Improving Predictive Accuracy in Enhancer Identification: ANALYSIS OF NEXT-GENERATION SEQUENCING DATA TO PREDICT ENHANCERS

Analysis of Next-Generation Sequencing Data to Predict Enhancers

Deniz Muratlı, 29480

9 July 2023 – 8 September 2023

Max Planck Institute for Molecular Genetics, Department of Computational Molecular Biology, Vingron Lab

Dr. Stefan Haas

1 October 2023

Sabanci University

Faculty of Engineering and Natural Sciences

Molecular Biology, Genetics, and Bioengineering & Computer Science and Engineering

Abstract

Enhancers are important regulatory sequences influencing transcriptional activity. Their identification and characterisation are complicated by their varying sizes, locations, actions across different tissues and conditions, and the lack of common sequences. While traditional methods of identifying enhancers have significant limitations, inventions such as ChIP-seq, ATAC-seq, and DNase-seq have revolutionised enhancer discovery. One modern tool that benefits from these technologies is CRUP, which predicts enhancers based on ChIP-seq data for histone modifications accepted to be indicative of enhancer function. However, its predictions are not perfect and suffer from shortcomings especially in regards to determining enhancer boundaries and revealing state-specific enhancers. In this paper, a suite of tools developed by the intern for Vingron Lab at Max Planck Institute for Molecular Genetics to adjust CRUP’s predictions to better reflect underlying raw data and to extract state-specific enhancers was described in detail. GREAT, a tool that predicts gene enrichment from cis-regulatory sequences, was used to successfully demonstrate the effectiveness of the developed software.

Table of Contents

1 Introduction………………………………………………………………………………………...4

1.1 CRUP……………………………………………………………………………………………..5

1.2 GREAT…………………………………………………………………………………………...7

2 Institution Background……………………………………………………………………………..8

3 Internship Project: Descriptions and Analysis……………………………………………………...9

3.1 Overarching Framework………………………………………………………………………….9

3.2 Problems………………………………………………………………………………………….9

3.3 Methods and Materials………………………………………………………………………….11

3.4 Results…………………………………………………………………………………………..17

4 Internship Experience: Observations and Analysis……………………………………………….21

5 Conclusion………………………………………………………………………………………...22

6 Recommendations………………………………………………………………………………...23

7 References………………………………………………………………………………………...23

8 Appendixes………………………………………………………………………………………..26

8.1 Appendix A……………………………………………………………………………………...26

Introduction

Enhancers are regulatory sequences that enhance the transcription of an associated gene when bound by transcription factors. An enhancer sequence may be located either upstream or downstream of the linked gene and may be as much as a million base pairs away from the gene. What matters is its physical proximity to the gene in the folded and coiled state of DNA (although it must be on the same chromosome as the gene), which, through bound transcription factors, allows it to lead the RNA polymerase to the promoter (Nature, n.d.; Panigrahi & O’Malley, 2021, p. 2).

Understanding enhancers’ key characteristics, mechanisms of action, and roles in evolution and disease is important, as the human genome [among others] is thought to contain hundreds of thousands of enhancers. However, it is a challenging task: they are scattered across 98% of the non-coding section of the human genome, their locations are highly variable, they can act on non-neighbouring genes, they can regulate multiple genes, their general sequence code is poorly understood (assuming that one exists at all), and their activity is often restricted to a specific tissue, cell type, a time point, and/or physiological/pathological/environmental condition. Contributing to complex gene expression patterns, these all complicate predicting enhancers *in silico* (Pennacchio et al., 2013, pp. 1-2).

Historically, transgenic reporter assays, functional assays, enhancer trapping, reporter gene arrays, comparative genomics (seeking enhancers in regions conserved across species), and methylation and DNase I hypersensitivity assays were used to identify enhancers (Kvon, 2015; Zhao et al., 2020; Wu et al., 2003; Visel et al., 2007; Inoue & Ahituv, 2015; Maricque et al., 2016). Although some of these methods are still used to identify enhancers, especially in certain contexts, they all have shortcomings. For instance, in many cases, transgenic assays cannot demonstrate functions for putative enhancers in conserved regions, either because the assay is insufficient to test for the specific function or because the conserved sequence is not necessarily an enhancer. Moreover, countless enhancers are species-specific, displaying little to no conservation across species (Pennacchio et al., 2013, p. 2). Predicting transcription factor affinity to putative enhancer sequences based on a biophysical model and using that information to determine enhancers was proposed as a more promising approach but left much to be desired (Roider et al., 2006, pp. 1-2; Manke et al., 2008, pp. 1-2).

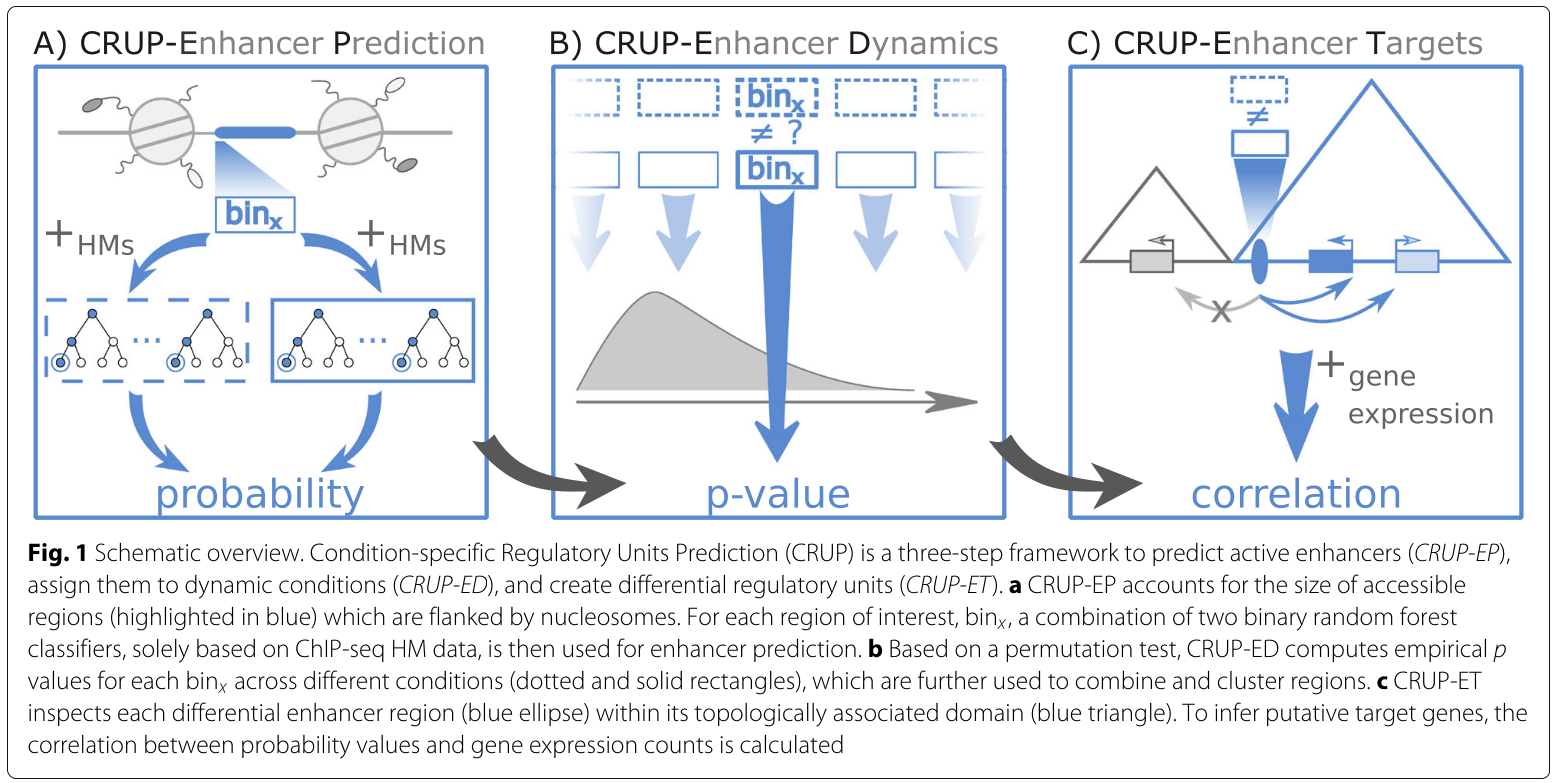
The advent of ChIP-seq revolutionised approaches to enhancer identification, enabling sequencing regions with epigenetic marks indicative of enhancer function. It is independent of conservation and easily applicable to specific tissues, cell types, time points, and conditions – suitable for dealing with diverse enhancers. The epigenetic marks used for this purpose are usually p300, histone H3 acetylated at lysine 27 (H3K27ac), and H3 monomethylated at K4 (H3K4me1) (Visel et al., 2009, p. 1). Additionally, other new high-throughput sequencing techniques such as ATAC-seq and DNase-seq allow mapping DNase I hypersensitive sites, offering another powerful way of identifying enhancers (Pennachio et al., 2013, p. 3). More recently, attempts are being made to detect enhancers via deep learning, removing human biases from the process (Tang, 2022).

