

# 1 CoolBox: A flexible toolkit for visual analysis of genomics data

2 Weize Xu<sup>1,2</sup>, Quan Zhong<sup>3</sup>, Da Lin<sup>1,2</sup>, Guoliang Li<sup>3,4,5,6,\*</sup> and Gang Cao<sup>1,2,3,4,\*</sup>

3 <sup>1</sup>College of Veterinary Medicine, Huazhong Agricultural University, Wuhan, China.

4 <sup>2</sup>State Key Laboratory of Agricultural Microbiology, Huazhong Agricultural University,  
5 Wuhan, China.

6 <sup>3</sup>College of Informatics, Huazhong Agricultural University, Wuhan, China.

7 <sup>4</sup>Bio-Medical Center, Huazhong Agricultural University, Wuhan, China

8 <sup>5</sup>National Key Laboratory of Crop Genetic Improvement, Huazhong Agricultural University,  
9 Wuhan, China

10 <sup>6</sup>Hubei Key Laboratory of Agricultural Bioinformatics, Hubei Engineering Technology  
11 Research Center of Agricultural Big Data, 3D Genomics Research Center, Huazhong  
12 Agricultural University, Wuhan, China

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

14 April 15, 2021

## 15 Abstract

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

## 24 1 Introduction

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

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

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

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

## 56 2 Materials and methods

### 57 2.1 Requirement and installation

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

### 64 2.2 Implementation

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

### 72 2.3 Availability

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

## 79 3 Result and discussion

### 80 3.1 Flexible and user-friendly API and CLI for producing high-quality genome 81 track plots

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

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

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

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

### 99 **3.2 Interactive exploration and reproducible analysis on genomic data**

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

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

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

### 111 **3.3 A testing and visualizing framework for new algorithm development**

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

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

### 129 **3.4 Comparison with other existing visualization tools**

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

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

## 143 4 Conclusion

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

## 152 References

- 153 [1] Nezar Abdennur and Leonid A Mirny. Cooler: scalable storage for Hi-C data and other genomi-  
154 cally labeled arrays. *Bioinformatics*, 07 2019.
- 155 [2] Kadir Caner Akdemir and Lynda Chin. Hicplotter integrates genomic data with interaction  
156 matrices. *Genome biology*, 16(1):198, 2015.
- 157 [3] Abbas Roayaei Ardakany, Ferhat Ay, and Stefano Lonardi. Selfish: discovery of differential  
158 chromatin interactions via a self-similarity measure. *Bioinformatics*, 35(14):i145–i153, 2019.
- 159 [4] Jason D Buenrostro, Paul G Giresi, Lisa C Zaba, Howard Y Chang, and William J Greenleaf.  
160 Transposition of native chromatin for multimodal regulatory analysis and personal epigenomics.  
161 *Nature methods*, 10(12):1213, 2013.
- 162 [5] Canhui Cao, Ping Hong, Xingyu Huang, Da Lin, Gang Cao, Liming Wang, Bei Feng, Ping  
163 Wu, Hui Shen, Qian Xu, et al. Hpv-ccdc106 integration alters local chromosome architecture  
164 and hijacks an enhancer by three-dimensional genome structure remodeling in cervical cancer.  
165 *Journal of Genetics and Genomics*, 47(8):437–450, 2020.
- 166 [6] M Ryan Corces, Anna Shcherbina, Soumya Kundu, Michael J Gloudemans, Laure Frésard, Jef-  
167 frey M Granja, Bryan H Louie, Tiffany Eulalio, Shadi Shams, S Tansu Bagdatli, et al. Single-cell  
168 epigenomic analyses implicate candidate causal variants at inherited risk loci for alzheimer's and  
169 parkinson's diseases. *Nature genetics*, 52(11):1158–1168, 2020.
- 170 [7] Neva C Durand, Muhammad S Shamim, Ido Machol, Suhas SP Rao, Miriam H Huntley, Eric S  
171 Lander, and Erez Lieberman Aiden. Juicer provides a one-click system for analyzing loop-  
172 resolution hi-c experiments. *Cell systems*, 3(1):95–98, 2016.
- 173 [8] Melissa J Fullwood and Yijun Ruan. Chip-based methods for the identification of long-range  
174 chromatin interactions. *Journal of cellular biochemistry*, 107(1):30–39, 2009.
- 175 [9] Björn Grüning, Ryan Dale, Andreas Sjödin, Brad A Chapman, Jillian Rowe, Christopher H  
176 Tomkins-Tinch, Renan Valieris, and Johannes Köster. Bioconda: sustainable and comprehensive  
177 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 Strobel, Jacob M Luber, Scott B Ouellette, Alaleh Azhir, Nikhil Kumar, et al. Higlass: 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. Comprehensive  
196 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 Khunsriraksakul,  
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 Abnousi, 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.

