Implementing a transcription factor interaction prediction system using the GenoMetric Query Language

Stefano Perna[•,1], Arif Canakoglu[§,1], Pietro Pinoli[†,1], Stefano Ceri[‡,1] and Limsoon Wong[*,2]

[1]DEIB - Politecnico di Milano, Via Giuseppe Ponzio 34/5, Milano, Italy
[2]School of Computing - National University of Singapore, 13 Computing Drive, Singapore 117417, Singapore

Short title: TF interaction prediction using GMQL

Abstract

Novel technologies and growing interest have resulted in a large increase in the amount of data available for genomics and transcriptomics studies, both in terms of volume and contents. Biology is relying more and more on computational methods to process, investigate and extract knowledge from this huge amount of data. In this work, we present the TICA web server (available at `http://www.gmql.eu/tica/`), a fast and compact tool developed to support data-driven knowledge discovery in the realm of transcription factor interaction prediction. TICA leverages both the GenoMetric Query Language, a novel query tool (based on the Apache Hadoop and Spark technologies) specialized in the integration and management of heterogeneous, large genomic datasets, and a statistical method for robust detection of co-locations across interval-based data, in order to infer physically interacting transcription factors. Notably, TICA allows investigators to upload and analyse their own ChIP-seq experiments datasets, comparing them both against ENCODE data or between themselves, achieving computation time which increases linearly with respect to dataset size and density. Using ENCODE data from three well-studied cell lines as reference, we show that TICA predictions are supported by existing biological knowledge, making the web server a reliable and efficient tool for interaction screening and data-driven hypothesis generation.

[•] stefano.perna@polimi.it
[§]arif.canakoglu@polimi.it
[†]pietro.pinoli@polimi.it
[‡]stefano.ceri@polimi.it
[*]wongls@com.nus.edu.sg

Keywords—transcription factor interaction, gene regulation, genomic computing, biostatistics, ChIP-seq analysis, data integration

1 Motivation

Gene expression in prokaryotes and eukaryotes determines almost every internal and external behaviour of the cell(s), from reaction to stimuli all the way to cell development and death. To modulate gene expression, cells have evolved different mechanisms. One of the most well known and studied is the activity of Transcription Factors (TFs): these proteins possess highly specific DNA-binding domains that they use to latch onto specific parts of the DNA. Once attached, TFs can enhance or repress RNA polymerase access to the DNA area encoding for a particular gene, thereby reducing or enhancing the amount of its expression. This is one of the most basic forms of regulation and is widely used across all species in the natural world; thus, it is of high interest for researchers to understand the role of each transcription factor in the regulatory machinery.

Transcription factors are known to implement their regulatory mechanisms in coordination, acting as functional groups. Ways to discover TF complexes include *in vivo* experiments, observation of live cells and testing potential interactors *in vitro*; however, given the intrinsic combinatorial nature of the problem, these approaches are unlikely to be complete or even feasible over the whole spectrum of TF-TF interactions. In the context of gene regulation, computational biology has become a powerful hypothesis generation tool, rooted in mathematical interpretation of experimental data: by screening unlikely interactions, the investigator can then focus resources on verifying the most interesting candidate interactors using more traditional methods.

In this chapter, we present the *TICA* (Transcriptional Interaction and Coregulation Analyser) web server, a convenient tool for analysing chromatin immunoprecipitation and sequecing dataset targeting TF binding locations and predicting TF-TF interaction. The TICA web server leverages two powerful assets:

- the expressive power of *GenoMetric Query Language* (GMQL) *[7]*, a novel high-level declarative language for seamless integration, management and querying of heterogeneous genomic datasets;
- a *statistical classifier* which predicts colocation between interval-based data on a single reference system by exploiting the structural and positional information given by the intervals themselves.

We developed TICA in the context of the TF-TF interaction prediction problem (hence the name), and therefore its model is tailored to the needs of this biological context. The TICA web server, developed in the *Django* framework, is available for both data exploration of ENCODE narrowpeaks on *Homo Sapiens* cell lines and for analysis of novel biological datasets, provided by biological investigators.

This Chapter is structured as follows: in **Section 2** we describe the web server, the main workflow and resulting output. In **Section 3**, we provide an overview of the implementation strategies we used to develop the web server and underlying algorithm, and discuss the advantages of using GenoMetric Query Language queries. In **Section 4**, we analyse the performance of the web server, in particular we describe datasets provided in the initial deployment and how the prediction algorithm scales with increasing amounts data provided by the user. Finally, in **Section 5**, we highlight the most interesting aspects of the web service in terms of performance, accuracy and acceptable data formats.

2 TICA web server

We have developed and deployed a web server (and related web application), with which investigators can use the TICA framework to predict TF-TF interaction on ChIP-seq datasets on a set of model cell lines from *Homo Sapiens*. The web server can be accessed at: `http://www.gmql.eu/tica/`. The web implementation can be employed in three ways:

1. users can investigate the latest version of ENCODE ChIP-Seq data available to search for evidence regarding interaction hypotheses;
2. they can upload their own TF ChIP-seq datasets to the database and analyse all possible interactor couples therein; or
3. they can upload their datasets and compare them with the ENCODE datasets, searching for potential interaction phenomena.

