Home

Edit New page

Jump to bottom

Mattis Kaumann edited this page now · 9 revisions

uGene

Working title: Improve phylogenetic profiling with uMap data science

Project Group

- Vinh Tran
- Ingo Ebersberger
- Mattis Kaumann

Project motivation

Genetic profiling plays a significant role in our research group, enabling us to gain insights into gene behavior and evolutionary relationships. Phylogenetic profiles have proven to be highly useful in various applications, and PhyloProfile is our primary tool for this purpose. However, interpreting the information within these profiles can be challenging, mainly due to the large volume of data and the order in which genes are presented. To address these challenges, we propose a novel approach that integrates the uMap algorithm, known for its speed and accuracy in single-cell clustering, to improve the interpretability of phylogenetic profiles. Two primary downsides of standard phylogenetic profiling are present. Firstly, the size of the data poses difficulties, as each gene is associated with a comprehensive column of information representing its orthologs in various taxa. This can be overwhelming, particularly with large datasets. The second challenge lies in the order of genes within the profiles. PhyloProfile selects one order from many possible arrangements of genes and this choice directly impacts the outcomes of the possible investigation. Although PhyloProfile has a couple of cluster algorithms to determine this order. This reduction to a one-dimensional representation is just one of many possible, which can limit the effectiveness of the analysis. To overcome these challenges, we propose the usage of the uMap algorithm. Thereby we end up in clusters, where every gene is represented by one dot and because of the uMap architecture, there will not be guided by a one-dimensional structure.

Project Implementation

To begin, we have chosen to utilize the uMap clustering algorithm provided by the sklearn Python library. We create a data pipeline to cluster any subset of data with manual settings downloaded from PhyloProfile software. This includes .out and .phyloprofile files. As well we provide an express version with a graphic user interface, to manage the input conversation, the cluster job, and the result presentation. For those use cases, with large datasets or polluted datasets, where an all-in-one solution is not useful we as well introduce a command line interface. The whole pipeline is organized in three steps. First data collection and translation, then in the second step perform the uMap clustering and at least the result presentations. The first step is performed by the outToCSV.py converter script. The second and main analytic process is implemented into the main.py file. The display part is managed from the uGen.py witch one includes a user interface to perform the most common use case of the upstream scrips. To keep it simple as possible, all handover data are stored in one molten data frame as Pandas readable .csv-file.

How to install

Download form uGene Download all files. Extract these file into the installation location.

Installation with Pip

To install uGene, first, open the 'packages.txt' file and install all the necessary Python libraries or use the the following command to install pip requirements file:

```
pip install -r requirements.txt
```

Once all the packages are successfully installed, run the 'uGeneGUI.py' file using Python:

```
python3 uGeneGUI.py
```

The interpreter will provide you with an IP address and a port number, which you should open in your preferred browser.

Installation with Conda/Pip

If you have Conda, you can create a dedicated environment for uGene and install all the required dependencies. If Conda is not already installed, please install Anaconda on your system and add Conda to your system's PATH.

Follow these steps: Create and activate a Conda environment named 'uGene':

```
conda create -n uGene
```

```
conda activate uGene
```

Install and use 'pip' in the Conda environment: conda install pip pip install -r requirements.txt