Linking identified enhancers to their target genes, which is critical for studying the effects of enhancer activity, presents another challenge. Despite being very much an unsolved question, various strategies exist to deal with the problem. A popular strategy is making use of Hi-C or other 3C-based approaches to identify TADs (topologically associated domains), domains of DNA where interactions within are more common than interactions between (sections of DNA that are clustered together in 3D space), and further using gene expression data to find possible links between genes and enhancers within the TADs. However, this is not only a costly approach, but sometimes also infeasible – high-resolution data that is often required to view an enhancer with the corresponding gene in the same TAD is usually not available (Yao et al., 2015, p. 5). Therefore, more intuitive but still fairly reliable approaches are often preferred – some used by GREAT are described below.

*CRUP*

CRUP (Condition-specific Regulatory Units Prediction) is a tool developed by Vingron Lab at MPIMG that relies on ChIP-seq data for the aforementioned epigenetic marks on the tissue under preferred conditions to predict tissue and condition-specific enhancers. Its workflow consists of 3 consecutive steps: CRUP-EP for the prediction of active enhancer elements, CRUP-ED for combining that with condition-specific enhancer dynamics, and CRUP-ET for the identification concurrently changing enhancer-target pairs (Ramisch et al., 2019, p. 2).

CRUP-EP is an enhancer classifier that takes a feature set of histone modifications H3K4me1, H3K4me3, and H3K27ac as input and uses two random forests to distinguish first active regulatory regions from the rest of the genome, and then enhancers from active promoters, producing enhancer probabilities for each bin of size 100 [base pairs] and enhancer region predictions based on those probabilities. CRUP-ED considers genome-wide enhancer predictions made by CRUP-EP for multiple conditions (e.g. different developmental states). It applies a permutation test on the predicted bin-wise enhancer probabilities and uses the resulting pairwise empirical p-values to combine differential bins and cluster them into dynamically changing enhancers. Finally, CRUP-ET pairs regions predicted by CRUP-ED to target genes by correlating enhancer probabilities and gene expression values under the same conditions within the same TAD (established by Hi-C sequencing) (Ramisch et al., 2019, p. 3). *Figure 1* presents a schematic overview of the workflow (Ramisch et al., 2019, p. 4).

Figure 1: Overview of CRUP.

*GREAT*

GREAT (Genomic Regions Enrichment of Annotations Tool) is a tool developed by Bejerano Lab at Stanford University to predict functions of cis-regulatory regions. It takes genomic regions (e.g. regions with TF-binding detected by ChIP-seq) as input, associates the regions with their putative target genes, get the GO annotations of those genes, and find whether any terms are enriched with the hypergeometric and binomial tests, outputting significant terms. Based on a website, the tool offers only two versions each of the mouse and human genomes for GO annotations. As such, test regions of any other genome has to be “lifted over” to either of the two genomes based on synteny (not a built-in option) if they are going to be provided as input. Background regions used can be restricted by the user if desired (McLean et al., 2010; Tanigawa et al., 2022).

For associating genomic regions with target genes, GREAT offers 3 user-modifiable options that are all less sophisticated than CRUP-ET. The default option is “basal + extension,” where each gene is designated a basal regulatory domain of a minimum distance upstream and downstream of the transcription start site (default: 5 kb upstream, 1 kb downstream), which can overlap with the basal domains of other genes. Then, each regulatory domain is extended in both directions to the nearest gene’s basal domain, as long as it is within the maximum distance set by the user (default: 1000 kb). If an inputted genomic region is anywhere within the domain of a gene, then that gene is considered to be the target for that genomic region. Other two options are similar to each other: one takes the nearest gene [to each input region] as the target gene, whereas the other takes the two nearest genes as target genes, as long as they are within the user-specified distance (default: 1000 kb) (McLean et al., 2010; Tanigawa et al., 2022).

Institution Background

The Max Planck Society for the Advancement of Science (German: Max-Planck-Gesellschaft zur Förderung der Wissenschaften) is a non-profit research organisation based in Munich, Germany. Consisting of 80 institutes all around Germany, 5 institutes and a research facility abroad, it conducts basic research in natural sciences, life sciences, social sciences, and the humanities, with a particular focus on demanding research (in terms of funding or time). Named after its first president, Max Planck, it was founded in 1948, succeeding the Kaiser Wilhelm Society, which was founded in 1911. It employs 23,950 staff members: 20,898 are contractually employed, 519 are scholarship holders, and 2,533 are guest scientists. Being among the most prestigious research institutions in the world, it has been home to 30 Nobel Laureates, including world-renowned names such as Fritz Haber and Albert Einstein, and most recently, in 2022, Svante Pääbo (Max Planck Society, n. d.; Max Planck Society, n. d.; Max Planck Society, n. d.; Max Planck Society, 2022; Max Planck Society, 2023). A diagram of its organisational structure can be found in *Figure 2* (Max Planck Society, 2023)*.*

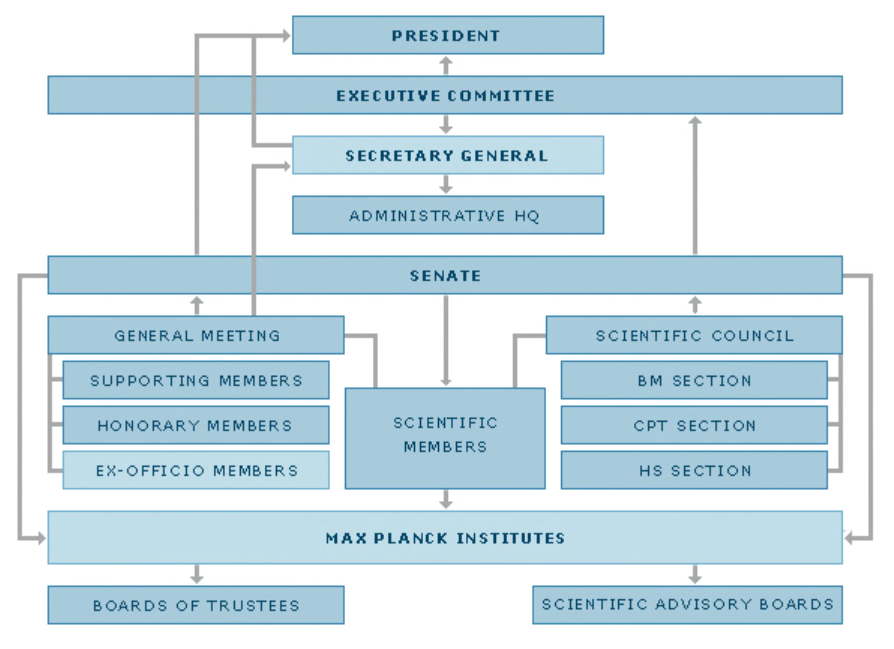


Figure 2: Governing bodies of the Max Planck Society.