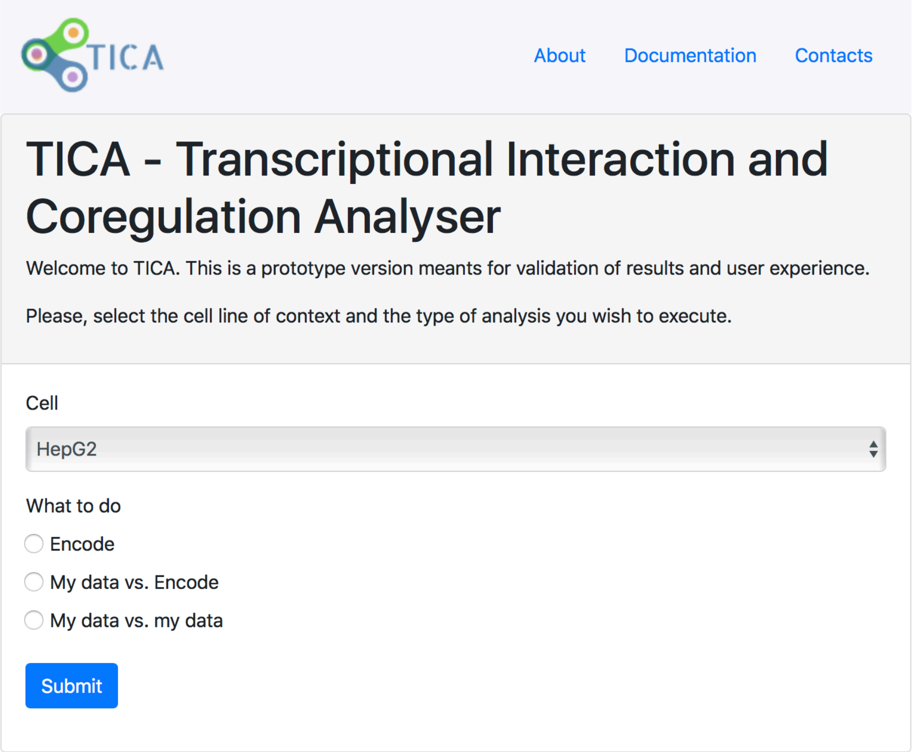
2.1 Workflow

Users connecting to the server see the welcome page reported in Figure 2.1. They are not required to create an account or authenticate in any way in order to use the web server: data uploaded is stored in a temporary folder (with a session ID for tracking during analysis), and subsequently discarded. In the welcome page, the user is prompted to select the context cell line: this determines the p-values for statistical tests (due to different null distributions) and the list of ENCODE TFs available for comparisons.

The workflow in the cases 1, 2 and 3 above is identical, except for the upload procedure required to submit, transform and filter user-provided datasets (see Section 3.1). Experimental data have to be uploaded via a single zip file containing one folder for each TF, which must be named as the TF itself. Each sample will be assigned to the TF inferred by its folder, regardless of the actual filename; single files should be in ENCODE bed narrow-peak format[•].

If the user selects ”ENCODE” in the main page , they will be immediately redirected to parameter selection.

[•] The schema for ENCODE narrowpeak data files is defined in `https://genome.ucsc.edu/FAQ/FAQformat.html#format12`



TICA - Transcriptional Interaction and Coregulation Analyser

Welcome to TICA. This is a prototype version meants for validation of results and user experience.

Please, select the cell line of context and the type of analysis you wish to execute.

Cell

HepG2

What to do

○ Encode

○ My data vs. Encode

○ My data vs. my data

Figure 1: Screenshot of TICA web application main page. Through the drop-down menu, the users can decide the context cell line among those available; users can also select whether they want to upload data or use ENCODE data.

2.2 Parameters

After uploading data (if required) users have to specify the parameters for the analysis using the parameter input page (Figure 2). A user can tune most of the TICA classifier parameters to suit biological assumptions or experimental conditions (cf. Table 2): among other choices, the user can restrict the analysis to a sublist of the TFs to be compared, define mindist couples maximum distance (from preselected values: 1100, 2200, 5500 bp), declare which test conditions have to be used (by ticking or unticking the corresponding test names) and state global significance level required and minimum number of test conditions to be satisfied (for additional details on the TICA classification algorithm, see Section 3.2). Default values are provided, matching specifications in Table 2.

2.3 Output

Results are presented to the user through a table and a heatmap (see Figure 3): the heatmap shows the number of test conditions satisfied, with -1 represents TF-TF pairs that do not meet the biological information screening criteria (see Section 3.2). Details on each feature extracted from observed mindistance couple distributions are given in a separate table, on the same page. Results can be exported as a .csv file using the ”Export to CSV” link (also in Figure 3).

