

CoolBox: A flexible toolkit for visual analysis of genomics data

Weize Xu^{1,2}, Quan Zhong³, Da Lin^{1,2}, Guoliang Li^{3,4,5,6,*} and Gang Cao^{1,2,3,4,*}

¹College of Veterinary Medicine, Huazhong Agricultural University, Wuhan, China.

²State Key Laboratory of Agricultural Microbiology, Huazhong Agricultural University, Wuhan, China.

³College of Informatics, Huazhong Agricultural University, Wuhan, China.

⁴Bio-Medical Center, Huazhong Agricultural University, Wuhan, China

⁵National Key Laboratory of Crop Genetic Improvement, Huazhong Agricultural University, Wuhan, China

⁶Hubei Key Laboratory of Agricultural Bioinformatics, Hubei Engineering Technology Research Center of Agricultural Big Data, 3D Genomics Research Center, Huazhong Agricultural University, Wuhan, China

*email: guoliang.li@mail.hzau.edu.cn; gcao@mail.hzau.edu.cn

April 15, 2021

Abstract

We developed CoolBox, an open source toolkit for visual analysis of genomics data, which is highly compatible with the Python ecosystem, easy to use and highly customizable with a well-designed user interface. It can be used in various visualization situations like a Swiss army knife, for example, to produce high-quality genome track plots or fetch common used genomic data files with a Python script or command line, interactively explore genomic data within Jupyter environment or web browser. Moreover, owing to the highly extensible API design, users can customize their own tracks without difficulty, which can greatly facilitate analytical, comparative genomic data visualization tasks.

1 Introduction

With the rapid development of next-generation sequencing technologies, more and more genomic assays have been developed for profiling the genome from various aspects, such as RNA expression[18], protein-DNA binding[20], chromatin accessibility[4] and 3D structure[16, 8]. By integrating data from these different kinds of assays or the so-called multi-omics approach, biologists can comprehensively investigate genome dynamics during biological processes. This methodology has been successfully applied to many biological fields, such as neurological diseases[6], development of nervous system[22], virus infection[10, 5]. Data visualization, especially the genome track like plots, is crucial for exploring or demonstrating some local or global properties of the genome data.

Many visualization tools have been developed to meet these demands, and these tools can be classified into three categories: (1) Command-line plotting tool[17, 2], (2) Graphical User Interface(GUI) software[13], and (3) Web-based track browser[23, 14, 12]. Each of them has its own advantages and limitations for different situations; for example, command-line tools are convenient for bioinformaticians who prefer the command-line environment to quickly plot and check their results. GUI tools are friendly to people without programming skills. Web-based browsers enable users to share the visualization result with others over the internet. These kinds of tools work well for providing an overview of the input genomic data. However, during actual scientific research, we need more than just the basic view of the data. There are more needs for comparative and analytical data visualization; for example, to visualize the differential contact interaction(DCI) of two Hi-C contact matrices[3] or predicted chromatin loops on the matrix[21]. In most cases, bioinformaticians work in programmatic and

interactive environments like RStudio, IPython console and Jupyter notebook to complete the data analysis, algorithm development and visualization tasks. However, there is a gap between the data analysis ecosystem and the existing genomic data visualization tools. Researchers spend on unnecessary stuff like file format conversion and environment switching. Therefore, a versatile tool that can fill this gap will significantly facilitate the genomics study.

To fill this gap, we developed CoolBox, a versatile toolkit for visual analysis of genomic data that combines advantages of existing tools, highly compatible with the Python scientific ecosystem, highly customizable, and easy to use with intuitive interface design and simple installation procedure. It can be used in different scenarios: (1) Python script or another python package for plotting and data fetching; (2) Shell as a command-line plotting tool; (3) Jupyter notebook environment for data fetching, plotting, and exploration; and (4) Web application for exploration and demonstration within a web browser.

