Bioinformatics, 38(8), 2022, 2348–2349
https://doi.org/10.1093/bioinformatics/btac109
Advance Access Publication Date: 18 February 2022
Applications Note

Genome analysis

Integrated Genome Browser App Store

Sameer Shanbhag, Riddhi Patil, Noor Zahara, Chirag Shetty, Rachel Weidenhammer, Sneha Watharkar, Pranav Tambvekar, Philip P. Badzuh, Chester Dias, Narendra Vankayala, Prutha Kulkarni, Charan Vallapureddy, Shamika Kulkarni, Pooja Nikhare, Nowlan H. Freese and Ann E. Loraine *

Department of Bioinformatics and Genomics, University of North Carolina at Charlotte, Charlotte, NC 28223, USA

*To whom correspondence should be addressed.

Associate Editor: Tobias Marschall

Received on August 30, 2021; revised on December 11, 2021; editorial decision on February 15, 2022; accepted on February 16, 2022

Abstract

Summary: Rapid progress in genome science requires equally rapid visualization software development so that researchers can better explore and understand novel datasets. To make developing new visualizations faster and easier, we previously re-factored the Integrated Genome Browser (IGB), a desktop Java application with dozens of features, into a pluggable application framework that can accept new functionality as plug-ins, called IGB Apps. However, developers lacked a centralized location for sharing Apps, making it hard to connect with potential users. To fill this gap, we created an App Store for IGB, a user-friendly Web site for developers to release and document Apps, and for users to find them.

Availability and implementation: The IGB App Store is available from https://bioviz.org.

Contact: aloraine@uncc.edu

1 Introduction

Integrated Genome Browser (IGB) is a fast, flexible, free and open-source genome browser implemented in Java that runs natively on users' personal computers (Nicol *et al.*, 2009). IGB is best known for its responsiveness and fast animated zooming, which evoke the feeling of effortless flight through a genomic data scene. IGB is also highly interactive, offering many ways to manipulate and transform data, from selecting and inspecting individual data items to performing multi-step visual analytics operations on entire datasets, separated into tracks.

Since its first release in the early 2000s, a small community of developers and users has continuously maintained and improved IGB, transforming it into an extensible, pluggable framework based on OSGi (Freese *et al.*, 2016), an open standard for building modular programs in Java. Using this new framework, we and others have developed plug-ins, called 'IGB Apps', that add new visual analytics operations, new menu items or entirely new windows to IGB (Céol *et al.*, 2016, 2020; Céol and Müller, 2015; Mall *et al.*, 2016).

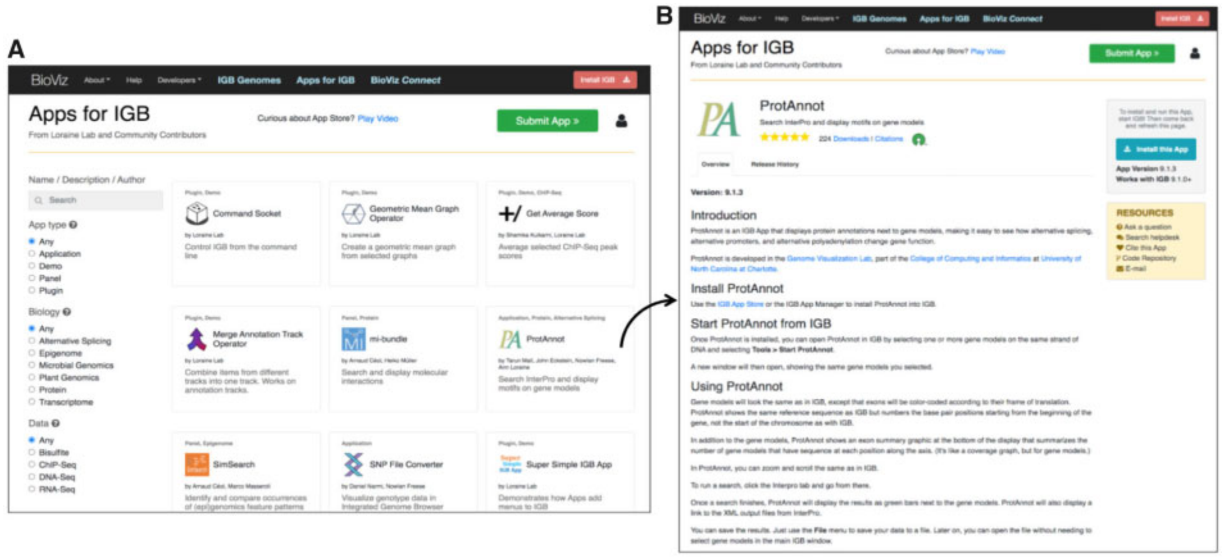
However, until now, developers had to deploy and distribute their Apps themselves. They needed to set up a Web site to host their App and then advertise the URL to potential users, who had to enter the URL manually into IGB. This was inconvenient, reducing enthusiasm for the IGB platform among developers and users alike.