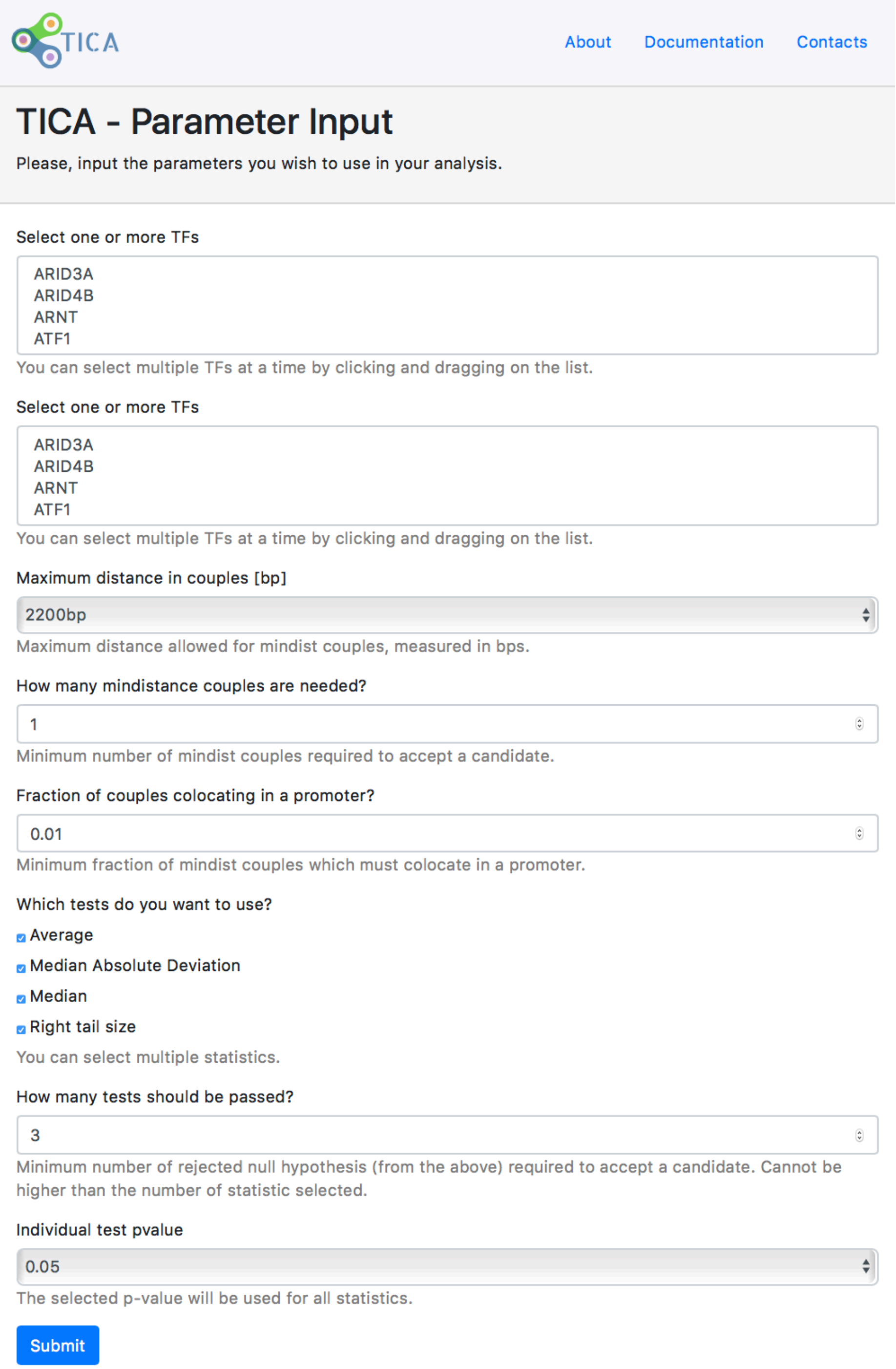
2.4 Deployment

All mindistance couples and related distances for the default cell lines in ENCODE data are precomputed and stored in a PostgreSQL database. These tables are only refreshed during major data updates; when user-provided data is uploaded in the system, only minimal distance couple distance distributions between TFs provided are computed on the fly. The server was developed using the Django v1.11.7 framework (`http://djangoproject.com`); queries are implemented inside the Django framework using the Python API for GMQL, PyGMQL *[9]*.

3 Implementation

The back-end supporting TICA is made of two conceptual blocks:

- a data preprocessing step, which takes either ENCODE or user-provided narrowpeaks and removes noisy binding sites and inactive transcription start sites, according to the context cell line (described in Section 3.1) and is implemented using GMQL;
- the prediction algorithm, a statistical procedure that compares candidate TF-TF pairs against null distributions from random pairs in the same cell line, with respect to a set of statistical aggregators (Section 3.2).



TICA - Parameter Input

Please, input the parameters you wish to use in your analysis.

Select one or more TFs

ARID3A
ARID4B
ARNT
ATF1

You can select multiple TFs at a time by clicking and dragging on the list.

Select one or more TFs

ARID3A
ARID4B
ARNT
ATF1

You can select multiple TFs at a time by clicking and dragging on the list.

Maximum distance in couples [bp]

2200bp

Maximum distance allowed for mindist couples, measured in bps.

How many mindistance couples are needed?

1

Minimum number of mindist couples required to accept a candidate.

Fraction of couples colocating in a promoter?

0.01

Minimum fraction of mindist couples which must colocate in a promoter.

Which tests do you want to use?

☑ Average
☑ Median Absolute Deviation
☑ Median
☑ Right tail size

You can select multiple statistics.

How many tests should be passed?

3

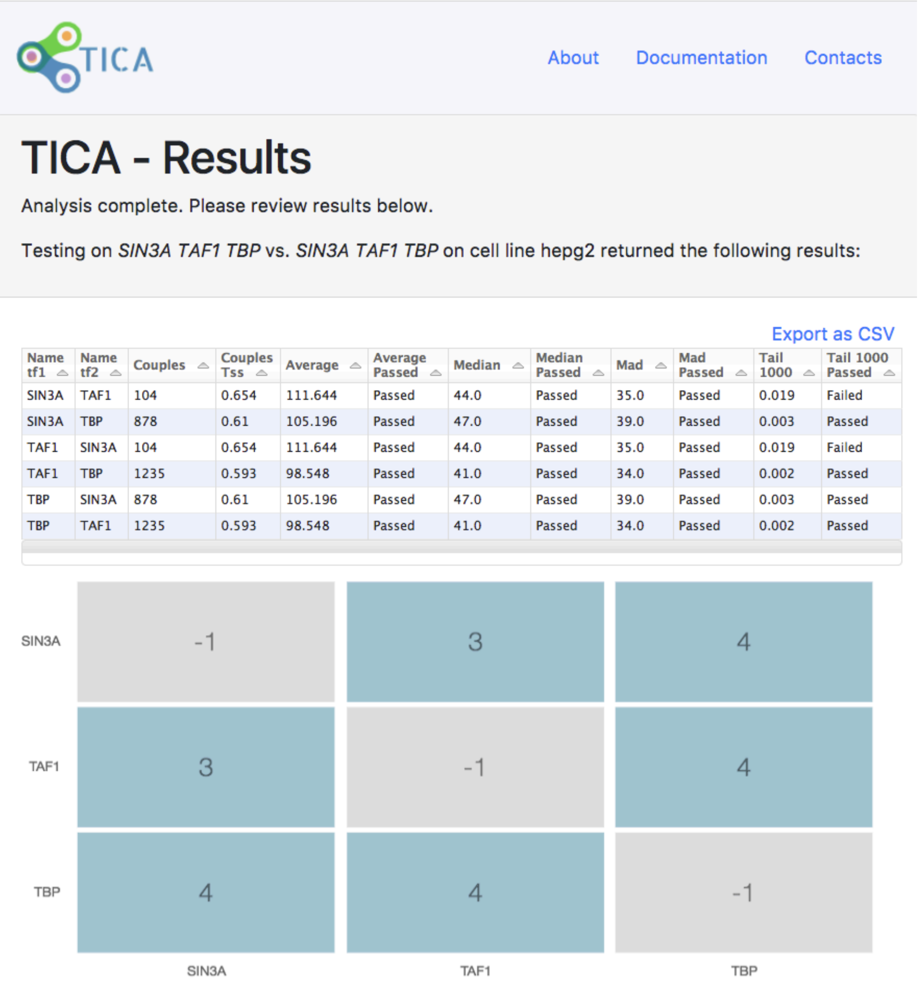
Minimum number of rejected null hypothesis (from the above) required to accept a candidate. Cannot be higher than the number of statistic selected.

