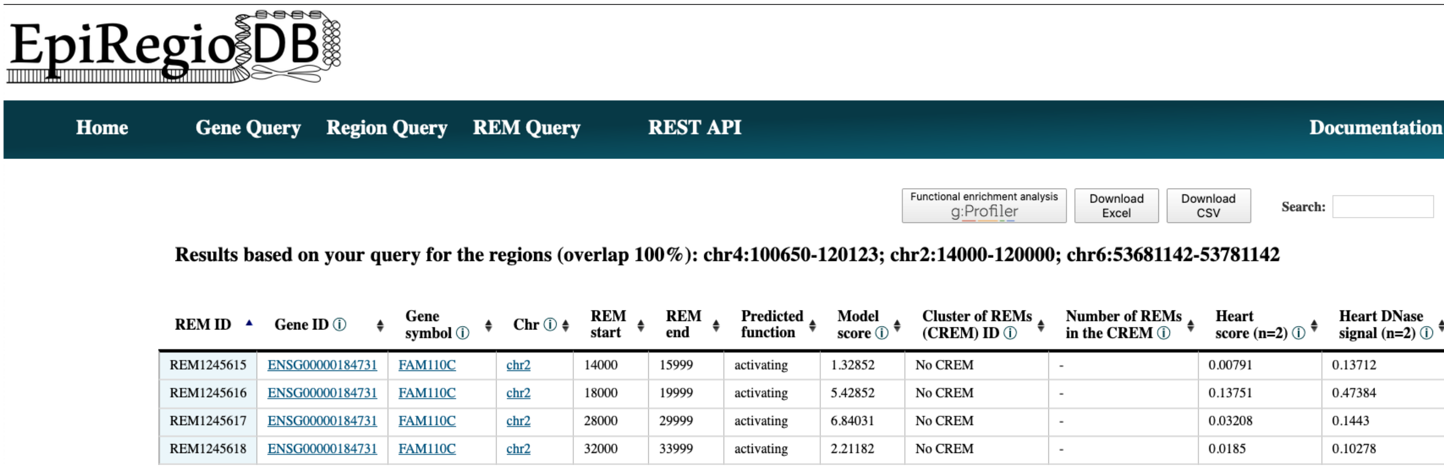
```
chr1, 8000, 100000, chr2, 40000, 80000, chr5, 60000, 100000
```



```
chr1    8000    100000
chr2    40000   80000
chr5    60000   100000
```

4. Choosing cell types/tissues: The selection of cell types functions in the same way as described above in the **:href:'Gene Query'** at point 4.

5. The output is very similar for all queries. Have a look at point 5 of the **:href:'Gene Query'** or at the **:href:'Results in detail'**. Below you can see how the output of the Region query looks like.

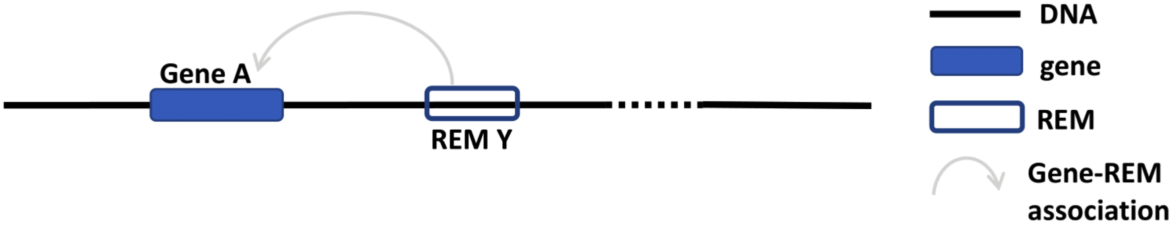


Results based on your query for the regions (overlap 100%): chr4:100650-120123; chr2:14000-120000; chr6:53681142-53781142