222 [25] Chao Zhang, Zihan Xu, Shangda Yang, Guohuan Sun, Lumeng Jia, Zhaofeng Zheng, Quan Gu,  
 223 Wei Tao, Tao Cheng, Cheng Li, et al. taghi-c reveals 3d chromatin architecture dynamics during  
 224 mouse hematopoiesis. *Cell Reports*, 32(13):108206, 2020.

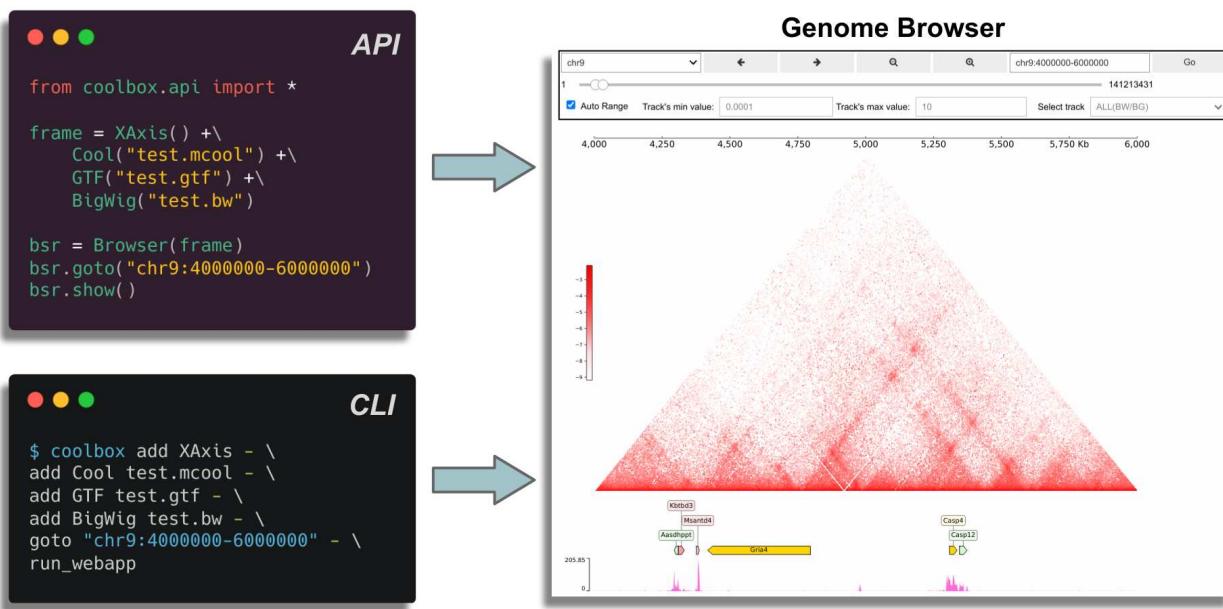


Figure 1: CoolBox has a clear and intuitive syntax to compose genome browser in both API and CLI mode. Inspired by the ggplot2 syntax, figures in CoolBox can be composed and adjusted(color, height, style etc.) from different tracks and features by using the ‘+’ operator in API or ‘add’ in CLI, almost every figure composed in the API mode has a paired CLI composing commands that produce identical figures. This design facilitates bioinformaticians that works usually in both environments.