The Max Planck Institute for Molecular Genetics is part of the Max Planck Society and is located in Dahlem, Berlin. It was founded in 1964 and currently employs more than 300 employees in over 20 research teams, service groups, technical services, and administration (Max Planck Institute for Molecular Genetics, 2023). Among its departments is the Department of Computational Molecular Biology, which contains 4 labs with distinct research fields. I completed my internship within this department, in Vingron Lab, which studies transcription factor-based and epigenetic transcriptional regulation as well as the regulatory effects of mutations (Max Planck Institute for Molecular Genetics, n. d.). I worked with Dr. Haas, who is currently interested in enhancers, implementing improvements to software previously developed by the lab and demonstrating the results.

Internship Project: Description and Analysis

*Overarching Framework*

One of Dr. Haas' current research topics is why bats are the only mammals capable of flight. Since bats do not have entirely unique genes, their capability to fly is thought to be stemming from a unique gene expression pattern rather than a mysterious “flight gene” (Haas, personal communication, n. d.). It is Dr. Haas’ hypothesis that this distinct gene expression pattern is the result of enhancer combinations not present in other mammals. Inquiring this hypothesis was the ultimate goal of the internship work, although, as an intern, I was tasked with dealing with rather technical aspects of the research.

*Problems*

In concrete terms, the research required to first find enhancers active in bats’ forelimbs but not hindlimbs, thus spotting enhancers that serve a specific function in the wings, and then figuring out whether those enhancers affect the transcription of forelimb-related genes. The motivation was that if genes that influence the development of forelimbs are enriched due to the activity of enhancers, that could be the reason why bats develop wings.

Even though the first task appears to be trivial with CRUP, the fact is that CRUP is far from ideal. The main issue dealt with throughout the internship project was “correcting” the way CRUP (specifically CRUP-EP) defined enhancer regions. To define an enhancer, the native algorithm selects a central bin with a strong signal and extends 5 bins to the right and 5 bins to the left as long as the signal does not diminish too much. If the conditions are not met, no enhancer is defined; if they are met, a 1100 bp region is returned as an enhancer. Besides the fact that strong but slightly intermittent signals can interfere with the predictions, setting the size of every enhancer in stone at 1100 bp is a questionable choice. While it has some scientific merit on the grounds that most enhancers tend to be around 1100 bp long, this strategy results in odd enhancers (Haas, personal communication, n. d.). Outliers where strong signals persist for thousands of base pairs are sliced up into countless mini-enhancers, and outliers where signals are observed for much shorter lengths are falsely assigned into longer regions.

CRUP’s strategy for assigning scores to enhancer regions is also potentially misrepresentative. Instead of using a metric that takes the scores of multiple bins into account, the highest bin score in an enhancer region is designated as the score for the whole region. Although the method of assigning scores was suboptimal, this issue was not dealt with since it involved changing an integral part of CRUP; no modifications to CRUP itself were made during the internship. Furthermore, the alternatives were either computationally intensive or would have significantly reduced the score’s accuracy – for instance, taking the average score instead of the maximum would make all scores seem similar. As a result, the existing scoring method was continued to be used, with an acknowledgement of its limitations.

Lastly, CRUP-ED and CRUP-ET, which are meant to find differentially active enhancers in a tissue and associate them with target genes respectively, appear to produce “faulty” results (no further explanations were provided by Dr. Haas), so the version of CRUP used for the project was tailored specifically by Dr. Haas to include only CRUP-EP. Even then, super-enhancers (enhancer clusters) designated by CRUP-EP were not considered at any step of the project, as only single enhancers were relevant for the research topic.

*Methods and Materials*

Tools used for the project were CRUP, GREAT, bedtools, a Perl script for lifting over genomic regions of other species to the human genome based on synteny, and various Python and Bash scripts written by me. The main data worked on was produced beforehand by MPIMG: it was ChIP-seq data for histone modifications indicative of enhancer function in bats’ forelimbs (FL) and hindlimbs (HL) – the scientific name of the species was *Carollia perspicillata*. It was supplemented by the annotated genome of the same species, incorporated into a genome browser personalised by Dr. Haas. The genome browser was based on an internal server of the institute and could take ChIP-seq-generated bins as well as CRUP-generated and/or modified enhancer predictions as input. It was used frequently to verify individual enhancers after each change to the Python scripts that modified enhancer boundaries, with the goal to avoid errors or misjudgments in the process. In the last few days of the project, ChIP-seq data (with the same histone modifications as before targeted) for 6 human tissues (pancreas, liver, lung, thymus, hippocampus, and kidney) in use by one of the PhD students of Dr. Haas was also used to ensure the robustness of the scripts. All scripts were modified several times during the course of the internship due to bugs, miscalculated consequences, and feedback from Dr. Haas – only the final versions are described below. Scripts written and used for the preliminary examination of data are not included.

A smoothing script (smoothen\_bins.py) was written to spread out scores across bins prior to adjusting enhancer boundaries. The goal was avoiding sudden drops in score and ensuring that enhancer predictions are consistent with ChIP-seq reads, not affected by random variation or false signals. A rolling mean approach was used, where each bin except a few at the beginning and at the end (the exact number depends on the window size) is assigned the average of the scores of bins within the rolling window. The rolling window size is a user-configurable option that is set to 5 bins by default.

A versatile recomputation script (recompute\_scores.py) was written to recalculate region scores at any step where it was necessary (e.g. recalculating the enhancer scores of CRUP after smoothing the underlying bins). It takes one or more bins and genomic regions whose scores will be recomputed as input. Each region’s score is calculated as the maximum score of bins covering that region in every inputted bin file. Each score is added to a new column, which is automatically given a fitting name in the header. Naming is based on tissue names, obtained by the script through parsing the part of the corresponding file name before the first “naming symbol” (e.g. FL\_bins.bed will yield FL) – a user-specified separator, “\_” by default.

The main script (adjust\_regions.py) takes genomic regions (in this case enhancers) and scored bins as input and adjusts them based on user-specified cut-offs. Three adjustment operations are available: 1) the regions are shrunk as long as there are bins on the edges with scores below the shrinkage cut-off, 2) the regions are extended as long as adjacent bins have scores higher than or equal to the extension cut-off, and 3) the regions are first shrunk then extended with the same criteria. Cut-offs for both shrinkage and extension can be defined either in absolute terms (e.g. 0.3) or in relation to the highest score within the respective region (e.g. /3 means the cut-off will be the highest score divided by 3). By default, the cut-off is 0.5 for extension and /2 for shrinkage; the regions are conservatively shrunk and extended, which presumes that predictions of CRUP reflect the reality for the most part. An optional, post-adjustment setting is merging, where two regions are merged if the distance between them is shorter than or equal to the user-specified margin – if a region is shorter than or equal to the margin in size and yet not surrounded by regions that can be merged, it is simply removed. The motivation was avoiding enhancers sliced into bits and pieces by CRUP.

