fortune Journals

Research Article

JOURNAL OF BIOINFORMATICS AND SYSTEMS BIOLOGY

ISSN: 2688-5107

Genomics to Notebook (g2nb): Extending the Electronic Notebook to Address the Challenges of Bioinformatics Analysis

Michael Reich[1*], Thorin Tabor[1], John Liefeld[1], Jayadev Joshi[2], Forrest Kim[1], Edwin Huang[1], Helga Thorvaldsdottir[3], Daniel Blankenberg[4,5], Jill P. Mesirov[1,6]

Abstract

We present Genomics to Notebook (g2nb), an environment that combines the JupyterLab notebook system with widely-used bioinformatics platforms. The analyses and visualizations within those platforms are presented as cells in a notebook, making thousands of genomics methods available within the notebook metaphor and allowing notebooks to contain workflows utilizing multiple software packages on remote servers, all without the need for programming.

Keywords: Computational notebook; Reproducible research; Bioinformatics; Open science

Background

The computational notebook has become the de facto medium for much of data science and bioinformatics due to its accessibility and ease in incorporating scientific exposition with executable code to form a reproducible "research narrative." Jupyter Notebook[1] and its successor JupyterLab[2] are the preeminent platforms. However, there are well-established bioinformatics analysis platforms, such as Galaxy[3] and GenePattern[4], that collectively host thousands of tools on their own compute resources. A system that could incorporate these platforms into the notebook metaphor would bring the benefits of their tools to notebook users and would allow non-programming scientists to author notebooks. We have released Genomics to Notebook (g2nb), which builds on JupyterLab and our previous GenePattern Notebook environment[5] to add access to multiple bioinformatics platforms and other functionality through the components described below. While we focus on functionality for the non-programmer, we note that all programmatic features of JupyterLab are retained, making g2nb appealing to a wide range of users.

Results and Discussion

The g2nb environment incorporates multiple bioinformatics software platforms within the notebook interface. A standard Jupyter notebook consists of a sequence of cells, each of which can contain text or executable code. We have added a new cell type that provides an interface within the notebook to tools that are hosted on a remote Galaxy or GenePattern server. These new analysis cells present a form-like interface that is similar to the web interfaces of the original platforms, requiring only that an investigator provide the input parameters and data (Figure. 1). When an analysis is launched, it is executed on the specified remote server, with the job execution status displayed within the cell. When the job completes, links to the result files are presented in the notebook cell and can be used easily as input to further analyses (Figure. S1).

Affiliation:

[1]Department of Medicine, School of Medicine, University of California San Diego, La Jolla, CA, USA

[2]Genomic Medicine Institute, Lerner Research Institute, Cleveland Clinic, Cleveland, OH, USA

[3]The Broad Institute of MIT and Harvard, Cambridge, MA, USA

[4]Genomic Medicine Institute, Lerner Research Institute, Cleveland Clinic, Cleveland, OH, USA

[5]Department of Molecular Medicine, Cleveland Clinic Lerner College of Medicine, Case Western Reserve University, Cleveland, OH, USA

[6]Department of Medicine, Moores Cancer Center,University of California San Diego, La Jolla, CA, USA

***Corresponding author:**

Michael Reich, Department of Medicine, School of Medicine, University of California San Diego, La Jolla, CA, USA

Citation: Michael Reich, Thorin Tabor, John Liefeld, Jayadev Joshi, Forrest Kim, Edwin Huang, Helga Thorvaldsdottir, Daniel Blankenberg, Jill P Mesirov. Genomics to Notebook (g2nb): Extending the Electronic Notebook to Address the Challenges of Bioinformatics Analysis. Journal of Bioinformatics and Systems Biology. 8 (2025): 01-07.

Received: December 17, 2024
Accepted: December 23, 2024
Published: January 27, 2025

