SOFTWARE TOOL ARTICLE

REVISED MetaGenSense: A web-application for analysis and exploration of high throughput sequencing metagenomic data [version 3; peer review: 1 approved, 2 approved with reservations]

Previously titled: MetaGenSense: A web application for analysis and visualization of high throughput sequencing metagenomic data

Damien Correia^{1*}, Olivia Doppelt-Azeroual^{2*}, Jean-Baptiste Denis³, Mathias Vandenbergert¹, Valérie Caro¹

¹Pôle Génomique des Pathogènes, Unité Environnement et Risques Infectieux, Institut Pasteur, F-75724, Paris, France

²Centre de Bioinformatique, Biostatistique et Biologie Intégrative (C3BI, USR 3756 Institut Pasteur et CNRS), Institut Pasteur, F-75724, Paris, France

³Groupe Exploitation et Infrastructure, Institut Pasteur, F-75724, Paris, France

* Equal contributors

V3 First published: 02 Apr 2015, 4:86
<https://doi.org/10.12688/f1000research.6139.1>
 Second version: 22 Aug 2016, 4:86
<https://doi.org/10.12688/f1000research.6139.2>
 Latest published: 01 Dec 2016, 4:86
<https://doi.org/10.12688/f1000research.6139.3>

Abstract

The detection and characterization of emerging infectious agents has been a continuing public health concern. High Throughput Sequencing (HTS) or Next-Generation Sequencing (NGS) technologies have proven to be promising approaches for efficient and unbiased detection of pathogens in complex biological samples, providing access to comprehensive analyses. As NGS approaches typically yield millions of putatively representative reads per sample, efficient data management and visualization resources have become mandatory. Most usually, those resources are implemented through a dedicated Laboratory Information Management System (LIMS), solely to provide perspective regarding the available information. We developed an easily deployable web-interface, facilitating management and bioinformatics analysis of metagenomics data-samples. It was engineered to run associated and dedicated Galaxy workflows for the detection and eventually classification of pathogens. The web application allows easy interaction with existing Galaxy metagenomic workflows, facilitates the organization, exploration and aggregation of the most relevant sample-specific sequences among

Open Peer Review

Approval Status ? ? ✓

	1	2	3
version 3			
(revision)			✓
01 Dec 2016			↑
version 2		?	✓
(revision)		view	view
22 Aug 2016		↑	↑
version 1	?	✗	?
02 Apr 2015	view	view	view

1. **Peter Li**, GigaScience, Hong Kong SAR, China
2. **Manuel Corpas** , Future Business Centre, Cambridge, UK
Benedikt Rauscher, The Genome Analysis Center, Norwich, UK

millions of genomic sequences, allowing them to determine their relative abundance, and associate them to the most closely related organism or pathogen.

The user-friendly Django-Based interface, associates the users' input data and its metadata through a bio-IT provided set of resources (a Galaxy instance, and both sufficient storage and grid computing power). Galaxy is used to handle and analyze the user's input data from loading, indexing, mapping, assembly and DB-searches. Interaction between our application and Galaxy is ensured by the BioBlend library, which gives API-based access to Galaxy's main features. Metadata about samples, runs, as well as the workflow results are stored in the LIMS. For metagenomic classification and exploration purposes, we show, as a proof of concept, that integration of intuitive exploratory tools, like Krona for representation of taxonomic classification, can be achieved very easily. In the trend of Galaxy, the interface enables the sharing of scientific results to fellow team members.

Keywords

High Throughput Sequencing , Next-Generation Sequencing , Laboratory Information Management System , Galaxy , Django

This article is included in the [Galaxy](#) gateway.

3. **Saskia Hiltmann** , Erasmus University

Medical Center, Rotterdam, The Netherlands

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Olivia Doppelt-Azeroual (olivia.doppelt@pasteur.fr)

Competing interests: No competing interests were disclosed.

Grant information: Damien Correia and Olivia Doppelt-Azeroual were financed by the "Commissariat à l'Energie Atomique et aux Energies Alternatives" in the scope of a national anti-terrorism fight NRBC project (from October 2011 to April 2014). *The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.*

Copyright: © 2016 Correia D *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Correia D, Doppelt-Azeroual O, Denis JB *et al.* **MetaGenSense: A web-application for analysis and exploration of high throughput sequencing metagenomic data [version 3; peer review: 1 approved, 2 approved with reservations]** F1000Research 2016, 4:86 <https://doi.org/10.12688/f1000research.6139.3>

First published: 02 Apr 2015, 4:86 <https://doi.org/10.12688/f1000research.6139.1>

REVISED Amendments from Version 2

The difference between version 2 and version 3 of this article is [Figure 2](#). It was slightly modified and built in a better resolution.

[See referee reports](#)

Introduction

Background HTS & metagenomics

The detection and characterization of emerging infectious agents has been a continuing public health concern. High Throughput Sequencing (HTS; or Next-Generation Sequencing, NGS) technologies have proven to be efficient at providing access to comprehensive analyses and unbiased detection of pathogens in complex biological samples.