2 Materials and methods

2.1 Requirement and installation

CoolBox is implemented with Python3; all dependencies can be installed and managed easily with conda(Anaconda package management tool) and the Bioconda channel[9]. CoolBox can be installed from the Bioconda channel using a single command line: “conda install -c bioconda coolbox”. Alternatively, users can utilize the latest features by installing from the source. CoolBox is developed and tested under Unix-based operating systems (Linux and macOS). Windows users can use it within Windows Subsystem for Linux(WSL) or Linux docker container.

2.2 Implementation

The plotting system of CoolBox is based on the matplotlib package. A part of the plotting code in the CoolBox is a fork from pyGenomeTracks[17] package. The data stored in bigWig, “.cool” and “.hic” file format are loaded through pybbi(<https://github.com/nvictus/pybbi>), cooler [1] and straw[7] package. Pairwise interaction data in BEDPE and Pairs format is indexed and randomly accessed using the pairix software (<https://github.com/4dn-dcic/pairix>). Other text-based genomic feature data format, like BED, GTF, and BedGraph is indexed and random accessed using the tabix[15] software. The widget panel in the GUI is implemented by using the ipywidgets package.

2.3 Availability

CoolBox is open-source under GPLv3 license at GitHub: <https://github.com/GangCaoLab/CoolBox>. It can be downloaded from this site, or directly installed from the Bioconda channel. Detailed usage about API and CLI and various data visualization example is available in the online documentation: <https://gangcaolab.github.io/CoolBox/index.html>. An interactive online demonstration notebook about a small testing dataset is available on binder: <https://mybinder.org/v2/gh/GangCaoLab/CoolBox/master?filepath=tests%2FTestRegion.ipynb>.

3 Result and discussion

3.1 Flexible and user-friendly API and CLI for producing high-quality genome track plots

The interface of CoolBox includes an Application Programming Interface(API) for using it in Python script or Jupyter environment and a Command Line Interface(CLI) for using it in Shell. Its design is inspired by the popular R package ggplot2 [24]. It allows users to compose their figures with highly intuitive syntax. In CoolBox, users can use the “+” operator in Python or “add” command in Shell to compose low-level track elements to a higher-level figure. For example, they can compose track objects of various kinds of genomic data into a single frame and interactively review interested

regions in genome browser with few lines of Python codes or Shell commands (Fig.1). Besides the 1-dimensional viewing mode supported by most other visualization tools, CoolBox supports a joint-view mode that enables users to visualize trans or cis-remote regions in a Hi-C contact matrix (Fig.2).

Most sets of commonly generated genomic assay data such as RNA-Seq, ChIP-Seq, ATAC-Seq, Hi-C, HiChIP[19] data which stored in bedgraph, bigwig[11], cool[1], .hic[7] file format (see Table.1) can be visualized in CoolBox by different kinds of tracks. Most tracks' features (color, height, style, etc.) can also be configured the same way via the API or CLI. In the CoolBox plotting system, the plot contains not only a single layer. Users can put another layer (Coverage) upon the original plot to produce more comprehensive and high-quality figures. Moreover, the output figures can be conveniently saved in different kinds of image formats such as PNG, JPEG, PDF, and SVG.

More details about how to use the API and CLI are available in the online documents.

3.2 Interactive exploration and reproducible analysis on genomic data

As shown in Fig.3, CoolBox provides a GUI for interactive data visualization, by which users can explore different genome regions by operating a simple widget panel and visualize the data within this region.

Besides, the data and the figures are bound together by Python objects. In this way, users can get the precise data of each track within the current view of the genome region through the API. Such design facilitates comparative visualization and statistical analysis. CoolBox can also be used as a general genomic-file reading package. Data within a particular genome region can be retrieved in a short time, as almost all supported file formats can be indexed and randomly accessed.

Moreover, by leveraging the power of the Jupyter notebook, the visualization result and the entire process can be recorded in the notebook. It is convenient for sharing the visualization result and reproducing the whole analysis by other researchers.