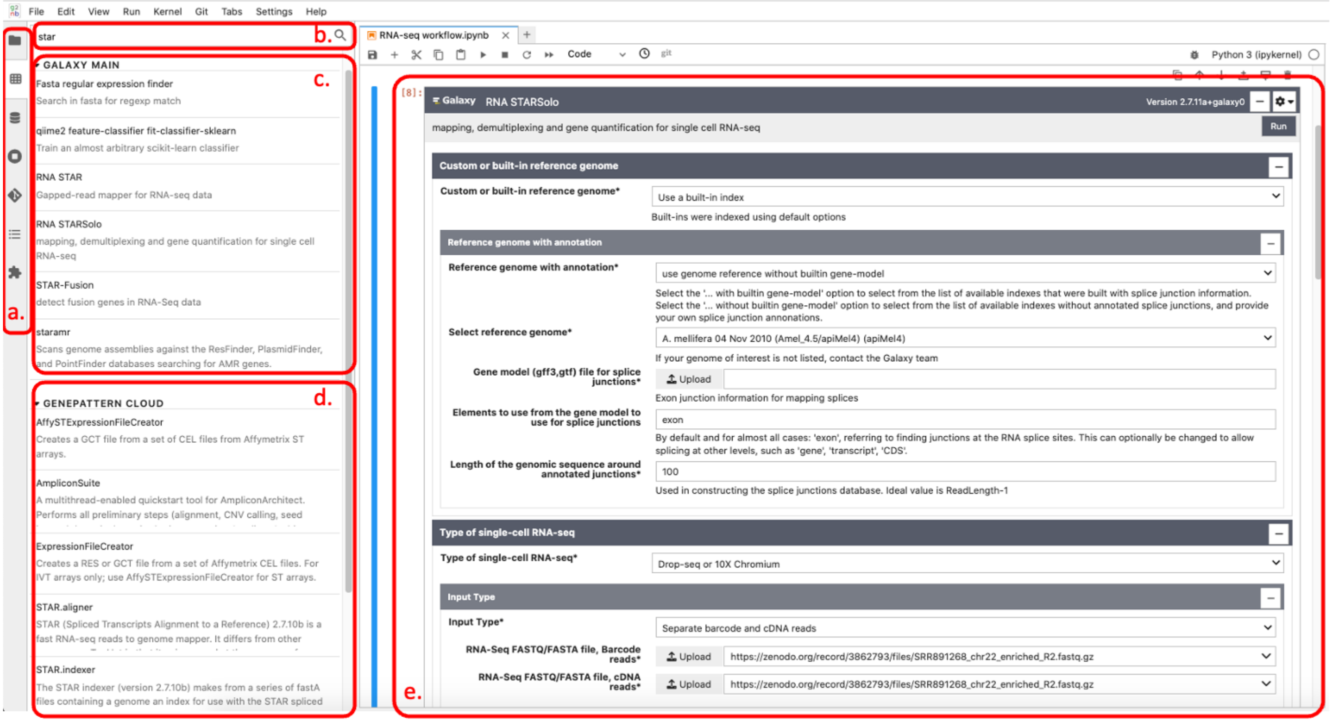
Thus, to the user, the entire analysis workflow appears to run seamlessly within the notebook.

The g2nb environment provides capabilities for interactively visualizing various genomics datatypes by including the JavaScript versions of Cytoscape[6] and IGV[7], as well as standard heatmap, dendrogram, and other plots provided in GenePattern and Galaxy. Visualizations appear within notebook cells and also can be launched in independent windows if more space is required.

To facilitate the seamless flow of data into a notebook and through notebook cells, the g2nb environment provides several new JupyterLab enhancements. These include: panels providing Galaxy histories and GenePattern result files, allowing users to easily reuse files from previous analyses; the ability to display a list of all analyses in a notebook that can receive a particular result file as input; and running a sequence of cells as an end-to-end workflow, including cells that launch jobs outside the notebook on Galaxy or GenePattern servers. Because links to results within analysis cells reference files on remote Galaxy or GenePattern servers, any necessary data transfers are handled automatically. A number of these features are illustrated in Fig 2. We have also incorporated the Globus[8] file transfer protocol into the g2nb interface, allowing users to robustly transfer files of any size between the g2nb workspace and any Globus endpoint, including the user's local storage. For programmers, we have implemented features to facilitate transfer between Python objects and Galaxy and GenePattern jobs. In a g2nb notebook, a Python variable can be provided as an input parameter to Galaxy and GenePattern analyses. The g2nb environment will evaluate the variable and pass its value to the analysis. Additionally, rather than having to download result files and read them into Python objects, users can automatically load the contents of a result file directly into a Python variable or, for array-type data, into a Pandas dataframe.

The g2nb environment provides a User Interface Builder (Figure. S3) that allows notebook developers to present Python code cells in a more user-friendly format. Notebooks frequently include cells with large blocks of code that require no interaction other than being run by the user, or for the user to provide a small number of inputs. The User Interface Builder allows authors to create a web form interface to a cell, exposing only those parameters they wish users to enter. Notebook authors can specify data inputs, text or numeric entry or dropdown lists, and can tailor the interface in many ways to the specifics of their code. The underlying code is always available via a toggle button.

We provide a freely available online workspace, g2nb.



star

GALAXY MAIN

Fasta regular expression finder
Search in fasta for regexp match

qiime2 feature-classifier fit-classifier-sklearn
Train an almost arbitrary scikit-learn classifier

RNA STAR
Gapped-read mapper for RNA-seq data

RNA STARSolo
mapping, demultiplexing and gene quantification for single cell RNA-seq

STAR-Fusion
detect fusion genes in RNA-Seq data

staramr
Scans genome assemblies against the ResFinder, PlasmidFinder, and PointFinder databases searching for AMR genes.

GENEPATTERN CLOUD

AffySTExpressionFileCreator
Creates a GCT file from a set of CEL files from Affymetrix ST arrays.

AmpliconSuite
A multithread-enabled quickstart tool for AmpliconArchitect. Performs all preliminary steps (alignment, CNV calling, seed

ExpressionFileCreator
Creates a RES or GCT file from a set of Affymetrix CEL files. For IVT arrays only; use AffySTExpressionFileCreator for ST arrays.