Most large-scale genomic (re)sequencing projects involve both sequencing technology, computational analyses, and genotyping expertise. Current NGS platforms including Illumina, Ion Torrent/Life Technologies, Pacific Bioscience and Nanopore can generate reads of 100–10,000 bases long allowing better coverage of the genome at lower cost. However, these platforms also generate huge amounts of raw data.

Besides those sequence files, it has become important to also consider and store associated sample related metadata (collection date, location, ...). In addition, NGS projects usually represent such a huge amount of relevant sample-specific sequences that efficient data management and visualization resources have become mandatory. Those challenges accompanying HTS technologies have raised the following fundamental questions: (1) How do we best manage the enormous amount of sequencing data? (2) What are the most appropriate choices among the available computational methods and analysis tools? The issue concerning the growing amount of data can be managed through a dedicated *Laboratory Information Management System* (LIMS), solely to organize and plan their analysis, and to provide perspective. The question regarding the lack of adapted intertwining among the wide spectrum of available tools was in part filled by workflow management systems, even though it still requires fairly advanced knowledge of the tools available at hand.

Indeed, today, hundreds of bioinformatics tools are available, each demanding specific parameterization¹. NGS data analysts have been designing workflows in order to automate complex processing pipelines, both possibly using existing codes, avoiding rewriting software, and supporting parallel, distributed computations. These features make the workflows relatively simple to construct, make them easily reusable, hence aid reproducibility. Modern workflow managers, such as Galaxy^{2–4}, offer the possibility of sharing them with others. As such, Galaxy as a data analysis and workflow management system, provides biologists with a hands-on toolbox to build multi-step computational workflows for data-processing, quality control, and analytic results aggregation, while additionally ensuring analysis reproducibility. As a prerequisite to a system for composing pipelines for large-scale analyses, there is a need for

an adapted and up-to-scale computational infrastructure capable of doing the processing and data storage.

We therefore developed MetaGenSense, a bioinformatics web-application framework whose principle purpose is to ease the scientists' work in management of NGS-project related data and analysis results. MetaGenSense is built on three fundamental components, two of which are specific to the project: a dedicated LIMS and a Django-based web user-interface. The third component is Galaxy, which is the main bioinformatics workflow management system. In the following paragraphs, we describe the interface's implementation and display how communication between the different parts takes place, in a smooth and user-friendly, managing web-user interface.

Software tool - implementation

MetaGenSense global description

MetaGenSense is a managing and analytical bioinformatics framework that is engineered to run dedicated Galaxy workflows for the detection and eventually classification of pathogens. It aims to facilitate large-scale genomic analysis for experts in sequencing among project partners. The web application was built and can be deployed in order to facilitate access to high throughput sequencing analysis tools, acting as an information resource for the project and interacting research partners. With its user-friendly interface, it was designed to take advantage of bio-IT provider resources (a local Galaxy instance, sufficient storage and grid computing power), for analysis of input data and its metadata. Also in MetaGenSense, a dedicated LIMS (postgresql-based) was implemented to ensure data coherence. The web interface design is based on the Django web framework (<http://www.djangoproject.com>). Moreover, the communication with Galaxy is covered by the Bioblend library⁵ which provides a high-level interface for interacting with the Galaxy application, promoting faster interaction, and facilitating reuse and sharing of scripts.

The use of the available Galaxy tools and workflows is automated and seamless with MetaGenSense. Galaxy, as a pipeline management software, lets you define workflows and “pushes” the data through that pipeline. The pipeline manager ensures that all the tools in the pipeline run successfully, generally spreading the workload over a computational cluster. For example, MetaGenSense is used at the Pasteur Institute to do the bulk of the data processing for a large number of HTS projects, and can be adapted to launch any set of instructions stitched together in a dedicated workflow available in the Galaxy workflow designer interface.

A dedicated LIMS

A LIMS can be described as a software-based laboratory that offers a set of key features that support modern laboratory operations. Those systems have become mandatory to manage the quantity of metadata related to both raw data and their analysis results, obtained through bioinformatic tools. In MetaGenSense, the LIMS is based on a postgresql database. It was designed and structured with expert knowledge from biologists and bioinformaticians. Its main feature is that it was designed to store note-worthy and shareable information resulting from analysis, as well as the applied workflow-related parametrization. The database's schema is available in the [Supplementary Figure 1](#).

An intuitive Django-based web user interface

Django is a high-level Python Web-framework. It provides rapid development, as well as clean pragmatic design, and serves as data management and displaying backbone for a large number of websites where interaction is important. Moreover, the python language (<https://www.python.org/>) has become a reference programming language for a huge number of scientific applications.

MetaGenSense's UI is divided in 4 modules: 1) User authentication management, 2) LIMS, 3) Workflow, 4) Analysis. Each has a specific function, and the task-partitioning has been designed to allow independent evolution of each part according to the user's needs.

