VESPUCCI

Release 1

Marco Moretto

Sep 27, 2022

TABLE OF CONTENTS

VESPUCCI is a COMMAND>_ technology-based database for exploring and analyzing a comprehensive *Vitis vinifera* specific cross-platform expression compendium. This compendium was carefully constructed by collecting, homogenizing and formally annotating publicly available microarray and RNA-seq experiments from Gene Expression Omnibus (GEO), Sequence Reads Archive (SRA) and ArrayExpress.

VESPUCCI's major features include:

1. Access to gene expression compendium that combine information from different platforms and experiments.

2. Integration of information from main curated databases allowing the user to interactively browse and query the compendia for specific genes, pathways, protein domains, and more.

3. An extensive formal sample annotation, allowing the user to interactively browse and query the compendia not only for specific arrays, samples, or experiments, but also for specific experimental conditions and biological processes.

4. A suite of flexible yet intuitive tools to explore, analyze and visualize the expression data in the compendium.

5. Powerful programmatic access to the database via a GraphQL web service. We also provide an API to connect to and query the VESPUCCI database directly from within Python and R.

HOW TO CITE

If you use VESPUCCI and find it useful please cite:

Moretto, M. et al. (2022). *A COMPASS for VESPUCCI: A FAIR Way to Explore the Grapevine Transcriptomic Landscape.* Frontiers in plant science, 13:815443. 10.3389/fpls.2022.815443

OR

Moretto, M. et al. (2016). *VESPUCCI: exploring patterns of gene expression in grapevine.* Frontiers in plant science, 7, 633. doi 10.3389/fpls.2016.00633

1.1 Query VESPUCCI

VESPUCCI is accessible via the COMPASS GraphQL interface that has been specifically developed to facilitate data retrivial and analysis of gene expression compendia built with the COMMAND>_ technology. You have four options to query VESPUCCI:

1. use the GraphQL endpoint directly, COMPASS

2. use the Python package pyCOMPASS

3. use the R package rCOMPASS

We recomend to use either option 2 or 3 unless you need to use the GraphQL endpoint specifically. Both the Python and R packages provide the same options as the GraphQL endpoint but with all the expressiveness of a higher order programming language. Option 4 is not meant for data analysis but to showcase how to create tools on top of pyCOMPASS, rCOMPASS or COMPASS directly.

1.1.1 Use the GraphQL endpoint

GraphQL is an open-source data query and manipulation language for APIs, and a runtime for fulfilling queries with existing data. The GraphQL interface to access VESPUCCI is COMPASS. The GraphQL endpoint is available at http://vespucci.fmach.it/compass.

1.1.2 Use Python package pyCOMPASS

pyCOMPASS is the Python interface for the COMPASS GraphQL interface. It is the preferred way to access VESPUCCI for data analysis purpose since it provides with higher level functionalities. You can install pyCOMPASS via PIP:

```
pip install pycompass
```

1.1.3 Use the R package rCOMPASS

rCOMPASS is the R interface for the COMPASS GraphQL interface. It is the preferred way to access VESPUCCI for data analysis purpose since it provides with higher level functionalities. You can install rCOMPASS from github:

```
devtools::install_github("onertipaday/rcompass")
```

1.2 Use Cases

Note: The following Colab notebooks are public and editing is allowed to specific users only thus in order to use them, logout from your Google account first or use the incognito mode!

1. In this first excercise we will perform few basic operations with VESPUCCI. We will create a Module starting from few genes and then automatically extend it by adding more genes. We will also have a look at gene and samples annotations. The use case Notebook is hosted in Google Colab

2. The second exercise tries to show a more general data analysis pipeline that relies on VESPUCCI for the data but integrates other tools and an packages, such as Scikit-learn, used to fit a Gaussian Mixture Model and GOAtools used for the Gene Ontology Enrichment. Everything is seamlessly integrated in one analysis workflow that ca be explored in this Notebook hosted in Google Colab.

3. In this use case, we have queried the database with a short list of genes putatively involved in pollen development with the final goal of retrieving additional genes with the same function. This Notebook hosted in Google Colab.

4. In this use case the main aim is the characterization of a grapevine gene family. Specific objetives are to verify if the different gene members have a tissue-specific expression and to identify the members those expression is significantly modulated upon biotic stresses. This Notebook hosted in Google Colab.

5. In this final exercise we start from few thousand of genes related to a Myb transcription factor and try to determine which are tissue-specific. After the initial filtering we attached the result with a GO enrichment analysis using GOAtools. This Notebook hosted in Google Colab.

TWO

AUTHOR

To send me an e-mail about anything else related to VESPUCCI write to

_static/email.png

LICENSE

The project is licensed under the GPLv3 license.