STAR.aligner
STAR (Spliced Transcripts Alignment to a Reference) 2.7.10b is a fast RNA-seq reads to genome mapper. It differs from other

STAR.indexer
The STAR indexer (version 2.7.10b) makes from a series of fastA files containing a genome an index for use with the STAR spliced

RNA STARSolo Version 2.7.11a+galaxy0

mapping, demultiplexing and gene quantification for single cell RNA-seq

Custom or built-in reference genome

Custom or built-in reference genome*: Use a built-in index
Built-ins were indexed using default options

Reference genome with annotation

Reference genome with annotation*: use genome reference without builtin gene-model
Select the '... with builtin gene-model' option to select from the list of available indexes that were built with splice junction information. Select the '... without builtin gene-model' option to select from the list of available indexes without annotated splice junctions, and provide your own splice junction annonations.

Select reference genome*: A. mellifera 04 Nov 2010 (Amel_4.5/apiMel4) (apiMel4)
If your genome of interest is not listed, contact the Galaxy team

Gene model (gff3,gtf) file for splice junctions*: Upload
Exon junction information for mapping splices

Elements to use from the gene model to use for splice junctions: exon
By default and for almost all cases: 'exon', referring to finding junctions at the RNA splice sites. This can optionally be changed to allow splicing at other levels, such as 'gene', 'transcript', 'CDS'.

Length of the genomic sequence around annotated junctions*: 100
Used in constructing the splice junctions database. Ideal value is ReadLength-1

Type of single-cell RNA-seq

Type of single-cell RNA-seq*: Drop-seq or 10X Chromium

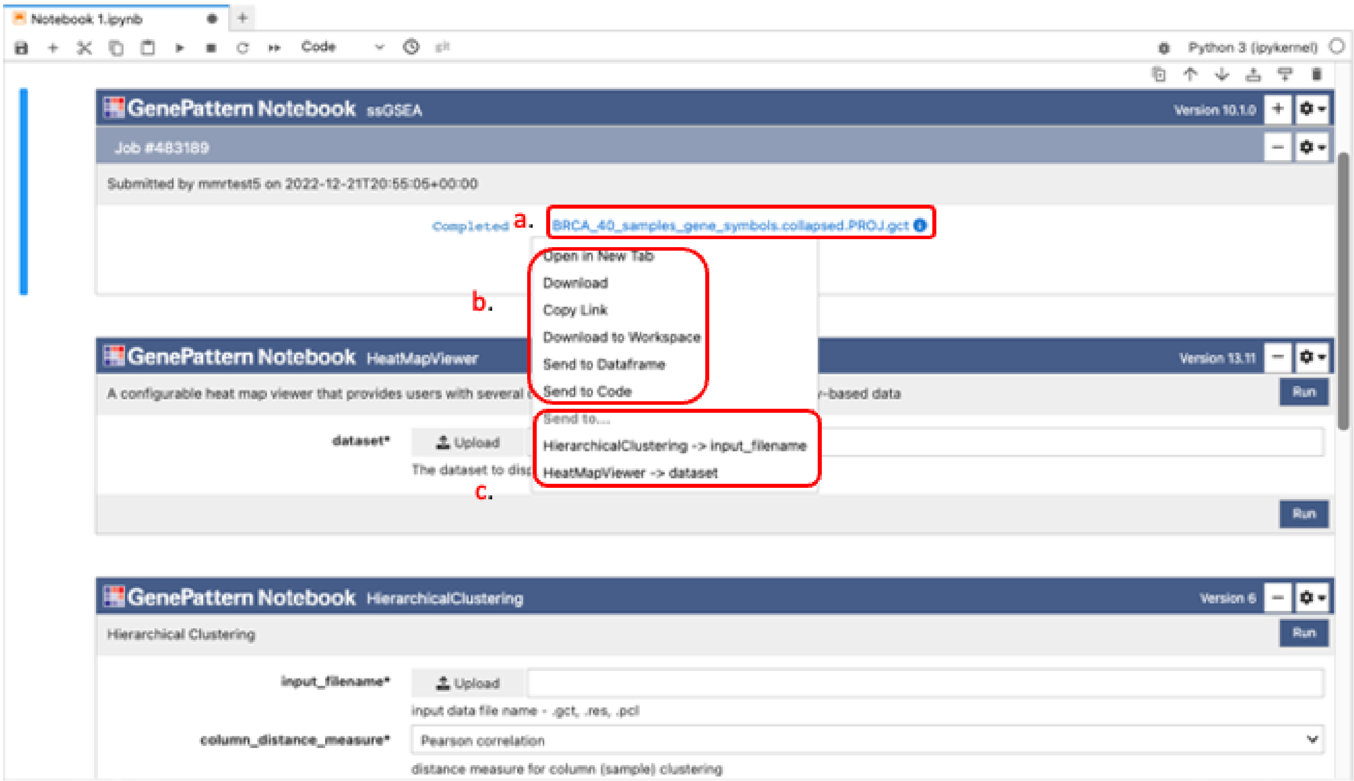
Input Type

Input Type*: Separate barcode and cDNA reads

RNA-Seq FASTQ/FASTA file, Barcode reads*: Upload https://zenodo.org/record/3862793/files/SRR891268_chr22_enriched_R2.fastq.gz