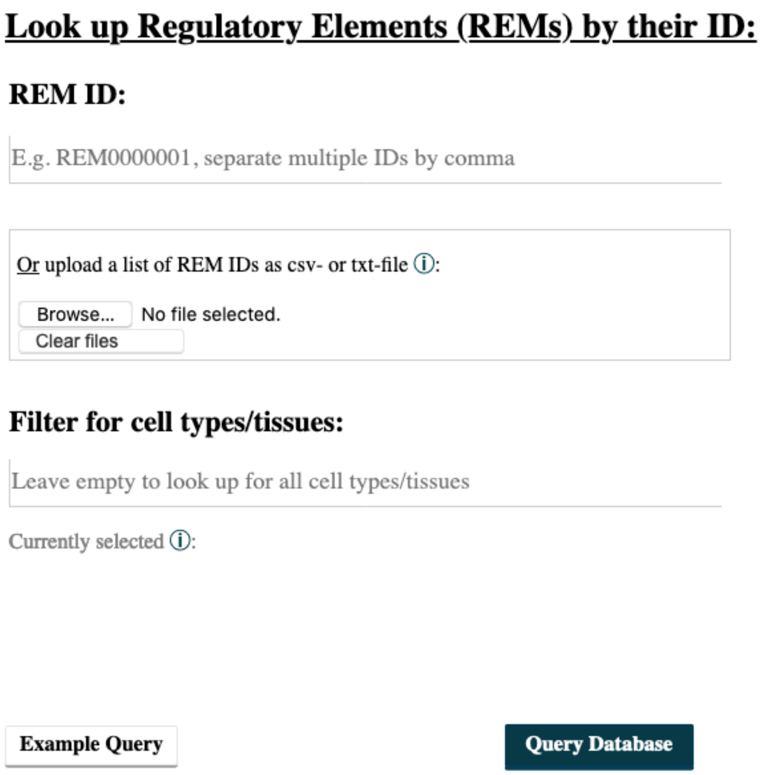
REM ID	Gene ID ⓘ	Gene symbol ⓘ	Chr ⓘ	REM start	REM end	Predicted function	Model score ⓘ	Cluster of REMs (CREM) ID ⓘ	Number of REMs in the CREM ⓘ	Heart score (n=2) ⓘ	Heart DNase signal (n=2) ⓘ
REM1245615	ENSG00000184731	FAM110C	chr2	14000	15999	activating	1.32852	No CREM	-	0.00791	0.13712
REM1245616	ENSG00000184731	FAM110C	chr2	18000	19999	activating	5.42852	No CREM	-	0.13751	0.47384
REM1245617	ENSG00000184731	FAM110C	chr2	28000	29999	activating	6.84031	No CREM	-	0.03208	0.1443
REM1245618	ENSG00000184731	FAM110C	chr2	32000	33999	activating	2.21182	No CREM	-	0.0185	0.10278

7.3 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.



Look up Regulatory Elements (REMs) by their ID:

REM ID:

E.g. REM0000001, separate multiple IDs by comma

Or upload a list of REM IDs as csv- or txt-file ⓘ:

No file selected.

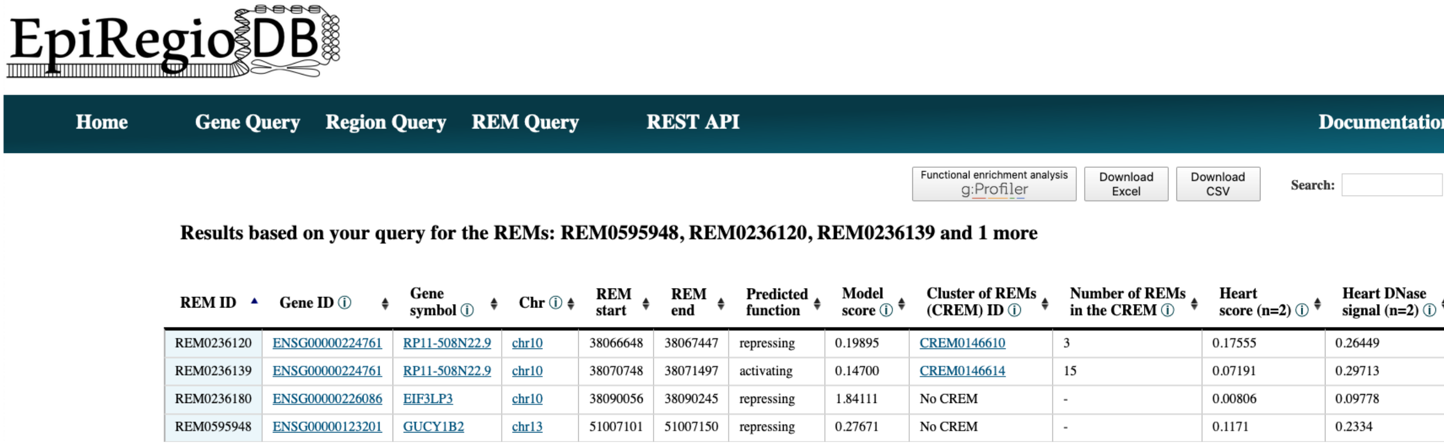
Filter for cell types/tissues:

Leave empty to look up for all cell types/tissues

Currently selected ⓘ:

3. Choosing cell types/tissues: The selection of cell types functions in the same way as described above in the **:href:'Gene Query'** at point 4.

4. The output is very similar for all queries. Have a look at point 5 of the **:href:'Gene Query'** or at the **:href:'Results in detail'**. Below you can see how the output of the REM query looks like.