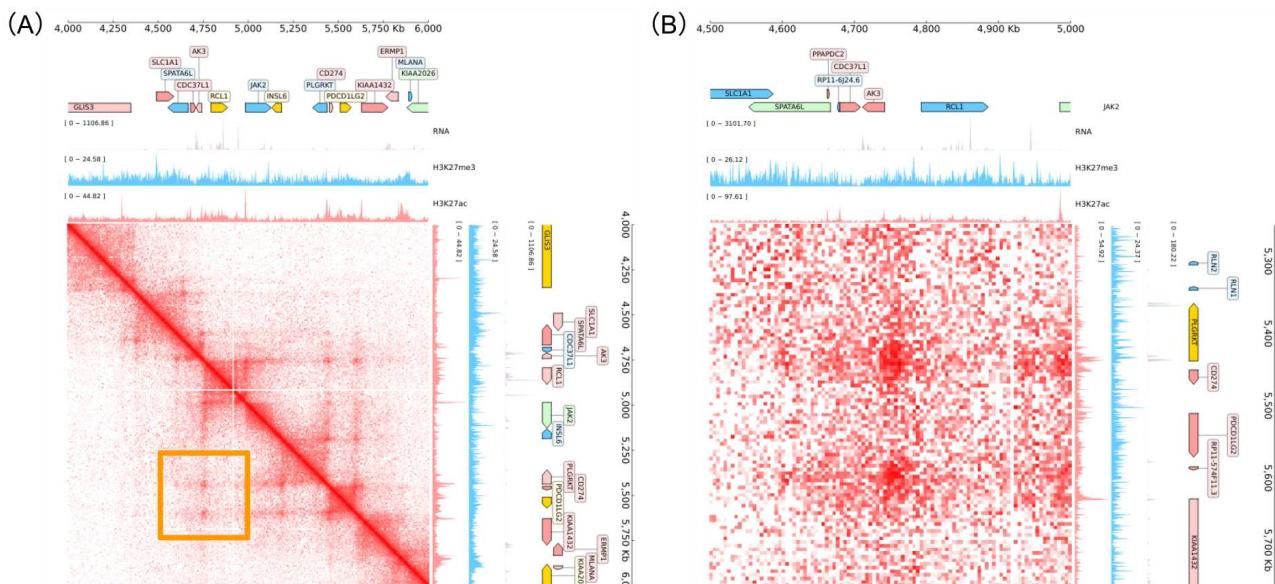


Figure 2: Joint(2d) view example, CoolBox can compose big figure which put frames around a center contact matrix. This allows to visualize the trans or cis remote(off-diagonal) contact matrix along with genome features. (A) Joint view on an on-diagonal region. (B) Joint view on a cis remote region, which shows the magnified detail of the orange box marked loop region that contains two chromatin loops in (A).

| 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.