Individual test pvalue

0.05

The selected p-value will be used for all statistics.

Figure 2: Screenshot of TICA parameter input page.



TICA - Results

Analysis complete. Please review results below.

Testing on *SIN3A TAF1 TBP* vs. *SIN3A TAF1 TBP* on cell line hepg2 returned the following results:

Name tf1	Name tf2	Couples	Couples Tss	Average	Average Passed	Median	Median Passed	Mad	Mad Passed	Tail 1000	Tail 1000 Passed
SIN3A	TAF1	104	0.654	111.644	Passed	44.0	Passed	35.0	Passed	0.019	Failed
SIN3A	TBP	878	0.61	105.196	Passed	47.0	Passed	39.0	Passed	0.003	Passed
TAF1	SIN3A	104	0.654	111.644	Passed	44.0	Passed	35.0	Passed	0.019	Failed
TAF1	TBP	1235	0.593	98.548	Passed	41.0	Passed	34.0	Passed	0.002	Passed
TBP	SIN3A	878	0.61	105.196	Passed	47.0	Passed	39.0	Passed	0.003	Passed
TBP	TAF1	1235	0.593	98.548	Passed	41.0	Passed	34.0	Passed	0.002	Passed

	SIN3A	TAF1	TBP
SIN3A	-1	3	4
TAF1	3	-1	4
TBP	4	4	-1

Figure 3: Screenshot of TICA results page, after submitting a query on cell line GM12878. Middle table report all features from statistical tests and deterministic filters. Blue squares in the heatmap denote higher number of tests passed.

3.1 Data preprocessing

We implement the preprocessing step of TICA by taking advantage of *GenoMetric Query Language* (GMQL), a high-level, declarative query language which supports data extraction as well as many standard data-driven computations required by tertiary data analysis *[7]*. We use mostly ChIP-seq datasets extracted from ENCODE, but GMQL supports an integrated repository with datasets extracted from ENCODE, TCGA, Epigenomic Roadmap, GDC, and GEO; integration of heterogeneous datasets is supported by the GDM data model *[8]*. In GDM, a dataset includes several samples; each sample is a pair of regions and metadata. For instance, in the case of a sample resulting from a ChIP-seq technology, regions describe the peaks of expressions (their start, stop, peak positions and score; region samples are similar to tracks that can be seen on a genome browser); metadata describe additional attributes of each sample, for instance the specific experiment name and tissue.

GMQL queries are written as sequence of statements operating on abstract variables, each representing a genomic dataset; it is a high-level language whose conditions apply both to regions and to metadata. GMQL implements most of the standard relational algebra operations *[2]*, such as SELECT, PROJECT, GROUP, ORDER, UNION, DIFFERENCE; it also supports domain-specific operations, such as genometric JOIN, MAP and COVER, whose semantics and implementation are defined in *[5]*[§].

GMQL is particularly powerful as a data extraction language, due to its implicit iteration over multiple samples of a dataset and its very compact and readable query specification. The language is also highly effective when integrating data coming from vastly different data sources, as the standardization to GDM allows for direct comparison between regions (represented by the same coordinates, such as chromosome, start and stop) while preserving all information ascribed to a particular data format (such as peak calling p-values from ChIP-seq experimental data, or rpkm values from RNA-seq). GMQL seamlessly combines these attributes using commands such as PROJECT and MAP, supporting and streamlining data analysis pipelines.

As an example of the above, we show the queries which are used for extracting TF binding sites (TFBSes) and transcription start sites (TSSes), relative to a given cell line, from the repository (Listing 1). The TFBS filtering query (lines 1 through 6, same Listing) is also performed on user-provided narrowpeaks.

```
1  # extracts 1−base exact TF peaks and produces one sample for each TF
2  TFS = SELECT(experiment_type == 'ChIP−seq' AND cell == 'target_cell')
       ENCODE_NARROWPEAK;
3  TF_PEAKS = PROJECT(region_update: left AS start + peak, right AS start +
       peak +1) TFS;
4  TF_PEAK = COVER(1,ANY; groupby: tf_name) TF_PEAKS;
5
```

[§] The full description of GMQL language for the latest version (2.1 at the time of writing) can be found at `http://www.bioinformatics.deib.polimi.it/geco/?try`.

```
 6 # extracts TFBSes by looking at enclosing windows with enough TF signal
       , i.e. enough peaks falling in a window of 1000 bases
 7 WINDOW = PROJECT(region_update: start AS start - 1000, stop AS stop +
       1000) TF_PEAK;
 8 MAPPED_WINDOW = MAP(joinby: tf_name) WINDOW TF_PEAK;
 9 TF_EXTRACTED = SELECT(region: count >= w) MAPPED_WINDOW;
10
11 # extract histone marks —— H3K9ac and H3K4me3 are found in promoter
       areas of actively transcribed TSSes. Similar queries are written
       for histones H3K4me1 (enhancers) and H3K36me3 (exons) - here
       omitted
12 HMS = SELECT((histone_name == 'H3K9ac' OR histone_name == 'H3K4me3')
      AND cell == 'target_cell') ENCODE_BROADPEAK;
13 HM = COVER(1,ANY) HMS;
14
15 # filter TSS with enough overlap with histone marks
16 TSS = SELECT(annotation_type == 'TSS') ENCODE_BED_ANNOTATION;
17 PROMOTER = PROJECT(region_update: start as start - 2000, stop as stop +
        200) TSS;
18 MAPPED_PROM = MAP() PROMOTER HM;
19 TSS_FILTERED = SELECT(region: count >= h) MAPPED_PROM;
20
21 # further filters TSS with enough overlap with TF—PEAKS - from
       arbitrary TF peaks
22 MERGED_PEAKS = MERGE() TF_PEAKS
23 MAPPED_TSS = MAP() TSS_FILTERED MERGED_PEAKS
24 TSS_EXTRACTED = SELECT(region: count >= k) MAPPED_TSS;
```

Listing 1: GMQL query used to filter TF binding sites and TSSes used by the method (summary).