Even though adjust\_regions.py yielded enhancers that better fit raw ChIP-seq reads and better represent the ground truth as will be discussed in *Results*, only adjusting enhancers was not enough to move on to inputting regions to GREAT. Both of the processed tissues were expected to have active enhancers that do not serve tissue-specific functions, which would influence the enrichment analysis. Therefore, a second script (project\_regions.py) was written that would project the given genomic region lists (labelled with the tissue names) over each other, thus returning a list of genomic regions labelled with the tissues they are active in. This allows easily parsing tissue-specific or shared regions according to preference. For the research topic in question, it allowed filtering out non-FL enhancers.

However, as expected, projections resulted in frequent, very short regions specific to one tissue or another. This is a natural consequence of overlapping regions, and would almost definitely increase further with a higher number of tissues – for example, consider projecting regions 1000-2000 HL, 1000-1400 FL, and 1500-2000 HL,FL. The short regions could interfere with GREAT’s results by causing the wrong genes to be hit. Moreover, they were likely to be the products of false signals or could be ignored in any case due to their negligible sizes. So, a strategy was necessary to deal with them. As such, a projection merging feature was implemented in a separate script (merge\_projections.py). The script takes the projected genomic regions and the desired margin (100 by default) as an input and outputs the list with no regions shorter than or equal to the margin in size. Regions that meet the criterion for removal are handled differently depending on the case. The script requires that all regions are scored for all tissues involved with recompute\_scores.py.

The input is processed in contiguous blocks. In each block, the shortest region shorter than or equal to the margin is repeatedly found and processed until all regions within the block are exhausted. The process can be broken down as follows, where “previous” and “next” imply adjacency to the “current” region on the same chromosome:

1. If all tissue labels of the current region are present in both the previous and the next region:
   1. If all tissue labels of the previous region are present in the next region:
      * The previous, current, and next regions are merged; the score is updated as the maximum of the three.
   2. If all tissue labels of the previous region are not present in the next region:
      * The current region is split in half, and one half is assigned to the previous region whereas the other is assigned to the next; the scores of both regions are updated as the maximum of the region in question and the current region.
2. If all tissue labels of the current region are only present in the previous region:
   * The current region is merged with the previous region; the score is updated as the maximum of the two.
3. If all tissue labels of the current region are only present in the next region:
   * The current region is merged with the next region; the score is updated as the maximum of the two.
4. If all tissue labels of the current region are present in neither the previous nor the next region:
   1. If any tissue label of the current region is present in both the previous and the next region:
      1. If merging the current region with the previous results in the highest score for any tissue:
         * The current region is merged with the previous region; the score is updated as the maximum of the two, and the tissue labels are updated as those of the longest of the two.
      2. If merging the current region with the next results in the highest score for any tissue:
         * The current region is merged with the next region; the score is updated as the maximum of the two, and the tissue labels are updated as those of the longest of the two .
      3. If merging the current region with the previous results in the same highest score as that with the next:
         * The current region is split in half, and one half is assigned to the previous region whereas the other is assigned to the next; the scores of both regions are updated as the maximum of the region in question and the current region, and the tissue labels of both are updated as those of the longest of the relevant two.
   2. If any tissue label of the current region is only present in the previous region:
      * The current region is merged with the previous region; the score is updated as the maximum of the two, and the tissue labels are updated as those of the longest of the two.
   3. If any tissue label of the current region is only present in the next region:
      * The current region is merged with the next region; the score is updated as the maximum of the two, and the tissue labels are updated as those of the longest of the two.
   4. If none of the tissue labels in the current region are present in both the previous and the next region:
      * Remove the current region as the last resort.

Since project\_regions.py and merge\_projections.py need to have certain conditions met for their inputs and arranging that can be cumbersome, a wrapper script (project\_regions.sh) for the two was written. Given two or more lists of genomic regions, the script labels all regions in both lists based on the file name properly using the provided naming symbol, concatenates the lists, and sorts them. Then, it can either 1) output the resulting list so that it can be manually inputted to project\_regions.py, 2) use the resulting list to run project\_regions.py and output the result, or 3) use the resulting list to run project\_regions.py, recompute the scores of regions in the output, and then run merge\_projections.py with the user-specified margin and bin files to return the final output.

Finally, a specificity score calculator (calc\_specificity.py) was designed to allow further specifying enhancers that would be inputted to GREAT. The script takes a merged projection of genomic regions and one tissue name as input and then calculates a specificity score for each region. The specificity score is defined as the average of differences between the score of the region for the tissue in question and that for all others. It can take values between -1 and 1, and higher scores indicate higher specificity.

After running all the aforementioned scripts on both FL and HL data with the margin set to 100 for adjustment (shrinkage and extension with the default cut-offs), and set again at 100 for merging projections, the final output’s specificity score distribution was examined. It was seen that most regions labelled only with FL had specificity scores above 0.2. Therefore, regions with scores higher than or equal to 0.2 were filtered into a separate file and then inputted to GREAT after being lifted over to the human genome via the Perl script written by Dr. Haas. The selected association rule was “basal + extension,” with the minimum distance set to 5 kb both upstream and downstream, and the maximum distance set to 100 kb.

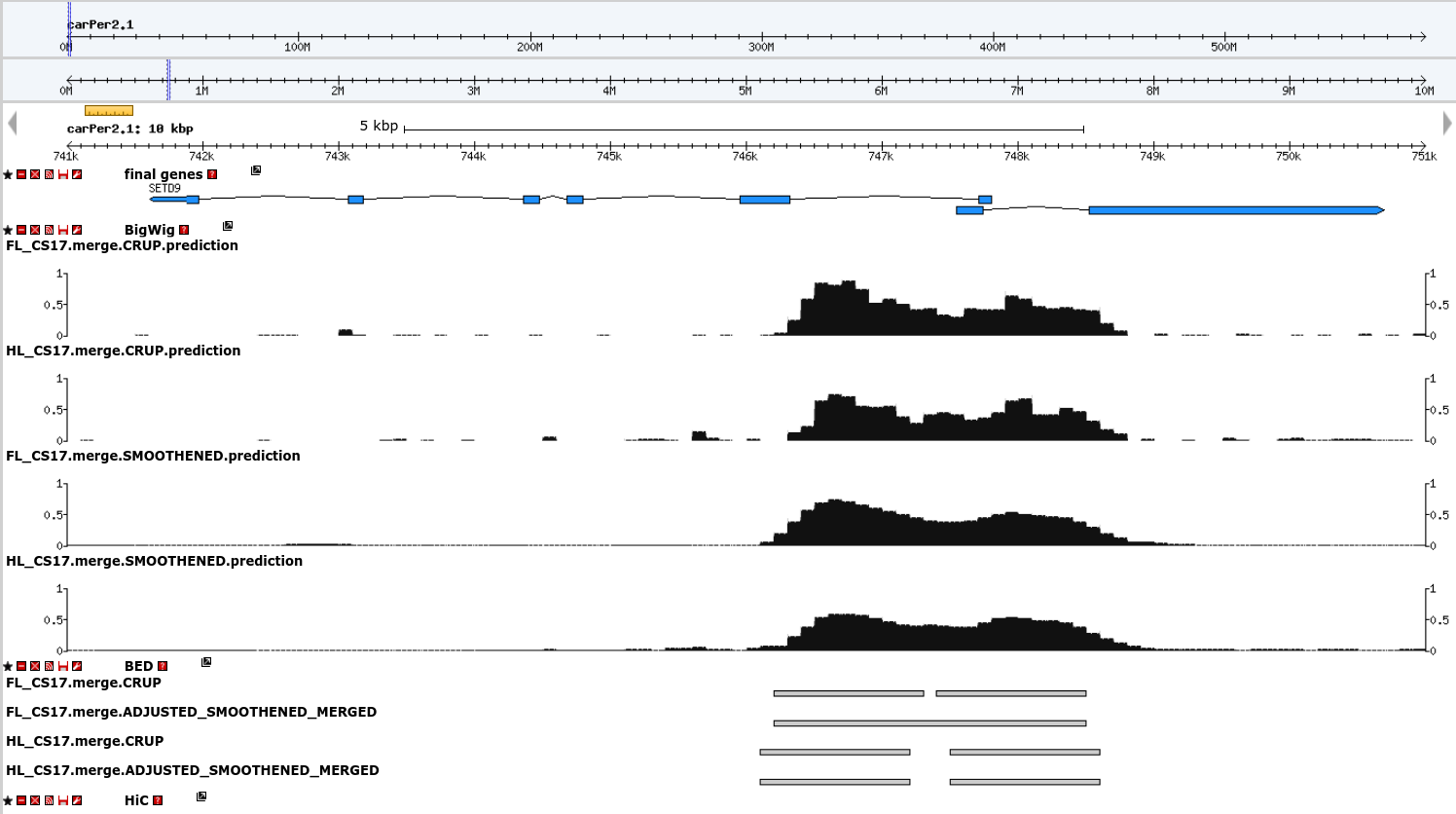
All code can be found on the Github repository provided in **Appendix A**. All inputs and outputs obey the .BED format and were constantly sorted first by the start coordinate and secondly by chromosome name and then merged with bedtools to avoid erroneous results. Each script is equipped with a --help option that describes how the script is used.