To make developing and releasing IGB Apps easier, we created and deployed an App Store for IGB, a Web site where developers can upload and publicize their Apps and users can find them. In this article, we summarize key features of the App Store, including how the App Store, a Web application, communicates with IGB, a native software that runs outside a Web browser.

2 Results

2.1 Submitting an App

To submit a new App or release a new version of an existing App, a developer first logs in to the App Store via social sign-in (currently, only social sign-in with Google is supported). Next, the developer uploads their App file, implemented as a java archive (jar) file, called a 'bundle' in OSGi parlance. The site unpacks the file and checks for compatibility with IGB, displaying a summary report that invites the developer to either submit the App file as-is or replace it with a new version. If the compatibility check passes, the developer can click a button to submit the App, which deploys the jar file to a private testing site which site managers use to install the App into IGB and try it out, manually checking App code and behavior for bugs or security-related red flags, such as sending data to external sites without a user's knowledge. Managers then either release the App as-is or contact the developer with feedback if changes are needed. Developers and site managers receive email notifications when Apps are submitted or released for public use. Once an App is released, it receives a dedicated home page, which the developer can edit and customize, adding other users as editors. They can also release new versions without further manual review by site maintainers.



Apps for IGB
From Loraine Lab and Community Contributors
Curious about App Store? Play Video

Name / Description / Author
Search
App type: Any, Application, Demo, Panel, Plugin
Biology: Any, Alternative Splicing, Epigenome, Microbial Genomics, Plant Genomics, Protein, Transcriptome
Data: Any, Bisulfite, ChIP-Seq, DNA-Seq, RNA-Seq

Command Socket, by Loraine Lab: Control IGB from the command line
Geometric Mean Graph Operator, by Loraine Lab: Create a geometric mean graph from selected graphs
Get Average Score, by Shamika Kulkarni, Loraine Lab: Average selected ChIP-Seq peak scores
Merge Annotation Track Operator, by Loraine Lab: Combine items from different tracks into one track. Works on annotation tracks.
mi-bundle, by Arnaud Céol, Heiko Müller: Search and display molecular interactions
ProtAnnot, by Tarun Mall, John Eckstein, Nowlan Freese, Ann Loraine: Search InterPro and display motifs on gene models
SimSearch, by Arnaud Céol, Marco Masseroli: Identify and compare occurrences of (epi)genomics feature patterns
SNP File Converter, by Daniel Narmi, Nowlan Freese: Visualize genotype data in Integrated Genome Browser
Super Simple IGB App, by Loraine Lab: Demonstrates how Apps add menus to IGB

ProtAnnot
Search InterPro and display motifs on gene models
224 Downloads | Citations
Overview | Release History
To install and run this App, start IGB. Then come back and refresh this page.
Install this App
App Version 9.1.3
Works with IGB 9.1.0+
RESOURCES: Ask a question; Search helpdesk; Cite this App; Code Repository; E-mail

Version: 9.1.3

Introduction

ProtAnnot is an IGB App that displays protein annotations next to gene models, making it easy to see how alternative splicing, alternative promoters, and alternative polyadenylation change gene function.

ProtAnnot is developed in the Genome Visualization Lab, part of the College of Computing and Informatics at University of North Carolina at Charlotte.

Install ProtAnnot

Use the IGB App Store or the IGB App Manager to install ProtAnnot into IGB.

Start ProtAnnot from IGB

Once ProtAnnot is installed, you can open ProtAnnot in IGB by selecting one or more gene models on the same strand of DNA and selecting **Tools > Start ProtAnnot**.

A new window will then open, showing the same gene models you selected.

Using ProtAnnot

Gene models will look the same as in IGB, except that exons will be color-coded according to their frame of translation. ProtAnnot shows the same reference sequence as IGB but numbers the base pair positions starting from the beginning of the gene, not the start of the chromosome as with IGB.

In addition to the gene models, ProtAnnot shows an exon summary graphic at the bottom of the display that summarizes the number of gene models that have sequence at each position along the axis. (It's like a coverage graph, but for gene models.)

In ProtAnnot, you can zoom and scroll the same as in IGB.

To run a search, click the Interpro tab and go from there.