RNA-Seq FASTQ/FASTA file, cDNA reads*: Upload https://zenodo.org/record/3862793/files/SRR891268_chr22_enriched_R2.fastq.gz

Figure 1: g2nb user interface showing a Galaxy analysis cell. A tools panel (a) provides tabs that display all available analysis tools, Galaxy histories, files, and other data. In this example, the analysis tools tab is selected and the user (b) searches for the STAR sequence alignment tool. Tools with STAR in their name or description are displayed for the two servers currently connected: (c) the Galaxy Main server and (d) the GenePattern cloud server. (e) A Galaxy analysis cell shows the interface to the STARsolo tool on Galaxy Main after it has been selected by the g2nb user. Input files are selected as they would in the original platforms. Figures. S1 – S2 show IGV and Cytoscape in their g2nb analysis cell formats.



Submitted by mmrtest5 on 2022-12-21T20:55:05+00:00

A configurable heat map viewer that provides users with several ... -based data

dataset* The dataset to disp...

Hierarchical Clustering

input_filename*
input data file name - .gct, .res, .pcl

column_distance_measure* Pearson correlation
distance measure for column (sample) clustering

Figure 2: Features for working with result files. A g2nb notebook is displayed containing three GenePattern cells. The result file from the ssGSEA analysis (a) displays a context menu when clicked, showing (b) standard options for displaying and downloading the file, as well as for automatically loading the file's contents into a Pandas dataframe (this option is available for files containing data in an array format) and a Python variable (Send to Code). The menu also presents a list of the analysis cells currently in the notebook that can accept that file as an input (c). Here, the results of ssGSEA can be used as input to the HierarchicalClustering analysis cell or the HeatMapViewer visualization cell. When one of these is selected, the file appears as the input of the corresponding analysis. In each case (b-c), the result data is automatically downloaded and transferred to the appropriate destination, providing a seamless user experience. Similar functionality is available in Galaxy analysis cells, where users can choose a file from their currently selected history and send it to another Galaxy or GenePattern cell in the notebook.

org, where investigators can create and run notebooks, share them with collaborators, and publish them for general use. The workspace includes several components to help scientists use the environment: a growing library of g2nb notebooks that implement common analysis workflows that investigators can copy and use as templates for their own research. Notebooks in the collection currently support several 'omics modalities. Single-cell analysis, including preprocessing, clustering, cluster harmonization, pseudotime trajectory analysis, and RNA velocity analysis, are available via notebooks that incorporate the Seurat[9], Conos[10], STREAM[11], and scVelo[12] tools; a number of gene set enrichment analysis modalities including "classic" two-phenotype GSEA[13], single-sample GSEA[14]; workflows for ATAC-seq and proteogenomics analyses, and general machine learning methods for clustering, classification, and dimension reduction.

To address the frequent problem of notebooks having incompatible dependencies, g2nb provides project spaces. Each project space provides a separate context that contains its own notebooks, packages, libraries, and files. Investigators can share projects with collaborators and publish projects on the g2nb workspace.

Software Architecture

The g2nb functionality is implemented as extensions on the JupyterLab code base. The server runs in a JupyterHub instance encapsulated in a Docker container. For Galaxy and GenePattern, we developed extensions that (1) provide login and credential authorization to running servers; (2) query a server to determine its available analyses and their parameters; (3) render analyses within notebook cells, in a user interface that closely resembles the server-based version; (4) allows a user to launch analyses on a server and monitor its progress; (5) provide links to download result files or pass them to downstream notebook cells.

For the visualization tools, g2nb takes advantage of functionality provided by those projects to run within the JupyterLab environment. The Cytoscape project has released

CyJupyter (https://github.com/cytoscape/cytoscape-jupyter-widget), a visualizer that supports a growing subset of the capabilities of the standard desktop Cytoscape application. The developers of the Integrative Genomics Viewer (IGV) have released igv-notebook (https://github.com/igvteam/igv-notebook), which packages the igv.js (https://github.com/igvteam/igv.js) JavaScript IGV browser for use in notebook environments. While these tools typically require Python knowledge to install and run in the notebook environment, g2nb makes them available as launchable tools within its tool panel (Figures. S1, S2).

The g2nb workspace runs as a JupyterHub server encapsulated in a Docker container. Anyone can run their own workspace instance simply by starting up the container via its associated bash startup script and passing in parameters as needed. The public g2nb workspace runs on AWS and is configured to launch new user containers on compute nodes in an AWS auto scaling group, networked through Docker Swarm and the JupyterHub SwarmSpawner plugin.

We extended JupyterHub to support the concept of notebook projects, which each get their own environment and directory on the file system. This is handled by our projects plugin, which integrates into the JupyterHub services API. It consists of a lightweight web service using tornado and making calls to the database via sqlalchemy.

Workspace authentication integrates with GenePattern, Google and Globus, and supports guest accounts as well. This multi-modal authentication is managed by our custom multiauthenticator plugin for JupyterHub, which then delegates to pre-existing JupyterHub authenticator plugins, such as those for Google and Globus. The GenePattern and Guest authenticator plugins we've custom implemented as well.