3.3 A testing and visualizing framework for new algorithm development

Owing to the user-friendly and highly extensible API design, users can implement their custom tracks with no difficulty, thus enabling seamless cooperation in Python-based algorithm development and scientific research. The algorithm developer can check and visualize the intermediate result produced by their algorithm and adjust parameters simultaneously. Besides, because CoolBox uses an object-oriented programming paradigm in its design, users can reuse each track's codes by inheritance, including data extraction and drawing-related functions. In most cases, users only need to write algorithm-related core parts, and the most tedious part including raw-data reading, preprocessing, and figure drawing are handed over to CoolBox through inheritance(implementation see method section). In this way, bioinformaticians can free themselves from those repetitive procedures and only focuses on the data post-processing.

We demonstrate these advantages by implementing a track that visualizes the outputs of the Peakachu algorithm[21], which is a RandomForest based method for detecting loops in the Hi-C contact matrix. As depicted in Fig.4, the main part of the whole track contains merely 20 lines of Python code. The data fetching and plotting functionality are fully reused by inheriting Cool/ArcsBase Track base class. Furthermore, the custom-defined track is empowered to be used in CLI, API, and browser mode in couple with other built-in tracks. More details of this example includes a reproducible code block and can be found in the online documentation.

3.4 Comparison with other existing visualization tools

As stated before, there is an urgent need for better visualization tools to accelerate the integration and mining of biological data. Therefore, more and more visualization tools have been developed in recent years. A comparison of features between CoolBox and these tools is listed in Table.2. Most of the visualization tools require a tedious installation process and are operated through the command line. Before visualization, the data needs to be preprocessed through specific steps, and then a static or interactive web interface is generated.

The visualization and data processing of most visualization tools are dissociated, which is not convenient for bioinformaticians whose routine works rely on Python-based scientific computation ecosystem. Except for the CLI mode supported by most visualization tools, the API that the CoolBox has been used internally and exposed follows the same design idea as the CLI, making switching between these two modes with no pain. More importantly, the API in CoolBox combines computation and visualization, users can dynamically add different tracks or even custom tracks in the python notebook while processing raw data or developing new methods.

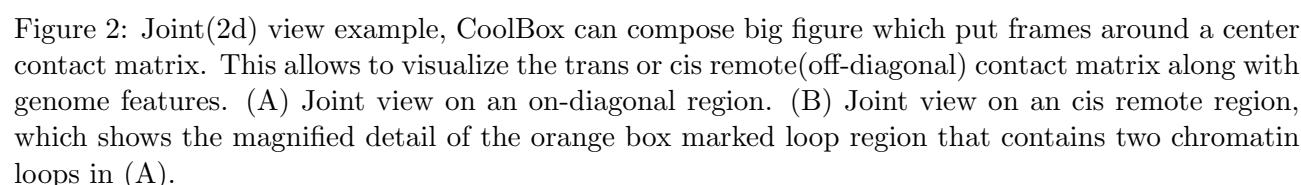
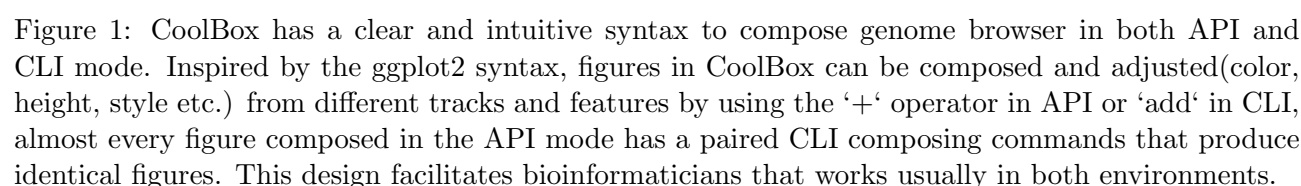
4 Conclusion

CoolBox is a versatile toolkit for the visualization and exploration of multi-omics data in the Python ecosystem. It provides a user-friendly ggplot2-like syntax for composing various kinds of tracks in CLI, API, GUI and web browser mode. More importantly, it's built on a highly extensible plotting system that allows users to implement their custom tracks without wasting time on data fetching and figure plotting procedures. Through the power of Jupyter notebook, it provides a convenient way for bioinformaticians to exploit this tool's versatility for better personalized data manipulation and demonstration. It could also increase the reproducibility of genomic data visualization tasks as codes and figures are all organized into the same page.