Once a search finishes, ProtAnnot will display the results as green bars next to the gene models. ProtAnnot will also display a link to the XML output files from InterPro.

You can save the results. Just use the **File** menu to save your data to a file. Later on, you can open the file without needing to select gene models in the main IGB window.

Fig. 1. IGB App Store interface. (**A**) App Store home page. (**B**) Home page for ProtAnnot, an IGB App

2.2 Installing an App

Finding and installing Apps into IGB is similarly straightforward. The App Store home page shows a grid of App tiles, sorted alphabetically by App name (Fig. 1A), which users can search using a free text query form or a faceted search interface, which filters Apps by categories created and curated by site maintainers. Future App Store releases will likely incorporate categories from systems, such as the EDAM ontology of bioscientific data analysis and data management (Ison *et al.*, 2013), further improving App dissemination and impact.

Clicking an App tile opens the App's home page, which describes App functionality and allows users to install an App into IGB. If IGB is running, and if the App has not yet been installed, the page displays a button labeled with 'Install the App'. If the App is already installed, the button displays the text 'Installed', and if it is installed but a newer version is available, the button displays the text 'Update the App'. Once an App is installed or updated, the button label changes to 'Installed'. If IGB is not running, the button displays a message advising the user to start the IGB program and links to a page where the user can download and install IGB. Users can also install or update Apps within the IGB interface, using the IGB App Manager window, which resembles the App Store interface, making both easier to learn and remember between sessions.

3 Materials and methods

We implemented IGB App Store using the Django Web application framework and starting with a fork of the open-source Cytoscape App Store code base (Lotia *et al.*, 2013). We customized and updated the code base extensively, describing two major changes here. First, we re-designed the home page, blending and improving on user interface conventions found across many app store-like applications. For example, we improved on the Grafana Plug-ins Web site faceted search interface by adding tooltips defining the curated categories used. Second, we re-factored the App Store to function as an OSGi Bundle Repository (OBR), conforming to the Apache Felix project OBR specification (Apache Foundation, 2021). On startup, or when a user adds a new App Store, IGB retrieves an OBR-compliant XML document listing the App Store's available Apps, their versions and their IGB platform requirements, ensuring that IGB installs compatible Apps only. When users visit an App home page, Javascript running in the Web page contacts a localhost REST endpoint within IGB to determine App status, modifying the Web page's 'Install this App' button accordingly (Fig. 1B).

The App Store is open source, and the full commit history of the project is available from https://bitbucket.org/lorainelab/appstore. Ansible playbooks used to deploy and update App Store on Amazon Web Service infrastructure are available from https://bitbucket.org/lorainelab/appstore-playbooks.

Other groups are welcome to deploy their own App Stores, or re-use the code to launch their own projects, with the caveat that supporting an application other than IGB would require modifying the user interface, along with the JavaScript code controlling the App page 'Install the App' button, code that depends on interacting with IGB itself via its localhost endpoint.

Acknowledgement

We thank the Cytoscape development team for sharing Cytoscape App Store source code, which helped us develop early versions of the IGB App Store.

Funding

This work was supported by United States National Institutes of Health grant numbers [R01GM103463 and R35GM139609].

Conflict of Interest: none declared.

References

Apache Foundation. (2021) Apache *Felix OSGi Bundle Repository (OBR)*. https://felix.apache.org/documentation/subprojects/apache-felix-osgi-bundle-repository.html (22 February 2022, date last accessed).

Céol,A. *et al.* (2020) Search and comparison of (epi)genomic feature patterns in multiple genome browser tracks. *BMC Bioinformatics*, **21**, 464.

Céol,A. and Müller,H. (2015) The MI bundle: enabling network and structural biology in genome visualization tools. *Bioinformatics*, **31**, 3679–3681.

Céol,A. *et al.* (2016) Genome and network visualization facilitates the analyses of the effects of drugs and mutations on protein-protein and drug-protein networks. *BMC Bioinformatics*, **17** (**Suppl. 4**), 54.

Freese,N.H. *et al.* (2016) Integrated Genome Browser: visual analytics platform for genomics. *Bioinformatics*, **32**, 2089–2095.

Ison,J. *et al.* (2013) EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats. *Bioinformatics*, **29**, 1325–1332.

Lotia,S. *et al.* (2013) Cytoscape app store. *Bioinformatics*, **29**, 1350–1351.

Mall,T. *et al.* (2016) ProtAnnot: an App for Integrated Genome Browser to display how alternative splicing and transcription affect proteins. *Bioinformatics*, **32**, 2499–2501.

Nicol,J.W. *et al.* (2009) The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics*, **25**, 2730–2731.