- *Lines 2-4:* the TFS variable includes all the relevant TF samples extracted from ENCODE narrowpeak datasets[†]. The PROJECT operation is used to reduce the size of ChIP-seq regions to a single base pair. The COVER(1,ANY) operation is used to combine replicates from different transcription factors, keeping all regions from all samples and merging any two or more regions which overlap. The *groupby* option limits the merging to samples that share the same *tf_name* metadata attribute, i.e. contain experiment data on the same transcription factor. The result includes one sample for each distinct TF, with regions corresponding to a single base pair where the peak is located.

- *Lines 7-9:* Candidate TFs for the method are selected. A window of 1000 base pairs is constructed around each peak, and TFs associated with windows enclosing a counter of peaks over a threshold (w) are extracted. The PROJECT operation builds the WINDOW, the MAP operation counts the number of peaks included in each window, and the final SELECTion extracts the TFs.

[†] ENCODE narrowpeaks are also given for ChIP-seqs targeting histone modifications. We remove them from the dataset by means of NOT clauses - omitted for brevity.

According to the method, TSSes are extracted based on three progressively applied conditions: overlap with histone marks of promoters, of exons, and of enhancers; we only explain how to select TSSes by using histone marks of promoters, as the second and third extractions are very similar.

- *Lines 12-13:* Histone marks are selected. Extraction is done by means of a SELECTion; replicates are then combined using the COVER, keeping all regions from all samples and merging any two or more regions which overlap. Eventually, each HM sample includes all the regions of a given (set of) histone modifications present in ENCODE.

- *Lines 16-19:* TSSes are filtered. Promoter regions are built, and overlapping histone modification regions are counted; a TSS is selected if it is supported by a sufficient number of overlaps (one for each histone mark in the relevant regions). As promoter regions, we take standardised extensions of transcription start sites; these are built using a PROJECT, which takes TSSes and modifies their start and stop positions by extending them 2000 pairs upstream and 200 pairs downstream[‡]. Then, the MAP operation counts the number of overlapping regions and the final selection filters the TSSes.

- *Lines 22-24:* Finally, TSSes to be used in the method are extracted. In addition to overlaps with histone modifications, we also require TSSes to be supported by a sufficient number of TF peaks. The MERGE operation puts all the peaks of different transcription factors into a single sample, then the MAP counts how many peaks overlap with promoter regions for TSS as defined above; the final SELECT extracts the TSSes.

3.2 Interaction prediction method

After TF binding site data has been filtered and reduced to 1bp length by means of the GMQL queries, TICA investigates colocation between two sets of transcription binding sites in a statistically robust way. It does that by performing a significance test based on the null hypothesis that two random TFs (named *candidate interactors*) are not found in close position to one another (according to suitable aggregation functions, as below).

Briefly, the main concept behind TICA is the *minimal distance couple* (or *mindist couple* for short), a pair of intervals which are found to be the closest to one another according to the given coordinate system, and are not located too far apart. Minimal distance couples for a given pair of transcription factors (represented by the positions of their binding sites) induce a distance distribution via the genomic distance function, which is used to generate a set of observations related to that particular pair of TFs. TICA uses both standard (average,

[‡] These are nominal values for promoter and exon length, chosen for our experiments. Different investigators can use their own values for regulatory regions extension, depending on their biological assumptions.

median) and novel (median absolute deviation, distribution right tail size) statistical aggregators of the distances as features to feed a statistical classifier. The output of the classifier is whether the null hypothesis above is rejected for a certain TF-TF pair.

TICA builds null distributions for each feature by randomly sampling pairs of TFs from those available in ENCODE phase 2 and 3 datasets (narrowPeak format) in a given cell line. Data comes from chromatin immunoprecipitation and sequencing experiments on three major context cells: HepG2, GM12878, K562. For each cell line, we also extract the TSSes which are more likely to be actively transcribed, based on available histone marks (see Section 4.1) and TF binding information, which we use to impose additional restriction on the candidate interactors: TICA rejects a candidate pair if the ratio of couples which colocate in the same promoter is too low, with respect to the total size of the distribution. This is done to make sure that results have biological relevance as indicators of potential coregulatory behaviour, which is linked with physical interactions *[3]*.

We calculate p-values of null distributions and TFBS colocation in promoters using a Python script (v3.6). In particular, mindistance couples are computed first with respect to one of the TF (meaning, for each of its binding sites, the algorithm find the ones for the potential partner which are closest and not above the distance threshold), then with respect to the other. The two results are then intersected, yielding the final mindist couple list: this is done to avoid scenarios where one binding site is the closest with respect to a target, but the reverse is not true (Figure 4).

3.3 Data format

TICA can in principle work with any kind of genomic regions, due to the fact that data is managed by the flexible GDM model via GMQL. However, it is reasonable to assume that the required maximum displacement between candidates will be small (in other words, we expect regions to be very close to each other with respect to the linear dimension of the universe set): this is due to the fact that physical interaction between TFs happens at molecule scale, where distances are in the order of 1 to 10 nucleotide base pairs *[4]* (compare with the average size of a human chromosome, $1.2 \cdot 10^8$ base pairs).

Data from ChIP-seq experiments is given in variable size, usually in the range of 10^1 (point-source information or TSS locations) to 10^3 base pairs (histone modifications, genes), making certain fine-grained analysis much more difficult. We overcome this by using ENCODE narrowpeak regions, which contain the position of the highest confidence point-source for each region (as offset from the starting point): we represent each binding site using only this high-confidence, 1 base pair-long peak in order to make statistics on small values of distance meaningful.