References

- [1] Nezar Abdennur and Leonid A Mirny. Cooler: scalable storage for Hi-C data and other genomically labeled arrays. *Bioinformatics*, 07 2019.
- [2] Kadir Caner Akdemir and Lynda Chin. Hicplotter integrates genomic data with interaction matrices. *Genome biology*, 16(1):198, 2015.
- [3] Abbas Roayaei Ardakany, Ferhat Ay, and Stefano Lonardi. Selfish: discovery of differential chromatin interactions via a self-similarity measure. *Bioinformatics*, 35(14):i145–i153, 2019.
- [4] Jason D Buenrostro, Paul G Giresi, Lisa C Zaba, Howard Y Chang, and William J Greenleaf. Transposition of native chromatin for multimodal regulatory analysis and personal epigenomics. *Nature methods*, 10(12):1213, 2013.
- [5] Canhui Cao, Ping Hong, Xingyu Huang, Da Lin, Gang Cao, Liming Wang, Bei Feng, Ping Wu, Hui Shen, Qian Xu, et al. Hpv-ccdc106 integration alters local chromosome architecture and hijacks an enhancer by three-dimensional genome structure remodeling in cervical cancer. *Journal of Genetics and Genomics*, 47(8):437–450, 2020.
- [6] M Ryan Corces, Anna Shcherbina, Soumya Kundu, Michael J Gloudemans, Laure Frésard, Jeffrey M Granja, Bryan H Louie, Tiffany Eulalio, Shadi Shams, S Tansu Bagdatli, et al. Single-cell epigenomic analyses implicate candidate causal variants at inherited risk loci for alzheimer's and parkinson's diseases. *Nature genetics*, 52(11):1158–1168, 2020.
- [7] Neva C Durand, Muhammad S Shamim, Ido Machol, Suhas SP Rao, Miriam H Huntley, Eric S Lander, and Erez Lieberman Aiden. Juicer provides a one-click system for analyzing loop-resolution hi-c experiments. *Cell systems*, 3(1):95–98, 2016.
- [8] Melissa J Fullwood and Yijun Ruan. Chip-based methods for the identification of long-range chromatin interactions. *Journal of cellular biochemistry*, 107(1):30–39, 2009.
- [9] Björn Grüning, Ryan Dale, Andreas Sjödin, Brad A Chapman, Jillian Rowe, Christopher H Tomkins-Tinch, Renan Valieris, and Johannes Köster. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nature methods*, 15(7):475–476, 2018.