1. **User authentication management:** can be done either by communication with an LDAP user authentication database, or through a user management database.
2. **LIMS:** ensures the organization of the data according to the project at hand. The LIMS can be provided with sample metadata, which can be shared with selected project-members, and ensures sample traceability, which is an important component of any present-day core resource laboratory management system.
3. **Workflow:** manages (a) the connection with the Galaxy instance, (b) execution tracking (the Galaxy "user histories") (c) the data from Galaxy "libraries" (the datasets), (d) association and import of data from a data-library to a Galaxy user-history instance. (e) Execution of the selected Galaxy workflow. This module handles data storage and links the samples to the selected workflow.
4. **Analysis:** A result file can be saved in order to be shared with other users involved in the project, or can be exported using the Galaxy export functionality, for download.

Communication between MetaGenSense and Galaxy is dealt systematically, using the BioBlend API⁵, a highly dedicated and specialized python library, giving access to most Galaxy functionalities. For example, it is possible to fetch the Galaxy users, create the user's data-libraries, ... As a side-note, interaction with

the BioBlend development team was necessary for fine-tuning specific core functionalities, leading to a concomitant finishing and perfection of the tools and accompanying API. Specifically, BioBlend functionalities are used to communicate with Galaxy, as described in Figure 1.

A use-case example of MetaGenSense typically involves the following steps (cf Figure 2).

- User Authentication with LDAP or any authentication system defined by admin sys.
- Creation of a new project, with a name, a context, a short description and (most importantly) the other persons involved in the project. Please note that to include a person, they need to have logged at least once on the application.
- Supplying the LIMS database with metadata including sample information as the library sequencing protocol, supplementary run details and the path to raw data files.
- At this point, the user needs to copy their input data files to their Galaxy transfer directory. MetaGenSense detects new files that are copied within the exchange Galaxy project directory. Those data files need to be copied manually. This solution was chosen such that the right files would be deposited deliberately in the right target Galaxy directory, with correct user/owner's file permissions (typically on a UNIX file-system). In the exchange Galaxy directory, a subdirectory can be created, named after the project, and the raw data can be copied in that directory. This way, MetaGenSense detects the files that will be taken into account for use with Galaxy and analysed.
- In the MetaGenSense GUI, the "Workflows" button allows to import new files to analyse into Galaxy.
- Using a new Galaxy "history", an appropriately selected workflow, and the set of workflow input data-files, an analysis can be launched. As usual in Galaxy, the workflow status can be monitored.
- Intermediate and final results can be exported for download (using the native Galaxy tools), or they can be

Figure 1. Communication details between MetaGenSense and Galaxy using BioBlend.

Figure 2. MetaGenSense functioning starting with user connection.

saved in the LIMS *i.e.* be tagged as potentially interesting and shared with project members.

- Further exploration of workflow results can be done using, for example in a metagenomics data analysis context, Krona⁶, that enables taxonomic information to be explored.

Conclusions/Discussion

The technology evolution in molecular biology, especially in NGS, has moved biology into the big data era (consisting of handling data, computation requirements, efficient workflow design, and knowledge extraction). With this trend, the challenges faced by life scientists have been shifted from data acquisition to data management, processing, and knowledge extraction. While many studies have recognized the big data challenge, few systematically present approaches to tackle it. New findings in biological sciences usually come out of multi-step data pipelines (workflows). Galaxy is a workflow-management tool which can deal with big data. However, it is still necessary to globally optimize the data flow in an overall multi-step workflow in order to eliminate unnecessary data movement and redundant computation. On the other hand, data information traceability has become an inevitable requirement in a present-day laboratory setup, foreseeing that knowledge-embedded data and workflows are expected to be an integral part of future scientific publications.

We therefore, engineered MetaGenSense, a Django-based web interface which helps biologists, who in particular are unfamiliar with the design of Galaxy workflows, to quickly obtain analysis results from HTS sequencing projects. It uses Galaxy as a workflow management software and the BioBlend API to remotely

manage data upload, workflow execution as well as analysis of results. MetaGenSense covers data processing up to presentation of data and results in a human-readable data format. Its main advantages encompass data handling through its incorporated LIMS, user and project handling in a cooperative context, it enables data sharing without compromising data confidentiality, it features automated workflow execution, resulting altogether in decreasing the data and analysis delivery time. MetaGenSense is available as open-source from GitHub, and can be deployed very easily. For testing, and for users to be able to evaluate MetaGenSense's modularity, we added to our Github directory, a special release coupled with a virtual machine image containing, the tools needed in our metagenomic example workflow, a galaxy instance containing those tools as well as a prototyped workflow, mainly focused on metagenomic sample analysis. MetaGenSense is pre-configured on it and directly usable through a web browser. Once tested, it can be easily adapted to a variety of other NGS projects.