Conclusion

While several projects have attempted to combine notebook systems with bioinformatics platforms, g2nb is novel in that it incorporates multiple platforms within the JupyterLab interface, with functionality to seamlessly transfer data between tools and to interweave data between analysis platforms and notebook code cells. The Galaxy project previously integrated Jupyter as a tool within its web interface [15]. In this approach Jupyter notebooks are launched within the Galaxy interface. The g2nb environment is a complementary approach in which Galaxy tools are launched within the notebook interface. Many other tools and resources provide packages that allow them to be used in a Jupyter-based environment, but these are generally code libraries that do not make use of the notebook's features beyond their use in code cells.

Code Availability

The g2nb platform is available for use at g2nb.org. For those who wish to run or host the g2nb environment on a local system, it is available as the g2nb/lab and g2nb/workspace Docker containers at hub.docker.com. The source code for the g2nb architecture is available at github.com/g2nb under a BSD-style open source license.

Declarations

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Availability of data and materials: Not applicable

Competing interests

The authors declare that they have no competing interests.

Funding

The work was funded by NIH grants U24CA248457 to JPM and NIH NHGRI U24HG006620 and NIH NCI U24CA231877 to DB.

Authors' contributions

Writing, original draft, MR, TT; Writing, review & editing, MR, HT, DB, JPM; Software, TT, JJ, JL, EH, FK; Project Administration, MR, HT, JPM; Funding Acquisition, DB, JPM

Acknowledgements

We thank Jim Robinson and Dexter Pratt for helpful discussions.

Authors' Information (optional)

Not applicable

References

1. Kluyver T, Ragan-Kelley B, Pérez F, et al. Jupyter Notebooks–a publishing format for reproducible computational workflows. InPositioning and power in academic publishing: Players, agents and agendas. IOS press (2016): 87-90.
2. Bisong E, JupyterLab Notebooks. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress (2019): 49-57.
3. Afgan E, Baker D, Batut B, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res 46 (2018): W537-44.

4. Reich M, Liefeld T, Gould J, et al. GenePattern 2.0. Nat Genet 38 (2006): 500-1.

5. Reich M, Tabor T, Liefeld T, et al. The GenePattern notebook environment. Cell Syst 5 (2017): 149-151.

6. Shannon P, Markiel A, Ozier O, et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13 (2003): 2498-504.

7. Robinson JT, Thorvaldsdóttir H, Turner D, et al. igv.js: an embeddable JavaScript implementation of the Integrative Genomics Viewer (IGV). Bioinformatics 39 (2023): btac830.

8. Foster I, Kesselman C. Globus: A metacomputing infrastructure toolkit. The International Journal of Supercomputer Applications and High Performance Computing 11 (1997): 115-128.

9. Hao Y, Hao S, Andersen-Nissen E, et al. Integrated analysis of multimodal single-cell data. Cell 184 (2021): 3573-3587.

10. Barkas N, Petukhov V, Nikolaeva D, et al. Joint analysis of heterogeneous single-cell RNA-seq dataset collections. Nat Methods 16 (2019): 695-8.

11. Chen H, Albergante L, Hsu JY, et al. Single-cell trajectories reconstruction, exploration and mapping of omics data with STREAM. Nat commun 10 (2019): 1-4.

12. Bergen V, Lange M, Peidli S, et al. Generalizing RNA velocity to transient cell states through dynamical modeling. Nat Biotechnol 38 (2020):1408-1414.

13. Subramanian A, Tamayo P, Mootha VK, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A 102 (2005): 15545-50.

14. Barbie DA, Tamayo P, Boehm JS, et al. Systematic RNA interference reveals that oncogenic KRAS-driven cancers require TBK1. Nature 462 (2009): 108-12.

15. Grüning BA, Rasche E, Rebolledo-Jaramillo B, et al. Jupyter and Galaxy: Easing entry barriers into complex data analyses for biomedical researchers. PLoS Computational Biol 13 (2017): e1005425.