- 178 [10] Sven Heinz, Lorane Texari, Michael GB Hayes, Matthew Urbanowski, Max W Chang, Ninvita
179 Givarkes, Alexander Rialdi, Kris M White, Randy A Albrecht, Lars Pache, et al. Transcription
180 elongation can affect genome 3d structure. *Cell*, 174(6):1522–1536, 2018.
- 181 [11] W James Kent, Ann S Zweig, G Barber, Angie S Hinrichs, and Donna Karolchik. Bigwig and
182 bigbed: enabling browsing of large distributed datasets. *Bioinformatics*, 26(17):2204–2207, 2010.
- 183 [12] Peter Kerpedjiev, Nezar Abdennur, Fritz Lekschas, Chuck McCallum, Kasper Dinkla, Hendrik
184 Strobelt, Jacob M Lubner, Scott B Ouellette, Alaleh Azhir, Nikhil Kumar, et al. Hiclass: web-
185 based visual exploration and analysis of genome interaction maps. *Genome biology*, 19(1):1–12,
186 2018.
- 187 [13] Rajendra Kumar, Haitham Sobhy, Per Stenberg, and Ludvig Lizana. Genome contact map
188 explorer: a platform for the comparison, interactive visualization and analysis of genome contact
189 maps. *Nucleic acids research*, 45(17):e152–e152, 2017.
- 190 [14] Daofeng Li, Silas Hsu, Deepak Purushotham, Renee L Sears, and Ting Wang. Washu epigenome
191 browser update 2019. *Nucleic acids research*, 47(W1):W158–W165, 2019.
- 192 [15] Heng Li. Tabix: fast retrieval of sequence features from generic tab-delimited files. *Bioinformatics*,
193 27(5):718–719, 2011.
- 194 [16] Erez Lieberman-Aiden, Nynke L Van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy,
195 Agnes Telling, Ido Amit, Bryan R Lajoie, Peter J Sabo, Michael O Dorschner, et al. Comprehen-
196 sive mapping of long-range interactions reveals folding principles of the human genome. *science*,
197 326(5950):289–293, 2009.
- 198 [17] Lucille Lopez-Delisle, Leily Rabbani, Joachim Wolff, Vivek Bhardwaj, Rolf Backofen, Björn
199 Grüning, Fidel Ramírez, and Thomas Manke. pygenometracks: reproducible plots for multi-
200 variate genomic data sets. *Bioinformatics (Oxford, England)*, 2020.
- 201 [18] Ryan D Morin, Matthew Bainbridge, Anthony Fejes, Martin Hirst, Martin Krzywinski, Trevor J
202 Pugh, Helen McDonald, Richard Varhol, Steven JM Jones, and Marco A Marra. Profiling the
203 hela s3 transcriptome using randomly primed cDNA and massively parallel short-read sequencing.
204 *Biotechniques*, 45(1):81–94, 2008.
- 205 [19] Maxwell R Mumbach, Adam J Rubin, Ryan A Flynn, Chao Dai, Paul A Khavari, William J
206 Greenleaf, and Howard Y Chang. Hicchip: efficient and sensitive analysis of protein-directed
207 genome architecture. *Nature methods*, 13(11):919–922, 2016.
- 208 [20] Gordon Robertson, Martin Hirst, Matthew Bainbridge, Misha Bilenky, Yongjun Zhao, Thomas
209 Zeng, Ghia Euskirchen, Bridget Bernier, Richard Varhol, Allen Delaney, et al. Genome-wide
210 profiles of stat1 DNA association using chromatin immunoprecipitation and massively parallel
211 sequencing. *Nature methods*, 4(8):651, 2007.
- 212 [21] Tarik J Salameh, Xiaotao Wang, Fan Song, Bo Zhang, Sage M Wright, Chachrit Khunsiraksakul,
213 Yijun Ruan, and Feng Yue. A supervised learning framework for chromatin loop detection in
214 genome-wide contact maps. *Nature communications*, 11(1):1–12, 2020.
- 215 [22] Michael Song, Mark-Phillip Pebworth, Xiaoyu Yang, Armen Abnoui, Changxu Fan, Jia Wen,
216 Jonathan D Rosen, Mayank NK Choudhary, Xiekui Cui, Ian R Jones, et al. Cell-type-specific 3d
217 epigenomes in the developing human cortex. *Nature*, 587(7835):644–649, 2020.
- 218 [23] Yanli Wang, Fan Song, Bo Zhang, Lijun Zhang, Jie Xu, Da Kuang, Daofeng Li, Mayank NK
219 Choudhary, Yun Li, Ming Hu, et al. The 3d genome browser: a web-based browser for visualizing
220 3d genome organization and long-range chromatin interactions. *Genome biology*, 19(1):1–12, 2018.
- 221 [24] Hadley Wickham. ggplot2: elegant graphics for data analysis. *J Stat Softw*, 35(1):65–88, 2010.