Software availability

Latest source code

<https://github.com/pgp-pasteur-fr/MetaGenSense>

Source code as at the time of publication

<https://github.com/pgp-pasteur-fr/MetaGenSense/releases/tag/1.0>
(Official v1.0 MetaGenSense release coupled with a virtual machine image for testing)

Archived source code as at the time of publication

DOI: [10.5281/zenodo.165107](https://doi.org/10.5281/zenodo.165107)

License

GPLv2

References

1. Ison J, Rapacki K, Ménager H, *et al.*: **Tools and data services registry: a community effort to document bioinformatics resources.** *Nucleic Acids Res.* 2016; **44**(D1): D38–D47.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
2. Goecks J, Nekrutenko A, Taylor J: **Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences.** *Genome Biol.* 2010; **11**(8): R86.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
3. Blankenberg D, Von Kuster G, Coraor N, *et al.*: **Galaxy: a web-based genome analysis tool for experimentalists.** *Curr Protoc Mol Biol.* 2010; **Chapter 19**: Unit 19.10.1–21.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
4. Giardine B, Riemer C, Hardison RC, *et al.*: **Galaxy: a platform for interactive large-scale genome analysis.** *Genome Res.* 2005; **15**(10): 1451–5.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
5. Sloggett C, Goonasekera N, Afgan E: **BioBlend: automating pipeline analyses within Galaxy and CloudMan.** *Bioinformatics.* 2013; **29**(13): 1685–1686.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
6. Ondov BD, Bergman NH, Phillippy AM: **Interactive metagenomic visualization in a Web browser.** *BMC Bioinformatics.* 2011; **12**: 385.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
7. Correia D, Doppelt-Azeroual O, Vandenbogaert M: **MetaGenSense: MetaGenSense v1.0 [Data set].** *Zenodo.* 2015.
[Data Source](#)

Open Peer Review

Current Peer Review Status: ? ? ✓

Version 3

Reviewer Report 01 December 2016

<https://doi.org/10.5256/f1000research.11103.r18134>

© 2016 Hiltemann S. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Saskia Hiltemann

Department of Bioinformatics, Erasmus University Medical Center, Rotterdam, The Netherlands

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 2

Reviewer Report 22 November 2016

<https://doi.org/10.5256/f1000research.10150.r15805>

© 2016 Hiltemann S. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Saskia Hiltemann

Department of Bioinformatics, Erasmus University Medical Center, Rotterdam, The Netherlands

With this revision, the authors have successfully addressed all my previous concerns.

The installation instructions were much improved, and I was able to set up MetaGenSense without any major difficulty. The availability of a Virtual Machine was especially convenient for quick testing/assessing of the application, and I was able to run the example pipeline on the provided test data easily.

The manuscript itself was also much improved and I am now happy to approve this manuscript. I think this is a nice example of how one can integrate a LIMS/project management system with Galaxy at the back-end for analysis.

Minor (optional) remarks:

1. Unless there is a compelling reason not to, please consider adding the requirements.txt file to the github repo, it was a little bit confusing that I had to create this file myself.
2. Your installation instructions refer to the command "manage.py migrate" but this appears not to be implemented until Django 1.7, yet your documentation lists the requirement "Django==1.6.2". Consider changing the requirement version for Django from 1.6.2 to 1.7 (I had no problem running the rest of the setup using Django 1.7) or changing this part of the installation instructions to fit with Django 1.6.2.
3. Capitalize "galaxy" in last paragraph of the manuscript ([...] the tools needed in our metagenomic example workflow, a galaxy instance containing those tools [...])
4. For the VM, it would be useful if you also provided the login credentials for the admin user (mgs_admin) for MGS/Galaxy in your README on github, for those readers who may be interested in seeing how the admin side of the application works.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 14 September 2016

<https://doi.org/10.5256/f1000research.10150.r15804>

© 2016 Corpas M. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuel Corpas

Future Business Centre, Cambridge, UK

I appreciate that there have been noticeable improvements, hence I am now able to accept this article with reservations. I hope authors find my suggestions useful.

I would have preferred if the response letter was more specific, quoting the suggestions offered by the reviewers and directly underneath the author responses showing how they have addressed them. Currently I have no easy way to check how you have addressed my previous comments. The response letter has some grammatical errors. I paste below the bits that are grammatically incorrect. I would recommend that authors have proof read their submitted materials before sending them.

- "according to reviewers remarks"

- "this application is more design"
- "Description of the software is also a more precise"
- I am unable to run the image <http://webext.pasteur.fr/metagensense/metagensense.ova> as I do not have the ability to create virtual machines. Perhaps having a set of images that describe exactly the point authors want to make regarding this virtual machine could help the review process. This seems to be done in Figure 2, although the resolution is not sufficient.
- I am unable to read the font inside of Figure 2's window screenshots. I would like to be able to see what each of them shows. This way I would be able to see how the interface works for each of the stages (e.g., Connection, Project creation, etc.).
- I went to GitHub and I went to "<http://metagensense.test.fr:8000>" but the answer I got from the server was "metagensense.test.fr refused to connect".
- In the installation notes of the README.md, the first step points to virtualenv, virtualenvwrapper. I would have appreciated if authors could provide in there how I can install virtualenv. It is just need to take a few instructions from <https://virtualenv.pypa.io/en/stable/installation/>.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 21 Nov 2016

Olivia Doppelt-Azeroual, Institut Pasteur, F-75724, Paris, France

Dear Dr. Corpas,

Thank you for your review and remarks. According to your advice, the documentation, publication and other materials have now been proofread by a colleague.