*Results*

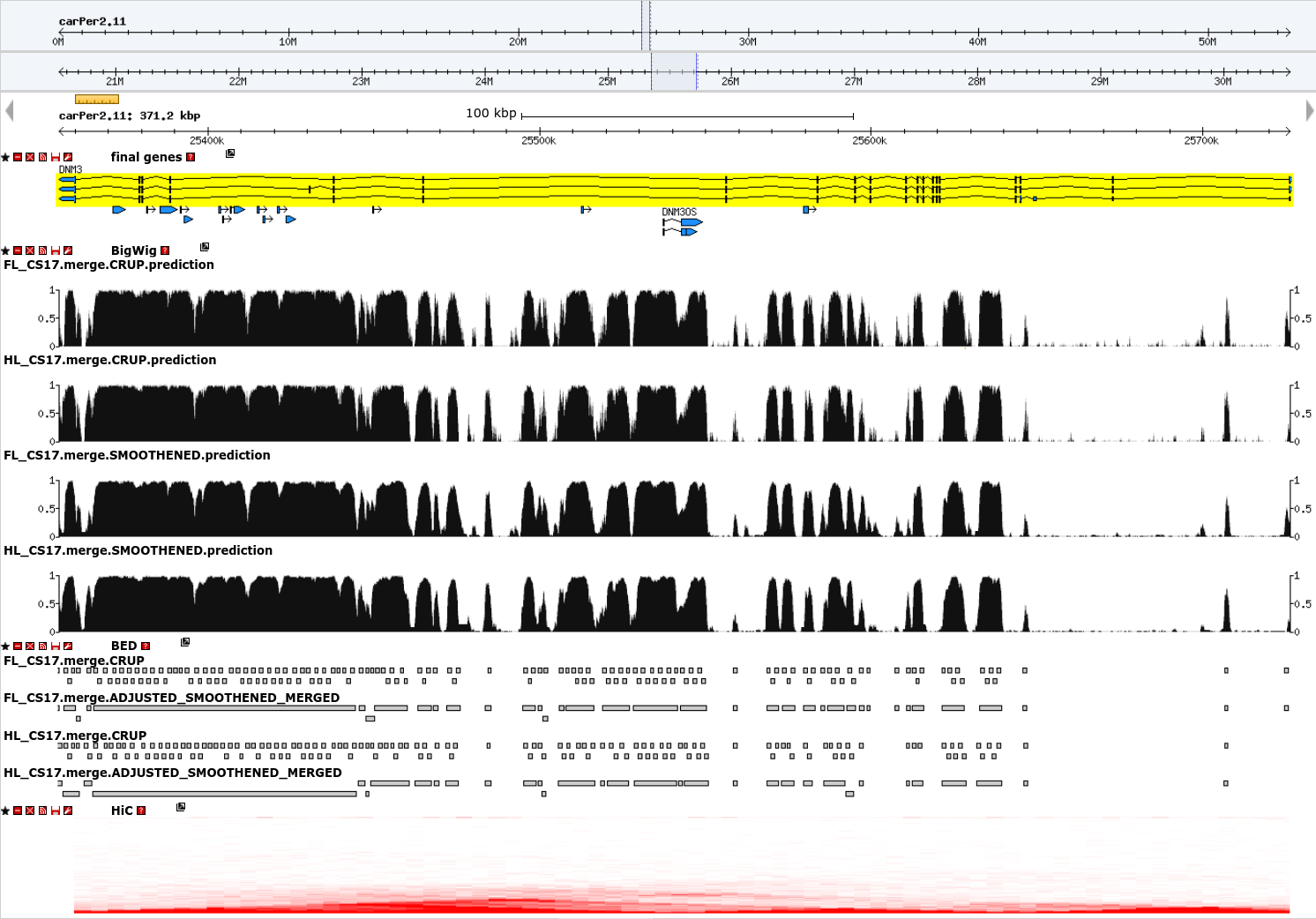
The task given by Dr. Haas for the internship was developing software to improve CRUP’s enhancer predictions. Such software, as detailed in *Methods and Materials*,was developed, thoroughly troubleshot, improved with additional functionality, and organised neatly in one of the institute’s clusters for future use by Vingron Lab.

The improvement in predictive accuracy after smoothing and adjustment was easily noticeable. *Figure 3*, *Figure 4*, and *Figure 5* display enhancer predictions before and after processing, alongside the corresponding bins before and after smoothing. The screenshots were taken from the genome browser mentioned earlier, and each loaded file was clearly labelled to increase comprehensibility.