16. Pratt D, Chen J, Welker D, et al. NDEx, the network data exchange. Cell Syst 1 (2015): 302-305.

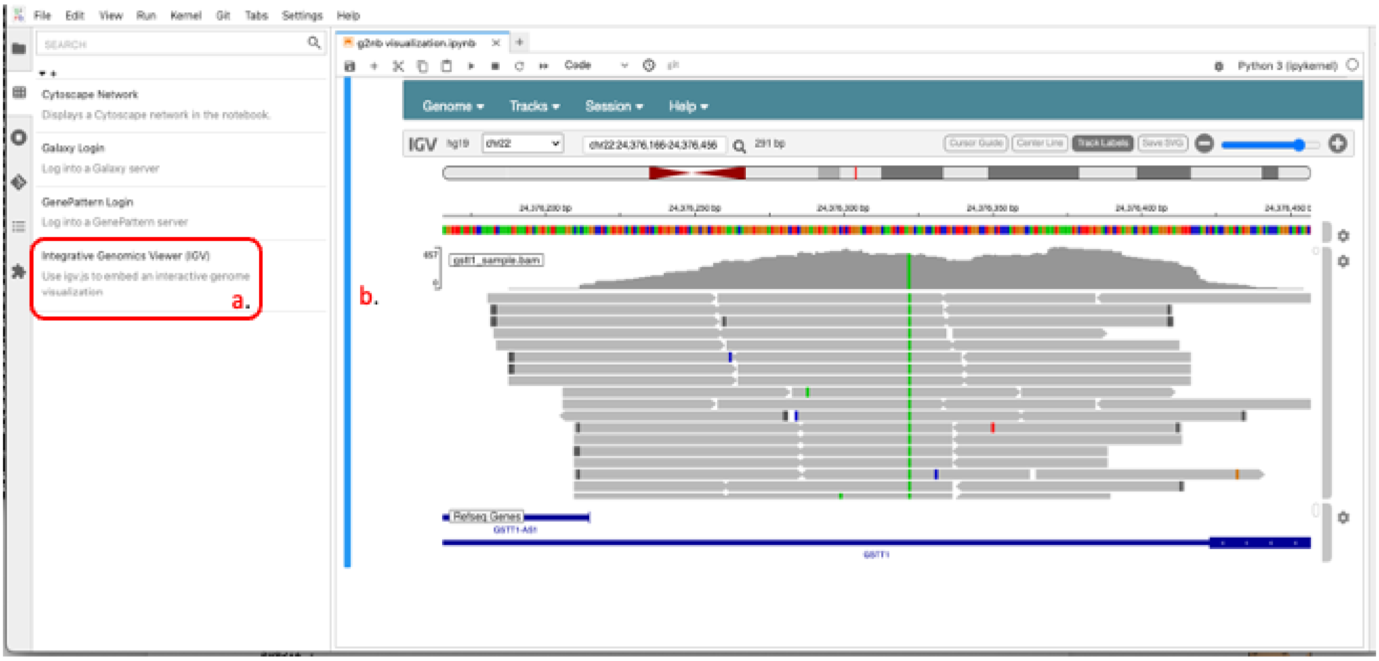
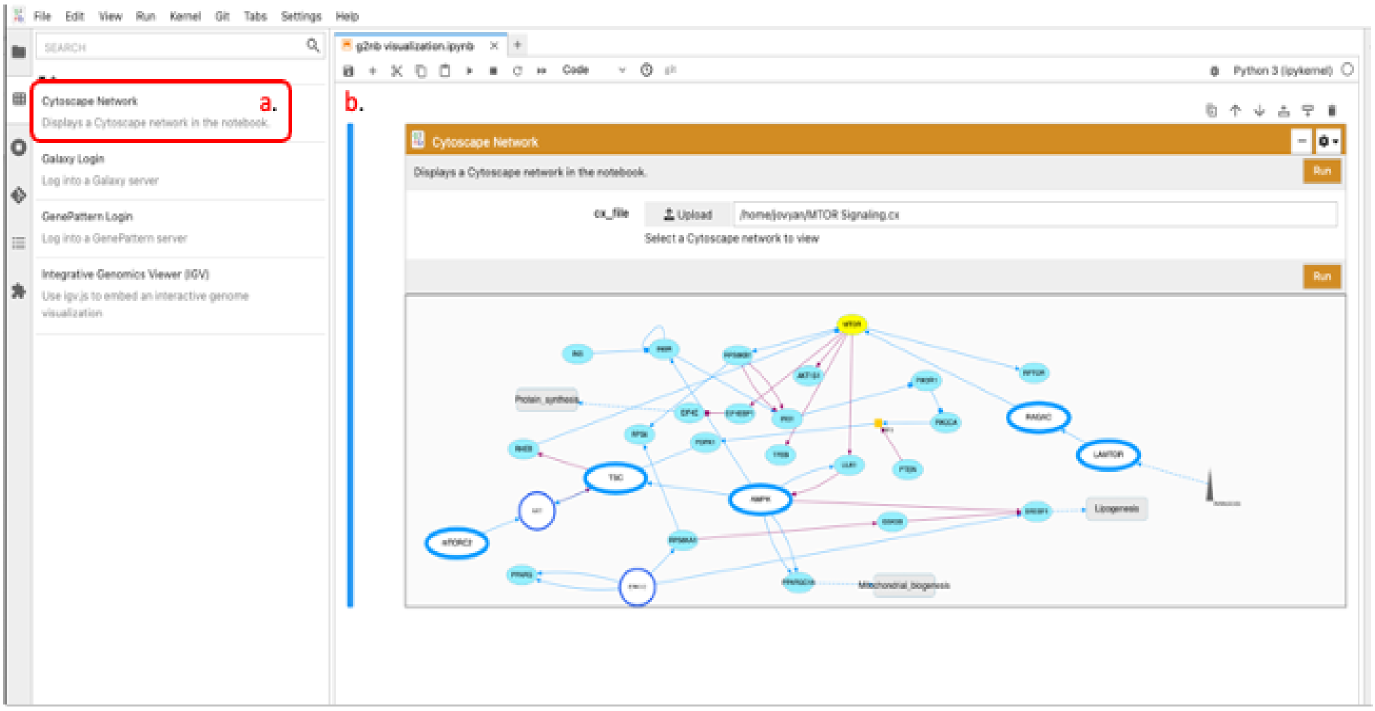