The main change since the second version of the publication is the documentation available in the github repository <http://metagensense.readthedocs.io/en/latest/?badge=latest>. We tried to clarify the following points:

- The documentation on the main github page was simplified and we clearly separated the virtual machine test documentation and the metagensense local installation procedure.
- The documentation available on the platform Read the Docs was proofread and that enabled us to emphasize each step of the installation procedure.
- We added documentation for the Virtualenv installation procedure
- The Figure 2 which was in a low resolution was updated on the github repository and

the publication.

You will find below the answers to your remarks:

[Q1]

I am unable to run the image <http://webext.pasteur.fr/metagensense/metagensense.ova> as I do not have the ability to create virtual machines. Perhaps having a set of images that describe exactly the point authors want to make regarding this virtual machine could help the review process. This seems to be done in Figure 2, although the resolution is not sufficient.

[A1]

The point of the Virtual Machine is to allow users to test the MetaGenSense application before installing it on their local infrastructure.

The online documentation available at the URL:

<http://metagensense.readthedocs.io/en/latest/connection.html> now includes a step by step user guide.

[Q2]

I am unable to read the font inside of Figure 2's window screenshots. I would like to be able to see what each of them shows. This way I would be able to see how the interface works for each of the stages (e.g., Connection, Project creation, etc.).

[A2]

The image for each stages has been uploaded with a better resolution so the readers can have a better idea of MetaGenSense's looks without installing the virtual machine.

It is available at the URL: [https://github.com/pgp-pasteur-](https://github.com/pgp-pasteur-fr/MetaGenSense/blob/master/doc/images/metagensense_complete.jpeg)

[fr/MetaGenSense/blob/master/doc/images/metagensense_complete.jpeg](https://github.com/pgp-pasteur-fr/MetaGenSense/blob/master/doc/images/metagensense_complete.jpeg)

[Q3]

I went to GitHub and I went to "<http://metagensense.test.fr:8000>" but the answer I got from the server was "metagensense.test.fr refused to connect".

[A3]

The metagensense.test.fr URL is only available when running the MetaGenSense virtual machine. The documentation available on the github repository has been corrected to clarify this point.

[Q4]

In the installation notes of the README.md, the first step points to virtualenv, virtualenvwrapper. I would have appreciated if authors could provide in there how I can install virtualenv. It is just need to take a few instructions from <https://virtualenv.pypa.io/en/stable/installation/>.

[A4]

We changed the documentation and added information regarding the installation of a virtual environment.

Thank you again for your review. We hope the changes we applied to our github documentation will clarify MetaGenSense's description and its installation procedure.

Best regards,

Olivia Doppelt-Azeroual

Competing Interests: None

Version 1

Reviewer Report 24 June 2015

<https://doi.org/10.5256/f1000research.6578.r8902>

© 2015 Hiltemann S. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Saskia Hiltemann

Department of Bioinformatics, Erasmus University Medical Center, Rotterdam, The Netherlands

General Comments:

The authors describe their application, MetaGenSense, a web-based application for analysing metagenomic data. It provides a user-friendly interface which combines a LIMS system with a Galaxy backend for computation and workflow management.

The Django framework is very nice, and I think the integration of Galaxy with a LIMS system is very useful and something many readers will be interested in.

However, many aspects of this application are tailored specifically to the authors' local setup. I installed parts of the application, but since no information is provided on how to install the various components (LIMS/Django/KRONA/BioBlend), and since the Galaxy server used in the code was not accessible to me, it was not fully functional, and because documentation was lacking, it was unclear to me how to proceed with the setup.

To make this work more valuable to the readers, the following additions would be helpful:

1. Installation instructions for the code on GitHub. The readme file is empty at the moment.

How to install the various components (LIMS, Django UI, KRONA, BioBlend)? And how to connect the different components together? How to configure the webserver correctly (apache/nginx/other)? Which parts of the code are specific to the authors' local setup and need to be adapted when readers install their own MetaGenSense instance?

2. A description of the Galaxy workflows used by the authors would also be very interesting, which tools are used? are they available from the Galaxy tool shed?
3. Either create a demo server with an example project or add screenshots of the application

to the manuscript. The UI looks quite nice, show it to the readers.

4. The case study section is very technical, and would be enhanced by showing the use-case in terms of a real biological example, add screenshots of a real-world analysis to the various steps in this section.

Minor Edits:

1. Capitalize the word "Galaxy" throughout.
2. In section "Bioinformatics and HTS projects", BiobBlend --> BioBlend

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 06 Aug 2015

Olivia Doppelt-Azeroual, Institut Pasteur, F-75724, Paris, France

Thank you very much for your critical review. We posted the answers for your remarks underneath them in bold.