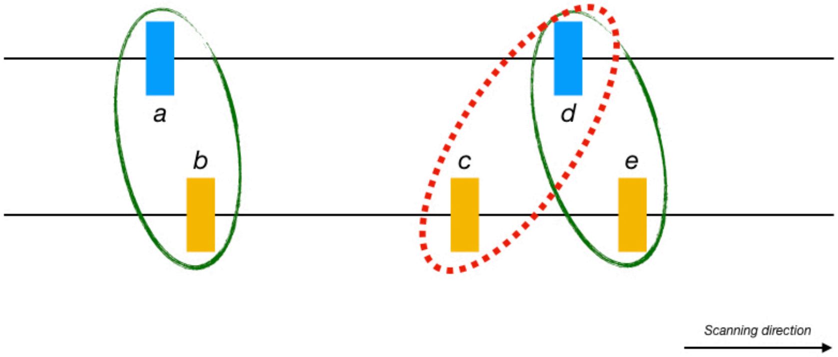


Figure 4: Example computation of mindistance couples, highlighting possible ambiguities. Two TF track snippets are given (blue and orange). Proceeding as per the scanning direction, if blue is chosen as anchor (and orange as experiment), the minimal distance couples are correctly identified as (a,b) and (d,e) (note that d is closer to e than to c). However, if roles are inverted, three couples will be found instead: (b,a), (c,d), (e,d). Intersecting results guarantees consistency with the model.

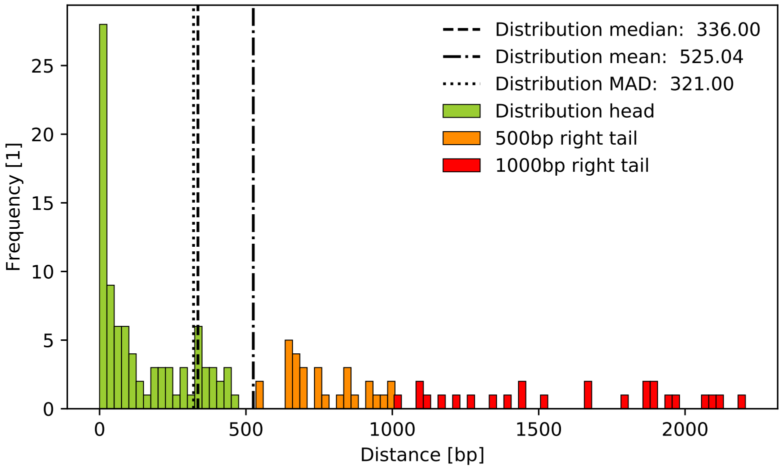


Figure 5: Distance distribution inferred from minimal distance couples of transcription factors CTCF and JUN in cell link HepG2. Vertical lines denote statistical aggregators used in TICA tests (mean, median and median absolute deviation). Two dimension for the right tail are given: long (distance greater than 500bp, orange) and short (distance greater than 1000bp, red). Right tail size in this case is approximately 15% of the total.

Cell line	TF number	File number	Data size [Gb]	Data size [Millions regions]	Actively transcribed TSSes number
HepG2	200	1085	13.16	181	25097
GM12878	148	794	8.66	121	31660
K562	288	2057	23.19	322	32356

Table 1: Data volume used in pipeline experiments, listed by cell. TSS numbers refer to sample size after GMQL filtering.

4 Performance

4.1 Materials

We test and validate our model using data from ENCODE phase 2 and 3 ChIP-seq experiments in narrowpeak format, currently available in GMQL public repositories. Our chosen model organism was *Homo Sapiens*. We use the following data in our experiments:

- *Context cell lines:* three cell lines were selected due to data availability and quality: *HepG2* (liver carcinoma), *K562* (myelogenous leukemia) and *GM12878* (healthy lymphoblastoids);

- *TF binding locations:* data representing transcription factor binding points (TFBSes) in narrowPeak format ***[6]***, due to higher peak precision and presence of point-source location information for each region;

- *Histone marks:* the following marks have been chosen for highlighting actively transcribed TSS (see Section 3): H3K36me3 (exons), H3K9ac and H3K4me3 (promoters), H3K4me1 (enhancers). Data from ENCODE phase 2 and 3 repository, limited to cell lines mentioned above. Data format chosen is ENCODE broadPeak ***[6]***;

- *Transcription start sites:* data also from ENCODE phase 2 experiments, in standard bed format. TSS are described in terms of the first exon base only (regions are 1bp in length).

Data quantities are listed in the Table 1.

4.2 Parameter settings

Parameter chosen for GMQL queries and TICA algorithm during performance evaluation are reported in Table 2. The choice of parameters is driven by the following biological considerations:

	Parameter	Value
Genomic dimensions(*)	Exon length	200bp
	Promoter length	2000bp
	Enhancer length	100kbp
Data filters	Clustering value k	3
	TFBS scanning window size	1000bp
	Min. number of TFBS in active promoters	50
Metric constraints	Mindist couple max distance	2200bp
Tests and thresholds	Number of points in nulls	$\geq$ 10000
	Right-tail threshold	1000
	Test p-value	0.2
	Required number of rejected null hypotheses	3
	Minimum number of mindist couples	1
	Minimum fraction of mindist couples colocating in a promoter	0.01

Table 2: Parameter setting for TF-TF interaction prediction pipeline. (*): extending TSS according to their strand.

- standardised regulatory region length is a common assumption when working with gene expression regulation;
- TFBS window of accumulation is chosen so that it covers most of a standard promoter size without overextending;
- mindist couple max distance is one promoter length plus one exon (assumed size of promoter area)
- the minimum number of TFBSes in active promoter is chosen as the first quartile of the overall distribution of the counts of TFBSes in promoters in HepG2 (taken as preferred modelling environment).

Experiments and performance evaluation have been performed on the GeCo server at DEIB, Politecnico of Milano. The TICA web server is hosted on a Dell PowerEdge R730xd server with 2 Intel Xeon E5-2660 v4 processors and 384 GB of RAM.

4.3 Performance assessment

Performance estimation for the web server can be divided in two blocks:

- computation time needed to (re)generate the database from ENCODE data and/or to analyse novel data;
- accuracy of predictions.