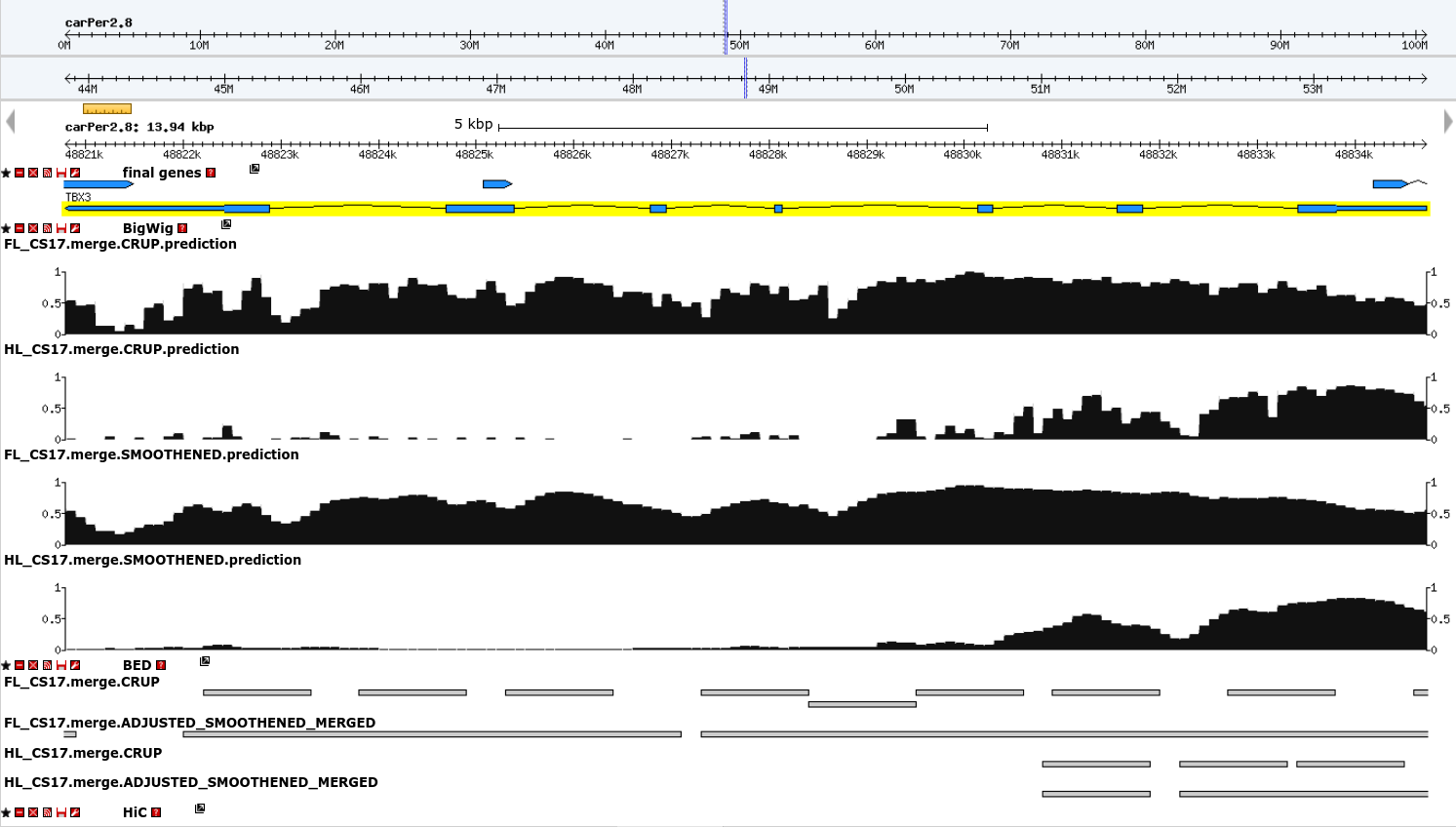
Enhancers in *Figure 3* were chosen randomly. It seems evident that although smoothing and adjustment did not affect CRUP’s predictions in HL, it was effective to merge what CRUP perceived to be separate enhancers in FL. Thus, the prediction was made consistent with the corresponding bins, which to the human eye appears as only one prolonged enhancer.

Figure 3: Enhancers near SETD9.

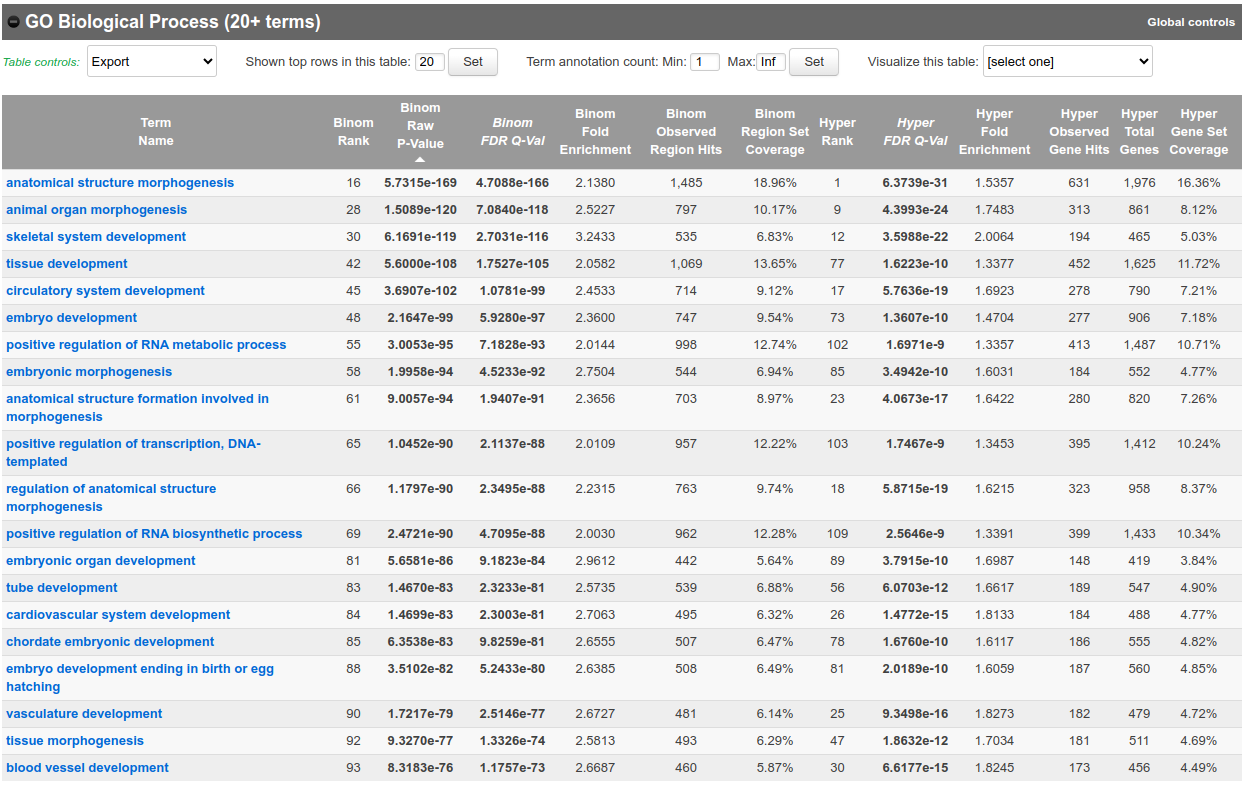
Enhancers found within genes are believed to be more crucial for enhancer function. They co-evolve with the gene they enhance, and are thus less likely to mutate – suggesting a stronger selective pressure favouring their preservation. However, it is often difficult to reliably spot them with a ChIP-seq experiment. What seem to be enhancers within genes may just be false signals resulting from the gene being open to transcription, which leads to histone modifications similar to those in active enhancers (Haas, personal communication, n. d.). Nevertheless, enhancer predictions within genes provide valuable test cases for evaluating the effectiveness of the developed software, particularly because these enhancers often exhibit longer sequences than usual. *Figure 4* and *Figure 5* are such cases. DNM3 in *Figure 4* is known to have a biased expression in the brain and likely not serve a limb-specific function, but it is special due to its tremendous length of 371.2 kb (NCBI, n .d.). CRUP predicts countless several, nearly adjacent enhancers within this range, even though the bins point to a different picture. The scripts have effectively merged the vast majority of these enhancers and extended short predictions in both HL and FL, resulting in enhancers that closely resemble the distribution of signals.