To make this work more valuable to the readers, the following additions would be helpful:

1. Installation instructions for the code on GitHub. The readme file is empty at the moment.

Yes, the README file is now complete. It contains installation information. Please tell us if you feel that it gathers all useful information to install and configure MetaGenSense.

Along with the version 2 of MetaGenSense publication, we built a release of the software. It is available at the URL (<https://github.com/pgp-pasteur-fr/MetaGenSense/releases/tag/v1.0>)

How to install the various components (LIMS, Django UI, KRONA, BioBlend)? And how to connect the different components together? How to configure the webserver correctly (apache/nginx/other)? Which parts of the code are specific to the authors' local setup and need to be adapted when readers install their own MetaGenSense instance?

Concerning the LIMS, Django, BioBlend, everything is in the application itself so the connection between the components is natively implemented.

For the apache, it is directly linked to Django which is deployed on an apache server. It is very well documented on this url:

<https://docs.djangoproject.com/en/1.8/howto/deployment/>

For KRONA, the javascript which enables the taxonomy distribution exploration is actually generated by a tool, installed in Galaxy. At the Institut Pasteur, it is part of an in-house package which gathers several tools for taxonomy analyses. Those tools are available on GitHub now (it was not a year ago) at the URL: https://github.com/C3BI-pasteur-fr/taxo_pack. To test those tools, we implemented a Virtual Machine image pre-configured to test MetaGenSense directly on a web browser. Instructions are also available in the GitHub README file.

For your last question, please look at the “set the settings” part of the README file.

2. A description of the Galaxy workflows used by the authors would also be very interesting, which tools are used? are they available from the Galaxy tool shed?

The workflow included on the virtual Machine Galaxy instance contains a light version of our metagenomic analysis workflow. A small fastq file is also included to test it.

3. Either create a demo server with an example project or add screenshots of the application to the manuscript. The UI looks quite nice, show it to the readers.

Thank you for that. Yes, another reviewer suggested that we put screenshots of the UI. We added a figure that resumes all the windows and their use for each step. The figure is available at this url: https://github.com/pgp-pasteur-fr/MetaGenSense/blob/master/doc/images/metagensense_complete.jpeg as well as each of the steps (in bigger picture). This figure is now added to the publication.

4. The case study section is very technical, and would be enhanced by showing the use-case in terms of a real biological example, add screenshots of a real-world analysis to the various steps in this section.

Please tell us if the figure resumes well the various steps.

For the minor remarks, they are all included in the second version of the publication. Thank you for your critical review of our article.

Competing Interests: No competing interests were disclosed.

Reviewer Report 22 June 2015

<https://doi.org/10.5256/f1000research.6578.r8217>

© 2015 Corpas M et al. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuel Corpas

Future Business Centre, Cambridge, UK

Benedikt Rauscher

The Genome Analysis Center, Norwich, UK

MetaGenSense is intended to help find pathogen data in metagenomic data created through next generation sequencing. Measured data including the sequencing reads and metadata are fed into a Laboratory Information Management System (LIMS). The application can fetch that information and pipe it into predefined Galaxy workflows, run them and visualise the output via a framework called KRONA.

The introduction to the article is perhaps too long (almost half of the article). There are sections that are not necessarily related to the research presented here, e.g., the paragraph focusing on the assembly problem of next generation sequencing reads. It would be useful, however, that the authors give a more comprehensive introduction into metagenomics as this topic is only covered very briefly at the beginning of the introduction.

The section on the software tool itself is very technical. I have trouble identifying a clear train of thought. Also it could be shorter and more precise. The case study does not really seem to be a case study on how the application can be used to actually find pathogen information in metagenomic data but is more like a step by step protocol on how to use the application. I suggest that this kind of information should be moved to the documentation and that instead a concrete biological example is demonstrated in the article. Moreover, the title says that MetaGenSense can visualise its results. However, this is not shown in the article. Therefore I would advise the authors to consider replacing the current figure with a figure demonstrating the results of a concrete biological use case.

The discussion and conclusion seem to be a summary rather than a discussion.

MetaGenSense seems to lack many of the standard requirements of a quality software product

- We could not find any documentation. The README file on GitHub does not contain any information.
- The last update to the code was months ago, suggesting that the program is not being developed and maintained actively.
- We could not find any tests.
- We could not find any examples, demos or even screenshots of the interface.

Therefore we are not convinced that MetaGenSense adheres to the journal's quality standards. We believe that the article should be revisited and documentation, live examples and tests should be added to the software before the article should be indexed.

Competing Interests: No competing interests were disclosed.

We confirm that we have read this submission and believe that we have an appropriate level of expertise to state that we do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Author Response 28 Aug 2015

Olivia Doppelt-Azeroual, Institut Pasteur, F-75724, Paris, France

We tried to answer each of your comments. A second version of the publication will be available soon.

Thank you for your critical review of our article.