In the context of this work, we focus mostly on evaluation of the actual computation performance (i.e., time consumed) as opposed to discussing algorithm accuracy. Future works will be targeted towards the correctness of the method.

4.3.1 Null distribution generation from ENCODE

Execution times for the full pipeline on ENCODE data are listed in Table 3. Cell lines and data volumes correspond to those reported in Section 4.1. The pipeline has been split in four major parts:

- *TFBS query:* corresponding to lines 2 through 9 of Listing 1;
- *TSS query:* corresponding to lines 12 through 24 of the same;
- *TSS map:* the mapping of each binding site to all TSS in the promoter of which it binds, used to determine whether a mindist couples binds to shared promoter;
- *Mindist couples:* where the mindistance couples are computed by TICA.

Computation times reported in Table 3 refer the full analysis of the entire ENCODE cell line they refer to, which can involve thousands of millions of regions at a time (in the case of K562, ca. $3 \cdot 10^8$ regions are analysed - cf. Table 1). In typical use cases, the computation times are faster by two to three orders of magnitude (cf. next paragraph).

Cell line	TFBS query	TSS query	TSS map	Mindist couples
HepG2	108	194	21	120
GM12878	77	138	15.5	60
K562	204	407	46	376.5

Table 3: Tabulation of execution times for TICA pipeline steps on the three context cell lines. Input data is taken directly from ENCODE (see Table 1). Time measured in minutes.

4.3.2 Analysis of novel data

As a simulation of typical levels of workload, we generate synthetic data in narrowpeak format with variable levels of data volume. Two scaling factors were considered:

- number of transcription factors (each with a given number of regions): this influences the amount of candidates and therefore the number of times each step must be executed;
- sample size (in number of regions per sample, for a fixed amount of TFs): influences the amount of data filtered by TFBS queries, the mapping times and the number of comparisons during mindist couples' distance distribution creation.

Note that each TF contains only one sample: giving more for each TF would not influence the computation times in a tangible manner (the COVER operation would collapse them to a single one).

We time the execution of the full pipeline on seven different scenarios, using HepG2 as context cell line: results are reported in Table 4. The datasets are built as follows:

- we first consider a baseline scenario where the user provides data for 20 TFs, each containing 5000 regions of 100bp length - we estimate this to be a typical data size for user-submitted datasets;
- moving on the TF number scale, we submit one small (10 TFs), one medium (100 TFs) and one large (1000 TFs) dataset. Each dataset contains one sample per TF, and all samples contain 1000 regions (lines);
- moving on region-per-sample number scale, we submit three other datasets: small (10^3 regions), medium (10^4 regions) and large (10^5 regions). Each dataset contains 50 TFs and one sample per TF as before.

Note that each level (small, medium, large) increases the raw amount of data by a factor of 10, hence the increase in time is linear rather than exponential. To visualize this, we provide loglog plot of the scaling curves for TF- and sample

Cell line	TFBS query	TSS map	Mindist couples	Total
Baseline	34	12	3	0.8’
TF-small	11	5	0.5	0.5’
TF-medium	35	52	23	2’
TF-large	219	525	802	26’
SAMPLE-small	13	28	7	1’
SAMPLE-medium	111	33	23	3’
SAMPLE-large	613	41	38	12’

Table 4: Tabulation of execution times for TICA pipeline steps on synthethic datasets. Context cell line chosen is HepG2. Time measured in seconds except for total, which is converted to minutes for clarity.
.

size-scaling in Figure 6. Note that TSS query filter time has not been timed in this scenario, as TSSes are not recomputed when user data is uploaded.

Baseline scenario is successfully computed in approx. 1 minute, which is also the expected time for a typical user-provided dataset.

4.3.3 Accuracy

Briefly, we compare TICA predictions against existing biological knowledge, represented by two databases: CORUM *[10]*, a collection of experimentally verified mammalian protein complexes, and BioGRID *[11]*, which reports functional interactions between proteins based on both high-throughput datasets and individual focused studies. We consider an interaction to be supported by evidence if its two components are mentioned in a complex (CORUM) *or* as a protein-protein interaction (PPI, in BioGRID). The quality metrics that we use are *recall* (fraction of interactions correctly as positives out of all interaction supported by evidence), *specificity* (fraction of intersection not identified as positives out of all interactions which are not supported by evidence) and *geometric mean performance* (square root of the product between recall and specificity *[1]*). Results are tabulated in Table 5 for the largest cell line, K562.

A caveat is that not all TF-TF interactions correspond to complexes or PPIs (e.g. antagonistic TF-TF interactions), and not all complexes and PPIs correspond to TF-TF interactions. Nonetheless, co-operative TF-TF interactions are expected to be enriched in complexes and PPIs. This enrichment can be computed as recall over 1 minus specificity, which evaluates to 1.95 in our specific example. That is, a TF-TF pair that is predicted by TICA to interact is twice as likely to be found in a complex or as a PPI than a pair that is predicted not