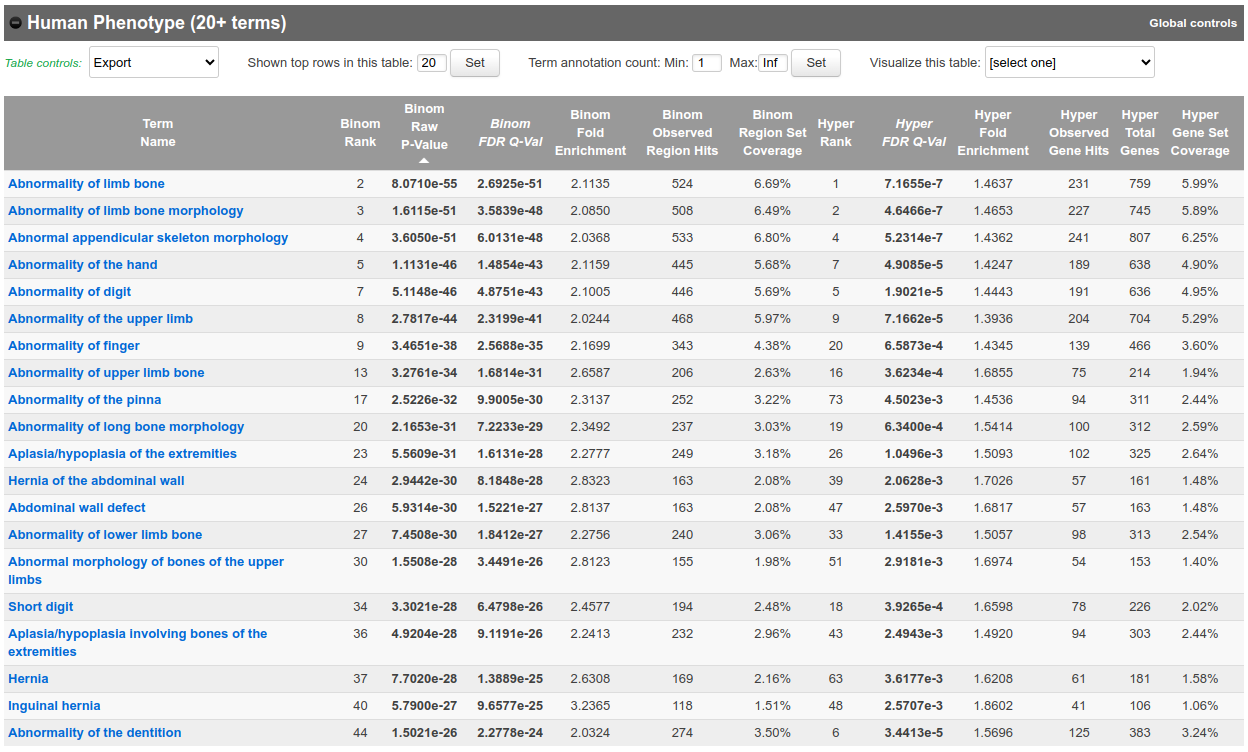
Figure 4: Enhancers within DNM3.

TBX3 in *Figure 5* is a gene typically involved in forelimb development and exhibits persistent signals in FL as expected, which may or may not correspond to an actual enhancer (see the discussion above) (NCBI, n. d.). CRUP's tendency to fragment major enhancers into smaller segments is also evident in this context, but the scripts address this issue almost perfectly, resulting in a single, significantly larger enhancer prediction.

Figure 5: Enhancers within TBX3.

Although many more individual cases of predictions consistent with raw ChIP-seq data can be shown, making judgments solely based on visual appearance can introduce bias. Therefore, the filtering criteria detailed in *Methods and Results* were applied to the processed list of enhancers, and the results were inputted to GREAT. The outcome was surprisingly definitive; the enhancers in FL enriched several biological process terms related to limb development (*Figure 6*), and phenotype terms almost exclusively related to the forelimb (*Figure 7*). In both cases, the p-values were extremely significant. Overall, the results obtained from GREAT strongly support the effectiveness of the software.

Figure 6: Significantly enriched biological processes.

Figure 7: Significantly enriched human phenotypes.

Internship Experience: Observations and Analysis

The internship exceeded my expectations. Based on my friends’ experiences, I was expecting a mostly absent supervisor and mundane work, but Stefan came to my office almost everyday to discuss the current status of the project and future directions with me for hours. His expressed interest for my work made me feel that it mattered and motivated me to spend more effort. Whenever I had questions, he made me answer them myself by breaking them down and guiding me towards the answer, which I believe helped me learn much more effectively. He corrected my bad practices and taught me life lessons about how to approach research, developing software, managing data etc., all of which I noted down. It was an academically and personally satisfying experience that I will always cherish and that made me confident in pursuing bioinformatics further.

The overall experience was markedly different from handling coursework. Lectures themselves take relatively little time, and one is free to plan the remainder of the week according to their preferences and course responsibilities. Although I can get to work when I need to, I work most effectively with random bursts of creativity - I usually study for hours on end with such bursts, regardless of the time of the day, even though I then sometimes skip studying for a couple of days. As such, the pace of Sabancı was perfect for me to be most productive. However, working from 9 am to 5 pm was a completely different experience. In my first week, I continued working until late at night to complete what I started in the morning, but was warned by my supervisor to limit working to working hours. I soon realised how important that was, as I was not busy throughout most of the week (unlike at university), and I felt exhausted at the end of almost every working day. Considering that I also had chores to deal with during the week, I would burn out if I had not recharged every evening. Spending leisure time outside was much less spontaneous than either at Sabancı or Gothenburg (where I did my Erasmus), and I could only visit places and meet with people only during the weekend. Free time was strictly free time – not allocated for completing work. Having a set schedule somewhat lowered my productivity, but I could still get work done much quicker than Stefan expected me to.

Even though I was a junior double major student who had not yet taken certain critical courses in the BIO and CS programs, I felt that I was more than equipped for the internship. I could program effectively on Python with the experience I accumulated through the courses I took and projects I was involved in. Not only that, but I was confident in my programming skills enough to experiment with different languages. For instance, I figured out Bash from scratch and wrote one of the key scripts in Bash. My supervisor knew neither Python nor Bash, so I had no direct help – Stefan only made suggestions for my algorithms. My biology background was also sufficient to understand articles, answer Stefan’s questions, and properly think about the problems I faced.

All the objectives of the internship stated in the learning agreement and more were met. I not only developed the software that is going to be useful to the lab in the future, but also made a key contribution to their research with my work on the data.

Conclusion

The ultimate goal of the internship was improving the accuracy of an existing enhancer prediction tool, CRUP, and develop a way to extract state-specific (“state” may refer to a tissue, condition etc.) enhancers. To this end, firstly, a tool that enables adjusting enhancer predictions based on raw ChIP-seq data was developed. This was supplemented by scripts that allowed using labelled enhancers to score and obtain those specific to desired states. Furthermore, all tools were generalised to work with all kinds of genomic regions and made customisable with user input.

The effectiveness of the tools was demonstrated on bat data. Hindlimb and forelimb enhancers were adjusted and forelimb-specific enhancers were extracted following the workflow with specific settings at each step to be inputted to GREAT. It was revealed that genes related to forelimb development were enriched by the enhancers, supporting the initial hypothesis that bats’ unique ability to fly as a mammal is rooted in an altered gene expression pattern.

Recommendations

The first and most important step is arranging the right internship for oneself. One should make a list of what they are interested in, do extensive research to find companies/universities/institutes which offer that, and be ready to mail back and forth with several employers, as it is a numbers game. It saves a lot of time to have an email template with a separate paragraph that can be customised for each recipient.