Reviewer Comment:

MetaGenSense is intended to help find pathogen data in metagenomic data created through next generation sequencing. Measured data including the sequencing reads and metadata are fed into a Laboratory Information Management System (LIMS). The application can fetch that information and pipe it into predefined Galaxy workflows, run them and visualise the output via a framework called KRONA.

The introduction to the article is perhaps too long (almost half of the article). There are sections that are not necessarily related to the research presented here, e.g., the paragraph focusing on the assembly problem of next generation sequencing reads. It would be useful, however, that the authors give a more comprehensive introduction into metagenomics as this topic is only covered very briefly at the beginning of the introduction.

Introduction is now shorter in the second version of the publication. It was modified in a way that more emphasis is made on the importance of a true metagenomics component for HTS-type analyses. Information regarding assembly problems are indeed of secondary importance, and they were removed for the new version.

Reviewer Comment:

The section on the software tool itself is very technical. I have trouble identifying a clear train of thought. Also it could be shorter and more precise. The case study does not really seem to be a case study on how the application can be used to actually find pathogen information in metagenomic data but is more like a step by step protocol on how to use the application. I suggest that this kind of information should be moved to the documentation and that instead a concrete biological example is demonstrated in the article. Moreover, the title says that MetaGenSense can visualise its results. However, this is not shown in the article. Therefore I would advise the authors to consider replacing the current figure with a figure demonstrating the results of a concrete biological use case.

Thank you for this interesting remark; the use-case part of the article was modified to be more precise. Moreover, a user can now use MetaGenSense on a concrete biological

dataset. We created a virtual machine image preconfigured to directly use MetaGenSense with a web browser. A fastq file, as well as a light version of our metagenomic Galaxy workflow is included. The user can connect to the framework, launch an analysis, explore the framework, as well as the results. Moreover, we also would like to change the title of the publication for it to be more adapted to our approach; replacing visualisation by the word exploration which is really the goal of an application like MetaGenSense. The title of the second version of the article is:
"MetaGenSense : A web application for analysis and exploration of high throughput sequencing metagenomic data."

Reviewer Comment:

The discussion and conclusion seem to be a summary rather than a discussion.

MetaGenSense seems to lack many of the standard requirements of a quality software product

- We could not find any documentation. The README file on GitHub does not contain any information.

MetaGenSense README file is now complete. We also have written on a userGuide available directly on our Github repository through the web tool readthedocs : <http://metagensense.readthedocs.io>.

Reviewer Comment:

- The last update to the code was months ago, suggesting that the program is not being developed and maintained actively.

The code in the GitHub repository was committed just before the submission of the article. A few debug and add-ons have been implemented since the previous release.

Reviewer Comment:

- We could not find any tests.

As mentioned earlier, we implemented a Virtual Machine Image containing the infrastructure to test our framework. It is pre-configured so that any user can start using MetaGenSense with a web browser. It is available on the Institut Pasteur server as it was too big to be uploaded on GitHub (<http://webext.pasteur.fr/metagensense/metagensense.ova>). Documentation about this image is available on the README file of our GitHub repository. As Metagenomic analyses are time and storage consuming, we made available a very light version of our workflow with a small fastq file. However, it is enough to test the framework and to understand how the database, the Django framework and the related Galaxy instance are working together.

Reviewer Comment:

- We could not find any examples, demos or even screenshots of the interface.

For the screenshots, we added a picture of the framework at each step of its use. It is also available at the url: https://github.com/pgp-pasteur-fr/MetaGenSense/blob/master/doc/images/metagensense_complete.jpeg

Moreover, as the screenshots are small, they are also available on our github repository in the doc directory.

Competing Interests: No competing interests were disclosed.

Reviewer Report 19 June 2015

<https://doi.org/10.5256/f1000research.6578.r8901>

© 2015 Li P. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peter Li

GigaScience, Hong Kong SAR, China

General comments

The authors have developed MetaGenSense, a web-based application for analysing metagenomics data. MetaGenSense also acts as a laboratory information management system, managing the metagenomics data and the results of their analysis by Galaxy workflows.

The work done by the authors sounds promising since I am a fan of Galaxy and of the Django framework. However, I believe the paper can be much improved by providing additional details in a number of places.

For example, it would be interesting to learn more about the two prototyped workflows for analysing metagenomics data which are alluded to in the "Pre-designed Galaxy workflow" section on page 5.

The "Case study - use example" section could also be improved by providing screenshots of the MetaGenSense GUI which are relevant to each or some of the steps. At the moment, I have no idea what the GUI for MetaGenSense looks like since there is also no example instance of MetaGenSense available on the Web which would have been useful for reviewing this paper.

In the final paragraph, it is stated that MetaGenSense can be easily deployed but looking at the source code in <https://github.com/F1000Research/MetaGenSense>, there appears to be a lack of documentation to enable me to do this. For example, which version of Django should I use with

MetaGenSense? How do I get the MetaGenSense source code integrated with the Django framework? The LIMS for MetaGenSense uses PostgreSQL; how do I set up this database and get it linked to MetaGenSense? How do I provide MetaGenSense with access to a Galaxy server? I think the authors need to provide this type of information to help readers install MetaGenSense for the authors' source code to be more useful.