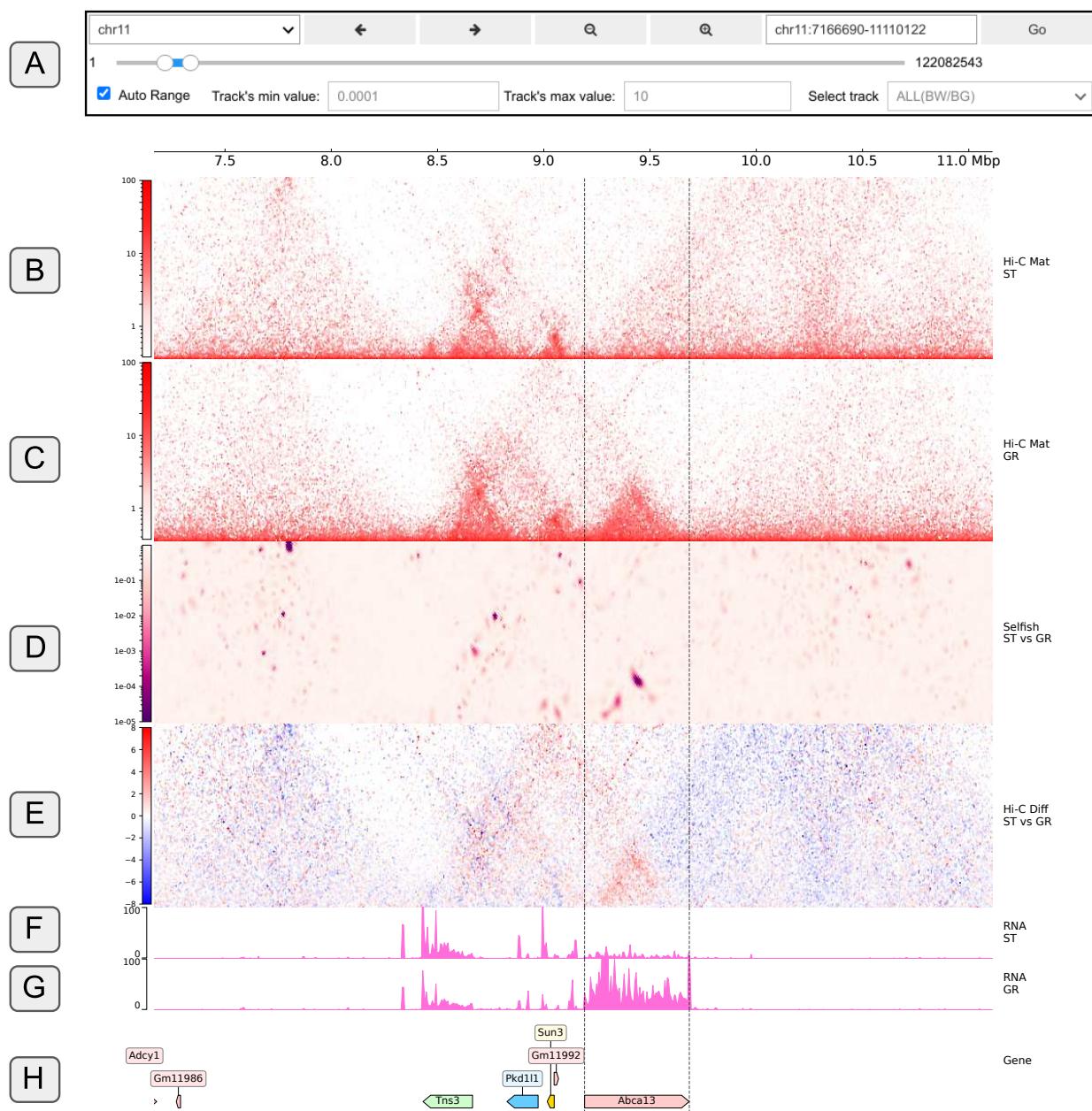


Figure 3: A CoolBox figure representing differential interactions of Hi-C contact matrices. Shown Hi-C and RNA-Seq data are produced from the process of hematopoietic differentiation[25]. It is clearly that there is a topological association domain(TAD) formation at the *Abca13* gene region and its RNA 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 draw bellow 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.

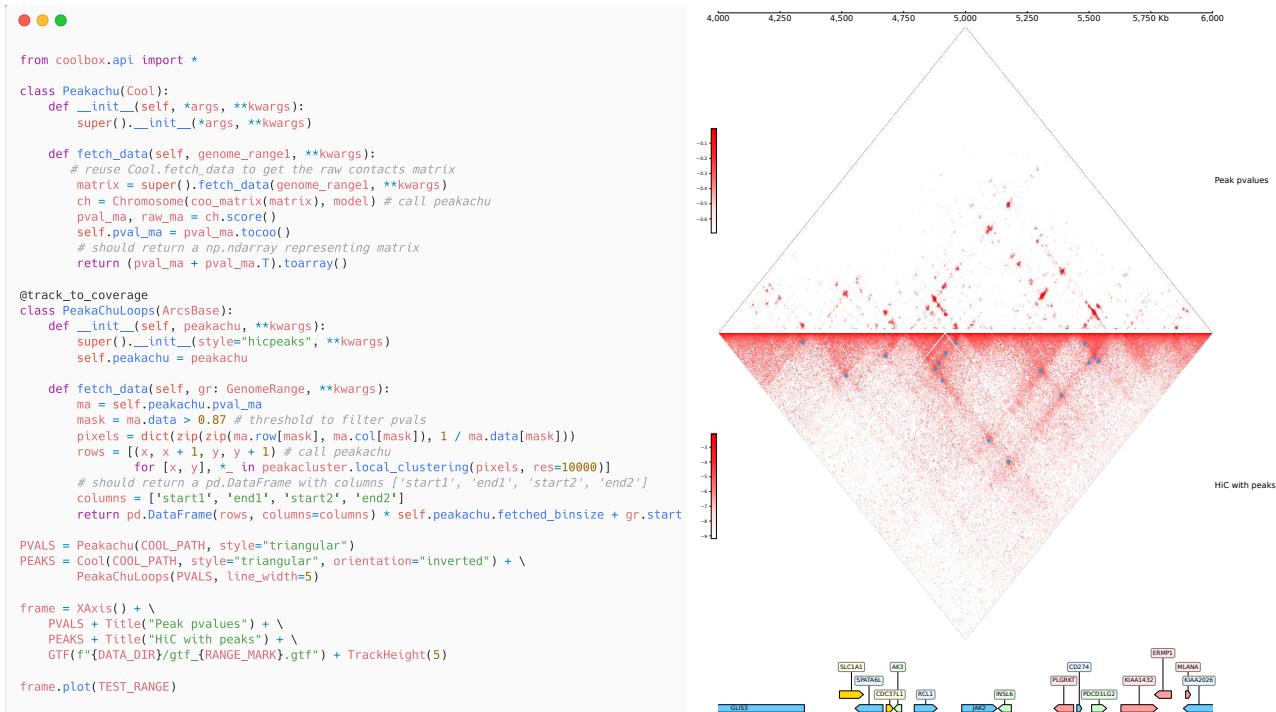


Figure 4: An example to define custom tracks that display Hi-C contact matrix along with peaks detected by Peakachu algorithm[21]. An example of peak prediction result is demonstrated in the right panel. The upper triangular matrix shows the peak p-value output by Peakachu algorithm. The predicted peaks drawn as blue squares upon the original matrix is shown in the lower triangular matrix. The Hi-C matrices and the peaks upon them will be automatically recomputed and updated after the change of genome region. The left panel is the full Runnable python codes used for generating the right panel. The custom track is implemented by following the same intuitive and clear design pattern as other built-in tracks: i.e., reusing the data fetching and plotting contact matrix with peaks are totally reused by inheriting Cool/ArcsBase track base class, and the rest of the codes merely calls the computing function of the peachachu package. After the track definition, we can see that the custom track is born to support being used in a ggplot2-like syntax with other tracks, and this capability is also valid in CLI and GUI mode.