As mentioned before, during the internship, it is important to limit working to working hours. I believe that this is the hardest aspect of work life for a university student to adapt. This is critical to strike a balance between work and life and have a pleasurable internship experience.

Furthermore, one should try eating lunch with co-workers to get to know them. Unlike university, workplaces offer little opportunity for socialising especially if one is working in a non-team setting, so lunch becomes the main source of social satisfaction. It is important to note that work relationships tend to be somewhat more distanced than friendships outside of work as well.

References

Inoue, F., & Ahituv, N. (2015). Decoding enhancers using massively parallel reporter assays. *Genomics, 106*(3), 159-164. <https://doi.org/10.1016/j.ygeno.2015.06.005>

Kvon, E. Z. (2015). Using transgenic reporter assays to functionally characterize enhancers in animals. *Genomics, 106*(3), 185-192. <https://doi.org/10.1016/j.ygeno.2015.06.007>

Manke, T., Roider, H. G., & Vingron, M. (2008). Statistical modeling of transcription factor binding affinities predicts regulatory interactions. *PLoS Computational Biology, 4*(3), e1000039. <https://doi.org/10.1371/journal.pcbi.1000039>

Maricque, B. B., Dougherty, J. D., & Cohen, B. A. (2016). A genome-integrated massively parallel reporter assay reveals DNA sequence determinants of cis-regulatory activity in neural cells. *Nucleic Acids Research*, e16. <https://doi.org/10.1093/nar/gkw942>

Max Planck Institute for Molecular Genetics. (2023, September 5). *Home*. Retrieved September 14, 2023, from <https://www.molgen.mpg.de/2168/en>

Max Planck Institute for Molecular Genetics. (n.d.). *Vingron Lab*. Retrieved September 14, 2023, from <https://www.molgen.mpg.de/93952/vingron-lab>

Max Planck Society. (2022, October 3). *Nobel prize*. Retrieved September 14, 2023, from <https://www.mpg.de/nobel-prize>

Max Planck Society. (2023). *Facts and figures*. Retrieved September 14, 2023, from <https://www.mpg.de/facts-and-figures>

Max Planck Society. (2023, June 22). *Governing bodies*. Retrieved September 14, 2023, from <https://www.mpg.de/about-us/governing-bodies>

Max Planck Society. (n.d.). *History of the Max Planck Society*. Retrieved September 14, 2023, from <https://www.mpg.de/history_mpg>

Max Planck Society. (n.d.). *Short portrait of the Max Planck Society*. Retrieved September 14, 2023, from <https://www.mpg.de/short-portrait>

Max Planck Society. (n.d.). *The founding of the Kaiser Wilhelm Society (1911)*. Retrieved September 14, 2023, from <https://www.mpg.de/946619/5_event2-1911>

McLean, C. Y., Bristor, D., Hiller, M., Clarke, S. L., Schaar, B. T., Lowe, C. B., Wenger, A. M., & Bejerano, G. (2010). GREAT improves functional interpretation of cis-regulatory regions. *Nature Biotechnology, 28*(5), 495-501. <https://doi.org/10.1038/nbt.1630>

Nature. (n.d.). *Enhancer*. Scitable. Retrieved September 14, 2023, from <https://www.nature.com/scitable/definition/enhancer-163/>

NCBI. (n.d.). *DNM3 dynamin 3 [Homo sapiens (human)]*. Retrieved September 21, 2023, from [https://www.ncbi.nlm.nih.gov/gene/26052#genomic-regions-transcripts-products](https://www.ncbi.nlm.nih.gov/gene/26052" \l "genomic-regions-transcripts-products)

NCBI. (n.d.). *TBX3 T-box transcription factor 3 [Homo sapiens (human)]*. Retrieved September 22, 2023, from https://www.ncbi.nlm.nih.gov/gene/6926

Panigrahi, A., & O’Malley, B. W. (2021). Mechanisms of enhancer action: The known and the unknown. *Genome Biology, 22*(1). <https://doi.org/10.1186/s13059-021-02322-1>

Pennacchio, L. A., Bickmore, W., Dean, A., Nobrega, M. A., & Bejerano, G. (2013). Enhancers: Five essential questions. *Nature Reviews Genetics, 14*(4), 288-295. <https://doi.org/10.1038/nrg3458>

Ramisch, A., Heinrich, V., Glaser, L. V., Fuchs, A., Yang, X., Benner, P., Schöpflin, R., Li, N., Kinkley, S., Römer-Hillmann, A., Longinotto, J., Heyne, S., Czepukojc, B., Kessler, S. M., Kiemer, A. K., Cadenas, C., Arrigoni, L., Gasparoni, N., Manke, T., … Vingron, M. (2019). CRUP: A comprehensive framework to predict condition-specific regulatory units. *Genome Biology, 20*(1). <https://doi.org/10.1186/s13059-019-1860-7>

Roider, H. G., Kanhere, A., Manke, T., & Vingron, M. (2006). Predicting transcription factor affinities to DNA from a biophysical model. *Bioinformatics, 23*(2), 134-141. <https://doi.org/10.1093/bioinformatics/btl565>

Tang, L. (2022). Predicting and designing enhancers. *Nature Methods, 19*(7), 783-783. <https://doi.org/10.1038/s41592-022-01557-5>

Tanigawa, Y., Dyer, E. S., & Bejerano, G. (2022). WhichTF is functionally important in your open chromatin data? *PLOS Computational Biology, 18*(8), e1010378. <https://doi.org/10.1371/journal.pcbi.1010378>

Visel, A., Blow, M. J., Li, Z., Zhang, T., Akiyama, J. A., Holt, A., Plajzer-Frick, I., Shoukry, M., Wright, C., Chen, F., Afzal, V., Ren, B., Rubin, E. M., & Pennacchio, L. A. (2009). ChIP- SEQ accurately predicts tissue-specific activity of enhancers. *Nature, 457*(7231), 854-858. <https://doi.org/10.1038/nature07730>

Visel, A., Bristow, J., & Pennacchio, L. A. (2007). Enhancer identification through comparative genomics. *Seminars in Cell & Developmental Biology, 18*(1), 140-152. <https://doi.org/10.1016/j.semcdb.2006.12.014>

Wu, C., Li, X., Yuan, W., Chen, G., Kilian, A., Li, J., Xu, C., Li, X., Zhou, D., Wang, S., & Zhang, Q. (2003). Development of enhancer trap lines for functional analysis of the rice genome. *The Plant Journal, 35*(3), 418-427. <https://doi.org/10.1046/j.1365-313x.2003.01808.x>

Yao, L., Berman, B. P., & Farnham, P. J. (2015). Demystifying the secret mission of enhancers: Linking distal regulatory elements to target genes. *Critical Reviews in Biochemistry and* *Molecular Biology, 50*(6), 550-573. <https://doi.org/10.3109/10409238.2015.1087961>

Zhao, H., Zhang, W., Zhang, T., Lin, Y., Hu, Y., Fang, C., & Jiang, J. (2020). Genome-wide MNase hypersensitivity assay unveils distinct classes of open chromatin associated with H3K27me3 and DNA methylation in arabidopsis thaliana. *Genome Biology, 21*(1). https://doi.org/10.1186/s13059-020-1927-5

Appendixes

*Appendix A*

GitHub repository where all the written code can be found: <https://github.com/insciencewetrust/CRUP-Enhancer-Tweaks>

Used inputs and outputs were excluded from the repository due to data privacy concerns.