Track type	File format	Description
XAxis	None	Coordinate of the reference genome.
Spacer	None	For add vertical space between two tracks.
BigWig	.bigwig	Track for bigWig file, draw the histogram.
BedGraph	.bedgraph	Track for BedGraph file, draw the histogram.
BAM	.bam	BAM track for visualize the coverage or alignment.
BED	.bed	For visualization genome annotation, like refSeq genes and chromatin states.
GTF	.gtf	Track of GTF file, for visualize gene annotation.
Arcs	.pairs, .bedpe	Show the chromosome interactions get from ChIA-PET, HiChIP or Hi-C loop data.
HiCMat	.cool, .mcool, .hic	Show the chromosome contact matrix from Hi-C data.
Virtual4C	.cool, .mcool, .hic	Virtual 4C track, using Hi-C data to mimic 4C.
DiScore	.cool, .mcool, .hic	Directional index of Hi-C matrix for detecting TAD.
InsuScore	.cool, .mcool, .hic	Insulation score of Hi-C matrix for inferring TAD borders.
HiCDiff	.cool, .mcool, .hic	Show the difference between two contact matrix.
Selfish	.cool, .mcool, .hic	Apply the selfish algorithm[3] on two contact matrices to detect differential contact interactions.
SNP	.tsv	Track for show SNPs Manhattan plot.

Table 1: A part of CoolBox builtin tracks for visualizing different kinds of genomics data formats.

∞

Tools	Programming language	API plot	CLI plot	Online Access	GUI	Input	Installation	Customization	
CoolBox	Python	✓	✓	✓	Web and Jupyter	Raw data	Bioconda or PyPI	Python	knowledge, very easy
pyGenomeTracks	Python		✓			Raw data	PyPI	Python	knowledge, easy
gcMapExplorer	Python	✓			Local	Preprocessed data	PyPI	Python	knowledge, easy
HiCPlotter	Python		✓			Preprocessed data	Manually install		
HiGlass	Python, HTML,CSS,JS	✓		✓	Web and Jupyter	Preprocessed data, via network	Docker	Web	knowledge
YueLab Browser	HTML,CSS,JS			✓	Web	Via network			
WashU Browser	HTML,CSS,JS			✓	Web	Via network			
TADkit	HTML,CSS,JS			✓	Web	Preprocessed data, via network	Manually install		
JuiceBox.js	HTML,CSS,JS			✓	Web	Via network			
JuiceBox	Java				Local	Raw data	Download		

Table 2: Summary of genomic visualization tools.