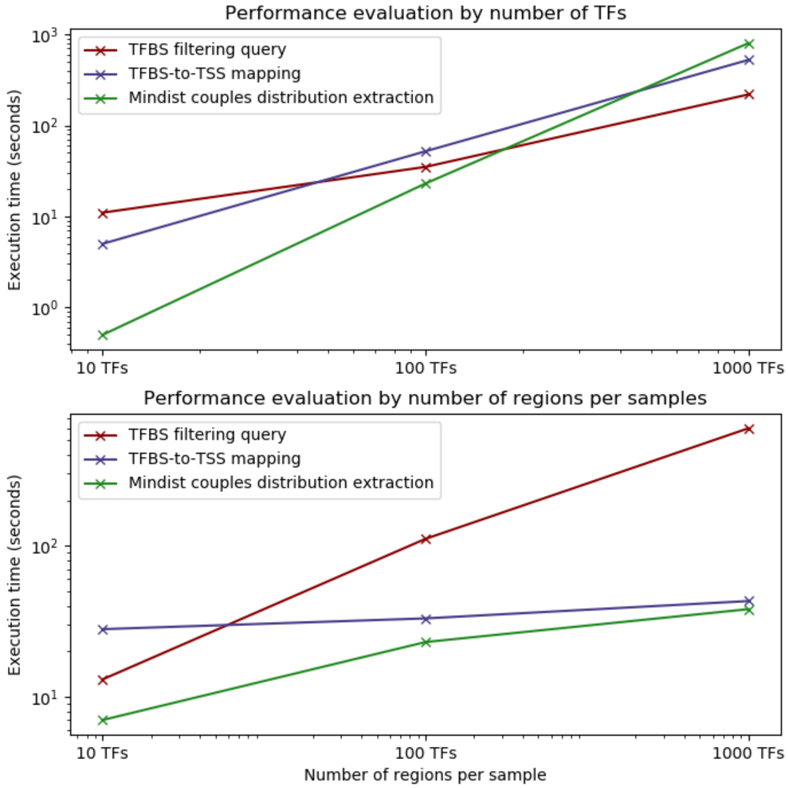


Figure 6: Loglog scale graph of execution time for TICA on ENCODE datasets. Each line corresponds to one of the three algorithm steps timed as per Table 4. *Upper:* scaling with respect to the number of TF in a datasets, with fixed number of regions per sample; *lower:* scaling with respect to number of region in a sample, with fixed number of TFs (and hence samples).

Cell line	Recall	Specificity	Geometric mean performance	Enrichment
K562	0.297	0.848	0.502	1.95

Table 5: Tabulation of quality measures for TICA predictions, with respect to the union of CORUM and BioGRID databases. Data from ENCODE cell line K562.

to interact.

5 Discussion

In this work, we introduce the TICA web server, a convenient tool for biologists to analyze ChIP-seq data on TF bindings for TF-TF interaction prediction. TICA leverages on GMQL, a novel language for data management, integration and querying of large, heterogeneous genomic datasets. Through the TICA web server, one can easily appreciate the expressive power and ease of use of the GMQL query language.

The TICA web server is a compact tool which nonetheless allows for fast analysis of entire cell lines from ENCODE ChIP-seq experiments: once data is generated (typically only after a major ENCODE release), running the prediction algorithm on repository data is computed in a short execution time. Updating the repositories with novel data has also very reasonable time requirements, considering that a repository's update rarely occurs (the cell line with the most data available, K562, takes about 16h from start to finish on the server specified in 4.1).

TICA scales very well with increasing data size provided by the user: as shown in Figure 6, it exhibits a linear or close to linear increase with respect to both the number of regions available in each samples, and the number of TFs (samples) in the user provided datasets. This gives us confidence in saying that TICA can be used as a component of larger pipelines in the investigation of TF-TF interactions.

When cross-checked with popular protein-protein interaction (PPI) and protein complex databases, TICA shows very good specificity (>=80%) while maintaining acceptable recall (circa 30%), considering that these reference datasets are currently incomplete. Given these quality measures, TICA can be used both as an effective screening tool in preparation for wet-lab experiments, and as direct computational tool for investigating the interaction between novel transcription factors and/or experiments in specific conditions, such as disease or different cell lines.

Thanks to the expressive power of GMQL, the user is not required to preprocess data or convert it to any particular schema: peaks called in the standard narrowpeak format are sufficient to perform analysis, and are reduced to their point-source form directly by the query tool. Also, the TICA web server supports a high level of customization, allowing investigator to tune almost every parameter of the prediction algorithm without any loss of performance with respect to what has been mentioned above. In conclusion, we suggest the TICA web server as a compact, reliable and efficient tool for tackling the TF-TF interaction prediction problem.

Acknowledgements

This work was supported by the ERC Advanced Grant *GeCo* (Data-Driven Genomic Computing) (Grant No. 693174) awarded to Prof. Stefano Ceri. We would like to thank members of the GeCo project for helpful insights. Prof. Limsoon Wong was supported in part by a Kwan-Im-Thong-Hood-Cho-Temple chair professorship.

References

[1] Batuwita R, Palade V (2012) Adjusted geometric-mean: a novel performance measure for imbalanced bioinformatics datasets learning. J Bioinform Comput Biol 10(04):1250,003

[2] Codd EF (1970) A relational model of data for large shared data banks. Commun ACM 13(6):377–387

[3] Geisel N, Gerland U (2011) Physical limits on cooperative protein-dna binding and the kinetics of combinatorial transcription regulation. Biophys J 101(7):1569–1579

[4] Jankowski A, Szczurek E, Jauch R, Tiuryn J, Prabhakar S (2013) Comprehensive prediction in 78 human cell lines reveals rigidity and compactness of transcription factor dimers. Genome Res 23(8):1307–1318

[5] Kaitoua A, Pinoli P, Bertoni M, Ceri S (2017) Framework for supporting genomic operations. IEEE Trans Comput 66(3):443–457

[6] Landt SG, Marinov GK, Kundaje A, Kheradpour P, Pauli F, Batzoglou S, Bernstein BE, Bickel P, Brown JB, Cayting P, et al (2012) Chip-seq guidelines and practices of the encode and modencode consortia. Genome Res 22(9):1813–1831

[7] Masseroli M, Pinoli P, Venco F, Kaitoua A, Jalili V, Palluzzi F, Muller H, Ceri S (2015) Genometric query language: a novel approach to large-scale genomic data management. Bioinformatics 31(12):1881–1888

[8] Masseroli M, Kaitoua A, Pinoli P, Ceri S (2016) Modeling and interoperability of heterogeneous genomic big data for integrative processing and querying. Methods 111:3–11

[9] Nanni L (2017) A python data analysis library for genomics and its application to biology. Master's thesis, Politecnico di Milano - DEIB, available at `https://www.politesi.polimi.it/handle/10589/135989`.

[10] Ruepp A, Waegele B, Lechner M, Brauner B, Dunger-Kaltenbach I, Fobo G, Frishman G, Montrone C, Mewes HW (2009) Corum: the comprehensive resource of mammalian protein complexes—2009. Nucleic Acids Res 38(suppl_1):D497–D501

[11] Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M (2006) Biogrid: a general repository for interaction datasets. Nucleic Acids Res 34(suppl_1):D535–D539