Minor corrections

Page 3:

Paragraph 4

Pacific Bioscience should be Pacific Biosciences.

Paragraph 7

BiobBlend should be BioBlend.

Page 4

workflow_remote should be workflow_remote

like BWA through galaxy - should be, "like BWA through Galaxy..."

Page 5

Case study - use example: Need to be consistent by starting bullet points with capital letters.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 06 Aug 2015

Olivia Doppelt-Azeroual, Institut Pasteur, F-75724, Paris, France

Thank you for your review. For each of your remarks, we wrote responses in bold:

For example, it would be interesting to learn more about the two prototyped workflows for analysing metagenomics data which are alluded to in the "Pre-designed Galaxy workflow" section on page 5.

This publication aims to present an application combining as you wrote above a lims, a direct link to any Galaxy and a way to sort and manage Galaxy results. The workflow you choose to use is totally arbitrary. In essence, MGS was designed in a way that any

Galaxy workflow can be plugged-in.

To facilitate testing, we implemented a Virtual Machine image pre-configured to test MetaGenSense directly on a web browser. Instructions for download and use are available in the GitHub README file. However, as instructions for installation of the framework are now available, any developer can download and link MetaGenSense to his Galaxy.

The "Case study - use example" section could also be improved by providing screenshots of the MetaGenSense GUI which are relevant to each or some of the steps. At the moment, I have no idea what the GUI for MetaGenSense looks like since there is also no example instance of MetaGenSense available on the Web which would have been useful for reviewing this paper.

Yes we agree with you, we omitted to add screenshots in the first version of the article. We added a new figure that resumes all MetaGenSense steps and functionalities. It is available at: https://github.com/pgp-pasteur-fr/MetaGenSense/blob/master/doc/images/metagensense_complete.jpeg. It is now the 2nd figure of the article. Moreover all the small figures (steps) are also available in the doc directory of our GitHub repository.

In the final paragraph, it is stated that MetaGenSense can be easily deployed but looking at the source code in <https://github.com/F1000Research/MetaGenSense>, there appears to be a lack of documentation to enable me to do this. For example, which version of Django should I use with MetaGenSense? How do I get the MetaGenSense source code integrated with the Django framework? The LIMS for MetaGenSense uses PostgreSQL; how do I set up this database and get it linked to MetaGenSense? How do I provide MetaGenSense with access to a Galaxy server? I think the authors need to provide this type of information to help readers install MetaGenSense for the authors' source code to be more useful.

We improved the documentation on our github repository (<https://github.com/pgp-pasteur-fr/MetaGenSense>) with elements facilitating deployment. We completed a README file containing well-informed installation and configuration instructions. We also wrote a UserGuide working with readTheDocs documentation tool (<http://metagensense.readthedocs.io>). Don't hesitate to tell us if parts of the documentation are still a bit blurry.

We will adapt the manuscript with all your minor corrections and suggestions.

Thank you very much for your review.

Competing Interests: No competing interests were disclosed.

Reviewer Response 27 Sep 2016

Peter Li, GigaScience, Hong Kong SAR, China

To test the authors' software, it will take over 6 hours for me to download the 2.7 GB machine image provided by the authors. I think this is due to my office internet connection being very slow and because of the fact that I am based in the Far East. Unfortunately, waiting 6 hours plus for the machine image to download is not really practical for me but this is not a fault of the authors.

I had a look at the instructions to install MetaGenSense which are available from the README.md file in their GitHub repository in an attempt to manually install the software. I feel that the instructions are minimal, for example, the installation requires a requirements.txt file but I could not find one in the MetaGenSense repo. You would need to know that the contents of requirements.txt are listed at the start of the manual installation instructions. The installation instructions also rely on the user doing background reading to find out how to install and use MetaGenSense's dependencies, for example, virtualenv and a database server.

Competing Interests: No competing interests were disclosed.

Author Response 21 Nov 2016

Olivia Doppelt-Azeroual, Institut Pasteur, F-75724, Paris, France

Dear Peter Li,

Thank you for your review and remarks.

Since last August, we asked a non familiar with MetaGenSense colleague to read and install locally the application. This exercise has enabled us to clarify some points. We simplified the README file (<https://github.com/pgp-pasteur-fr/MetaGenSense>) and added significant details to the installation procedure part in the readthedocs documentation:

- As you mentioned, the virtualenv installation procedure has been added.
- We also clarified the part concerning the requirements, i.e. the developer needs to copy the three requirements in a requirements.txt file and use the "pip" command to install them.
<http://metagenense.readthedocs.io/en/latest/installation.html#requirements>
- The part concerning the database configuration has also been a bit modified in order to be clearer.

We hope the changes applied recently enable you to successfully install MetaGenSense.
Best regards,

Olivia Doppelt-Azeroual

Competing Interests: None

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research