File Edit View Run Kernel Git Tabs Settings Help
SEARCH
Cytoscape Network
Displays a Cytoscape network in the notebook.
Galaxy Login
Log into a Galaxy server
GenePattern Login
Log into a GenePattern server
Integrative Genomics Viewer (IGV)
Use igv.js to embed an interactive genome visualization
a.
g2nb visualization.ipynb
Code
Python 3 (ipykernel)
b.
Genome ▾ Tracks ▾ Session ▾ Help ▾
IGV hg19 chr22
chr22:24,376,166-24,376,456
291 bp
Cursor Guide Center Line Track Labels Save SVG
gstt1_sample.bam
Refseq Genes
GSTT1-AS1
GSTT1

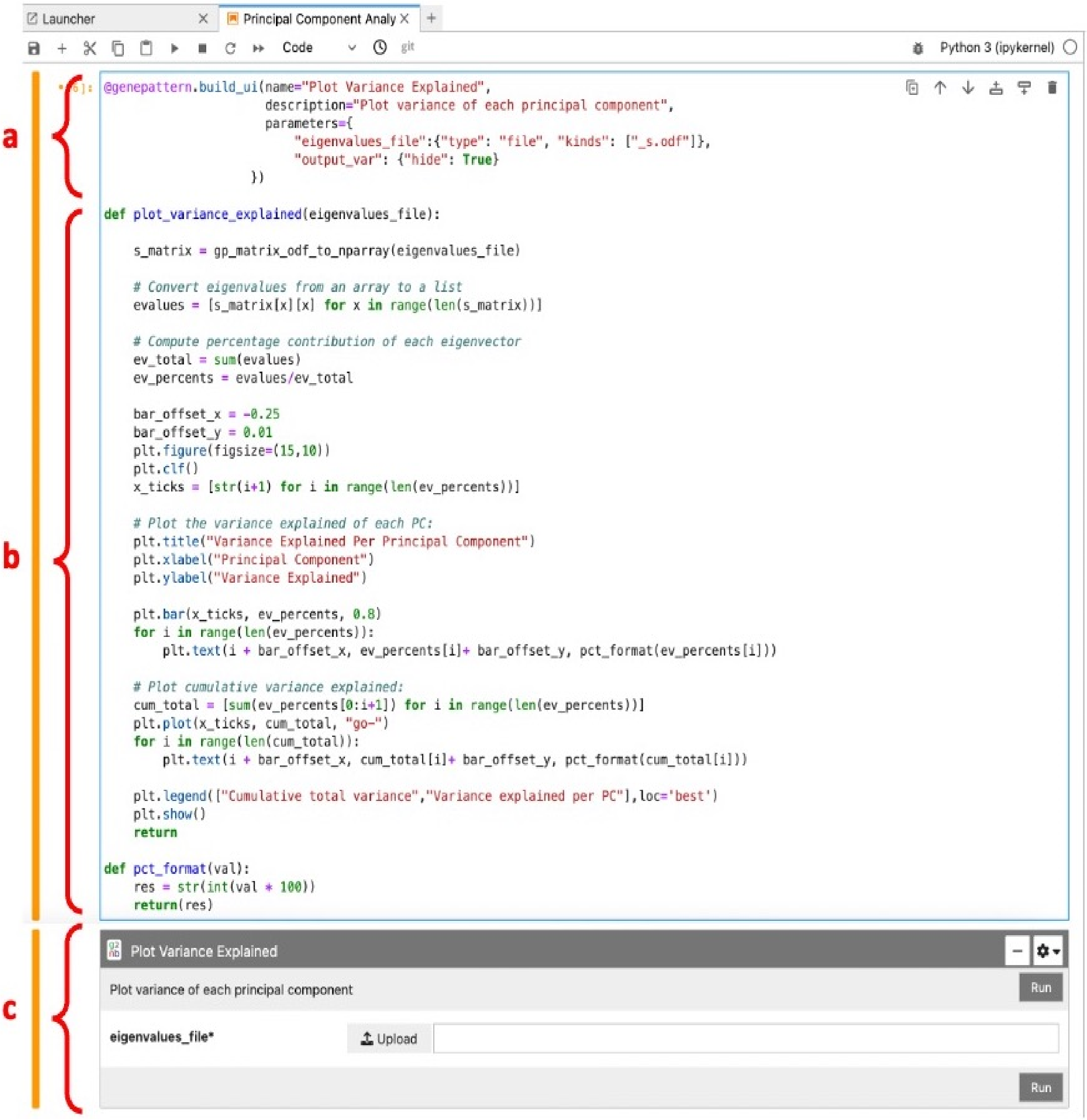
Figure S1: IGV in g2nb. To insert an IGV cell into a g2nb notebook, a user (a) selects the Integrative Genomics Viewer tool in the toolbar. After the user specifies the input data, (b) the IGV viewer is displayed as a notebook cell, with all of IGV's interactive capabilities, including zooming, panning, multi-locus view, searching by gene name or genomic locus, etc. Here IGV is displaying aligned reads from whole genome sequencing, with the view zoomed in to gene GSTT1. The green color represents mismatches between the data and the reference genome, clearly indicating a SNP in the sequenced sample.