Results based on your query for the REMs: REM0595948, REM0236120, REM0236139 and 1 more

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 DNase signal (n=2) ⓘ
REM0236120	ENSG00000224761	RP11-508N22.9	chr10	38066648	38067447	repressing	0.19895	CREM0146610	3	0.17555	0.26449
REM0236139	ENSG00000224761	RP11-508N22.9	chr10	38070748	38071497	activating	0.14700	CREM0146614	15	0.07191	0.29713
REM0236180	ENSG00000226086	EIF3LP3	chr10	38090056	38090245	repressing	1.84111	No CREM	-	0.00806	0.09778
REM0595948	ENSG00000123201	GUCY1B2	chr13	51007101	51007150	repressing	0.27671	No CREM	-	0.1171	0.2334

7.4 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. 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.

7.5 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

7.6 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.

7.6.1 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.

7.6.2 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.

7.6.3 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.

7.6.4 Model score

The *Model score* is the absolute binary logarithm of the p-value for the association between a REM and its target gene. It serves as an indicator on how important a REM is for the expression prediction of its target gene. The higher the score, the more impact the REM is supposed to have. This value is not cell type specific as it 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*.

7.6.5 Cluster of REMs (CREM) ID

As STITCHIT determine REMs for each gene seperately and not the other way around, the identified regions can overlap. A *REM cluster* is a region of neighbouring REMs that overlap by at least one base pair (bp). There have to be a minimum of two overlapping 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.

7.6.6 Number of REMs per CREM

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 overlapping other REM and therefore is not considered as a cluster.

7.6.7 Cell type score

Cell type score is the absolute product of the regression coefficient and the DNase1 activity in a REM, indicating how important a REM is in this cell type. The higher the value, the higher the REMs expected contribution to its target gene's expression in this cell type. The regression coefficient is not cell type-specific, but the DNase1 activity is. Therefore, 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.

7.6.8 Cell type DNase signal

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

CHAPTER 8

Application scenarios

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

8.1 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: Downloaded 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 by 50% with the TF-ChIP 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 the binding locations and the REMs should overlap. Since we want a 50% overlap, type 50 in this field and click *Query Database*. TODO: add screenshot where you see exactly this

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. TODO: add screenshot

8.2 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 the motif enrichment tool PASTAA and bedtools must be installed on your machine. TODO: add TRAP script with normalization on GitHub page

Step 1: As an example, we consider a set of differential expressed genes based on a single-cell RNAseq data set from Glaser et al. (cite), 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 you can download the differential expressed genes from our GitHub repository (link).

Step 2: Use EpiRegio's Gene Query to identify the REMs assoiacted to the genes of interest. Go to https://epiregio.de/geneQuery/, click *choose File* and upload the file from Step 1. Enter *heart* to the field *Filter for cell types/tissues*. We are interested in the regulatory effects of the REMs for the tissue heart because endothelial cells within the heart undergo EndoMT during cardiac development. If your are using an individual data set, please also choose a celltype or tissue which is most suitable for your data. Next click *Query Database*. TODO: add screenshot

Step 3: To apply *PASTAA*, we need a ranking of the resulting REMs. Therefore, we sort them in descending order based on the column *heart score*. To do so, click on the arrows next to *heart score*. Download the resulting table by clicking on the bottom *CSV*. TODO: add screenshot

Step 4: 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. Download the folder . . . (link). There we also provide a set of TF binding motifs downloaded from the JASPAR database (version 2020). To run the workflow the following command can be used:

```
bash workflow.sh <Motifs> <pathToPASTAA> <pathToBedtools> <REMs> <outputDir> <pvalue>,
```

where <*Motifs*> represents the path to the TF motif file, <*pathToPASTAA*> is the path to the PASTAA source folder, <*pathToBedtools*> is the path to the bedtools source folder, <*REMs*> is the path to the downloaded csv-file, and <*output*> is the path to a user-defined output folder. If the Benjamini-Hochberg adjusted p-value from PASTAA smaller or equal the parameter <*pvalue*> the TF 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. TODO: Addscreenshoot from result.

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

this is not a application scenario from the paper

Step 1: Gene Query

Step 2: bedtools getFasta

Step 3: use fimo, provide JASPAR motifs in meme format

CHAPTER 9

REST API

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

9.1 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.

9.2 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, version of the Epiregio database, number of REMs per CREM, CREM ID, and a list of the log2(dnase1 signal) of the cell type used in STITCHIT are displayed.

9.2.1 Example

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

9.3 RegionQuery

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 or equal than 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 their length.

9.3.1 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/

9.4 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.

9.4.1 Example:

> https://epiregio.de/REST_API/REMQuery/REM0000002/
>
> https://epiregio.de/REST_API/REMQuery/REM0000002_REM0000007_REM0000009/

9.5 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*.

9.5.1 Example:

> https://epiregio.de/REST_API/CREMQuery/CREM0000002/
>
> https://epiregio.de/REST_API/CREMQuery/CREM0000002_CREM0000008_CREM0000009/

9.6 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.

9.6.1 Example:

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

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

9.7 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 overlap by at least 50% with your regions. 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['cellTypeActivity']['left kidney']])
```

CHAPTER 10

Known Issues

We discuss here 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	

10.1 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.

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