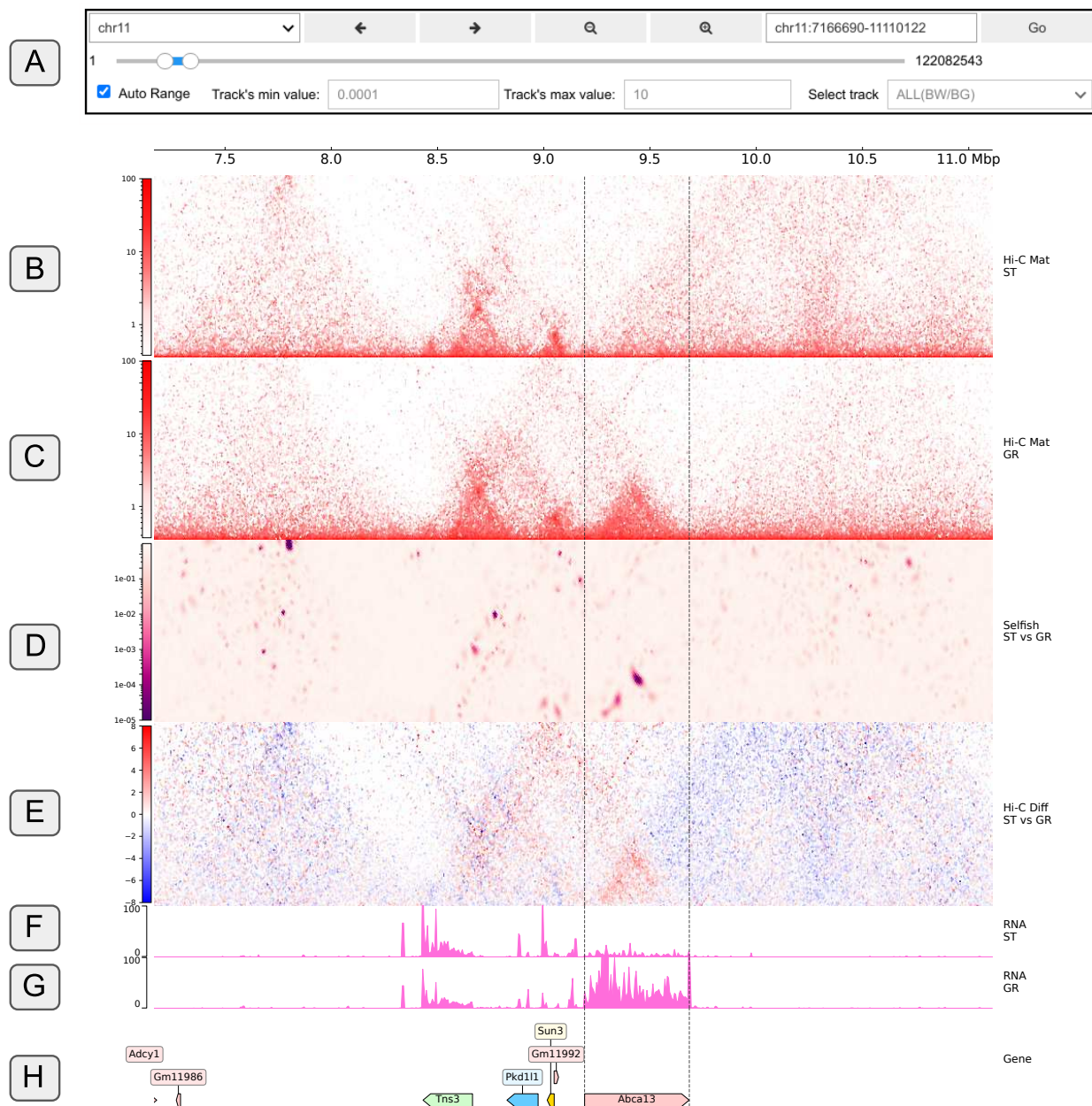


Figure 3: A CoolBox figure representing differential interactions of Hi-C contact matrices. Shown Hi-C and RNA-Seq data are produced from the process of hematopoietic differentiation[25]. It is clearly that there is a topological association domain (TAD) formation at the Abca13 gene region and its RNA expression is up-regulated at the same time after the differentiation. (A) The widgets panel of CoolBox browser, used for zooming, sliding, and locating the genome region. When moving to a new region, the figure drawn below will be updated automatically. (B) Hi-C track of short-term hematopoietic stem cell (ST) shows the contact map of ST sample. The color bar indicates the normalized value of the contact map. (C) Hi-C track of granulocyte (GR). (D) Differential contact interaction (DCI) result of the Selfish algorithm[3] on ST and GR Hi-C contact map. The color bar indicates the q-value (BH adjusted p-value) produced from the DCI analysis. Darker color means this interaction has a lower q-value; that is to say, two contact maps are more diverse at this location. (E) Hi-C Diff track. It shows the difference between GR and ST's z-score normalized contact matrices. The red region of the matrix indicates where GR has a more significant contact frequency compared to ST, and the opposite for blue areas. (F) BigWig track of ST RNA-Seq data, showing the RNA expression level of ST in this region. (G) BigWig track of GR RNA-Seq data. (H) A gene annotation track shows the corresponding genes within this genomic region.