File Edit View Run Kernel Git Tabs Settings Help
SEARCH
Cytoscape Network
Displays a Cytoscape network in the notebook.
a.
Galaxy Login
Log into a Galaxy server
GenePattern Login
Log into a GenePattern server
Integrative Genomics Viewer (IGV)
Use igv.js to embed an interactive genome visualization
g2nb visualization.ipynb
Code
Python 3 (ipykernel)
b.
Cytoscape Network
Displays a Cytoscape network in the notebook.
Run
Upload
Select a Cytoscape network to view
Run
MTOR
Protein_synthesis
TSC
AMPK
RAGAC
LAMTOR
Lipogenesis
Mitochondrial_biogenesis
PTEN

cx_file /home/jovyan/MTOR Signaling.cx

Figure S2: Cytoscape interface in g2nb. The g2nb environment uses the JavaScript-based Cy-JupyterLab package to implement a subset of available Cytoscape functionality. The user (a) selects the Cytoscape tool in the toolbar. After the user specifies the input data, the Cy-JupyterLab interface (b) is displayed as a notebook cell, with capabilities including zooming, panning, manipulation of nodes, and other functionality as described in the GitHub repository at https://github.com/cytoscape/cy-jupyterlab. Here the Cytoscape tool is displaying the MTOR signaling pathway as retrieved from the NDEx[18] resource (https://www.ndexbio.org/viewer/networks/34540ca5-1e5f-11e8-b939-0ac135e8bacf)

Citation: Michael Reich, Thorin Tabor, John Liefeld, Jayadev Joshi, Forrest Kim, Edwin Huang, Helga Thorvaldsdottir, Daniel Blankenberg, Jill P Mesirov. Genomics to Notebook (g2nb): Extending the Electronic Notebook to Address the Challenges of Bioinformatics Analysis. Journal of Bioinformatics and Systems Biology. 8 (2025): 01-07.



```
@genepattern.build_ui(name="Plot Variance Explained",
                      description="Plot variance of each principal component",
                      parameters={
                          "eigenvalues_file":{"type": "file", "kinds": ["_s.odf"]},
                          "output_var": {"hide": True}
                    })

def plot_variance_explained(eigenvalues_file):

    s_matrix = gp_matrix_odf_to_nparray(eigenvalues_file)

    # Convert eigenvalues from an array to a list
    evalues = [s_matrix[x][x] for x in range(len(s_matrix))]

    # Compute percentage contribution of each eigenvector
    ev_total = sum(evalues)
    ev_percents = evalues/ev_total

    bar_offset_x = -0.25
    bar_offset_y = 0.01
    plt.figure(figsize=(15,10))
    plt.clf()
    x_ticks = [str(i+1) for i in range(len(ev_percents))]

    # Plot the variance explained of each PC:
    plt.title("Variance Explained Per Principal Component")
    plt.xlabel("Principal Component")
    plt.ylabel("Variance Explained")

    plt.bar(x_ticks, ev_percents, 0.8)
    for i in range(len(ev_percents)):
        plt.text(i + bar_offset_x, ev_percents[i]+ bar_offset_y, pct_format(ev_percents[i]))

    # Plot cumulative variance explained:
    cum_total = [sum(ev_percents[0:i+1]) for i in range(len(ev_percents))]
    plt.plot(x_ticks, cum_total, "go-")
    for i in range(len(cum_total)):
        plt.text(i + bar_offset_x, cum_total[i]+ bar_offset_y, pct_format(cum_total[i]))

    plt.legend(["Cumulative total variance","Variance explained per PC"],loc='best')
    plt.show()
    return

def pct_format(val):
    res = str(int(val * 100))
    return(res)
```

Plot Variance Explained

Plot variance of each principal component

eigenvalues_file*

Figure S34: User Interface Builder. The User Interface Builder (UI Builder) provides a highly customizable interface to Python functions, allowing notebook authors to present a clean form-like interface that exposes only the parameters that the user needs to input. Below, an example shows a code cell containing (a) a Python decorator that allows the notebook author to specify the appearance of the UI Builder cell, including which input parameters of the function are displayed, the type of parameter (text, file, drop-down list, etc.), parameter descriptions, and many other details. The use of a Python decorator allows the actual Python function (b) to be used without modification. When the code cell is executed, it is replaced by the user interface specified in the Python decorator as an input form (c). Here, the code cell requests a file of eigenvalues that will be displayed as a graph of variance explained as part of a notebook that computes and visualizes principal components of a dataset. The code can be shown or hidden by clicking the gear icon in (c) and choosing Toggle Code View. Updates to the code are automatically incorporated into the UI Builder display when the code cell is executed. Complete documentation is available at https://docs.g2nb.org/en/latest/programmatic/#ui-builder, and a UI Builder tutorial notebook is available at https://workspace.g2nb.org/hub/preview?id=37.