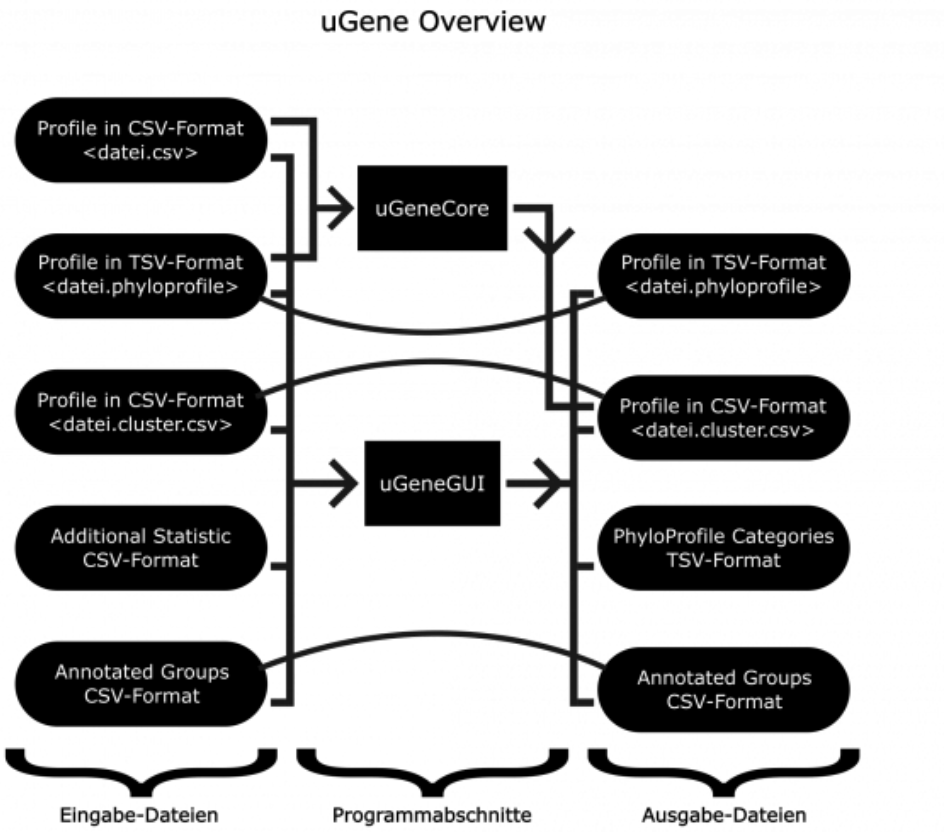
Enable PhyloProfile support

To enable the PhyloProfile support, make sure you be able to open your profiles with PhyloProfile. Follow the installation instruction and FAQ thread on the PhyloProfile wiki. PhyloProfile I assume that R is installed and usable on your machine. Make sure R is added to the path and callable by the command line. Windows users be able to use the following command:

```
setx /M path "%path%;YourRInstallationPath\bin\x64
```

Pipeline Usage

Keep in mind, this project is based on just a few scripts, you be able to start them with command line inputs as well by using scripts like .bat or .sh files. There are two major ways to use uGene. You be able to use it fully as a command line app or with our dashboard user interface.



The Chart shows both program sections of the uGene software package, along with the associated input and output files. The programs are represented as rectangular blocks. Arrows symbolize the data and information flow in the network. Blocks with rounded edges represent files, where their content, format, and optionally a file extension can be specified. If a file extension is provided, it is mandatory. Files with the same format and content are connected by a curved line. According to the specifications that are required, it is advisable to use either uGeneCore.py or uGeneGUI.py. If you are using the program for the first time, it is recommended to proceed with uGeneCore.py. If you are unable to create clusters with uMap due to hardware resources, you can use uGeneCore.py. Both options offer adjustments for the uMap parameters, however, the customization options with uGene are even greater.

1. uGeneGUI.py

To use the uGene dashboard start the uGeneGUI.py with the python3 interpreter and navigate into your browser to the app location. The app ip will be provided after the app is started. Eventually, your interpreter supports ip links, thereby click on the ip to open the browser. The dashboard version enables you to use the uGene pipeline on the most common use. Just load the .phyloprofile file and press "Run uGene". After the cluster job is done, you be able to investigate the results. Before using the investigate PhyloProfile tab, make sure the loaded profile can display by PhyloProfile and r is callable by the command line. Additionally, you be able to add a statistic file. Therefore use the sidebar. For more details read the how to add additional statistic tab. All in all, you be able to define groups and select genes to put them into these groups based on the clustering result. Use afterward third-party tools to investigate your genes of interest.

Load Profiles

There are two options available. Either you choose to load a .phyloprofile file, or you want to upload an already clustered file. A file clustered with uGene has the file extension .cluster.csv. It is important that uGene recognizes file types based on their file extension, so it is crucial for a .phyloprofile file to have the correct format. The key features of a .phyloprofile file are that it is a DataFrame with "\t" as the delimiter. The important column names are geneID, ncbiID, orthoID, FAS_F, and FAS_B, which must be included in the header. In addition to these column names, a .cluster.csv file must also contain the columns gene1d_x, gene2d_x, gene2d_y, gene3d_x, gene3d_y, and gene3d_z. Furthermore, the delimiter in a .cluster.csv file is "," since it is in CSV format.

Cluster Options

In the graphical interface, the most important settings for uMap are provided for general use. For example, if you change a value under min_dist, it will be taken into account in the subsequent cluster job. After adjusting the options, press "Run uGene" to update the clusters. In addition to these settings, there is also the possibility to set advanced options. Check the box for Advanced Options and enter your parameters in the text field below. For example, the parameter min_dist can be overridden as follows: "min_dist=0.5". All arguments of the function umap.UMAP() from sklearn can be set. See uMap documentation

Working with Groups

Once a cluster is successfully computed or loaded, the view automatically switches to the data viewer. Here, groups can be created with individual names, and data points can be added to or removed from these groups. Select the tool for adding and use the Select tool in the plot to mark several genes. They will be indicated as members of your group by an outline, and the IDs of all members will be listed in the table below. It is also possible to expand a group based on its known IDs. Open the designated dialog and enter the names of the IDs separated by commas in the text field.

Additional Statistics

It is possible to display information about genes in the sidebar. For this purpose, data can be sent for up to 10 bar plots. The data must be combined in a database, where the first column contains the ID of the data point, the second column contains the ID of the 1-10 plots, and the third column contains the information or property associated with the data point. The DataFrame must have a header and be in CSV format. Any number of pieces of information can be passed to a data point. By selecting some data points in the cluster, statistics of the most represented properties will be displayed. The number of properties to be shown per plot can be adjusted below all the plots.

Download

In the last tab, the results can be downloaded. You can either download the groups created based on the clusters or the entire profile. For both options, there is a general download in CSV format or specifically formatted files for PhyloProfile. These files can be used to achieve the same profile representation as in the Integrated PhyloProfile tab. Furthermore, there are additional settings for downloading the profile. These settings are identical to the additional display options in the PhyloProfile tab.

Download Option	Description
origin	The profile is downloaded in the original sorting order of the genes. If a .cluster.csv file has been loaded that no longer reflects the original order of your profile, this option will not be correct.

Download Option	Description
1d_order	The genes in the profile are arranged based on the one-dimensional sorting of uMap. This has the advantage that marked groups are displayed as separate blocks.
group_based	A profile with the original sorting order of the genes is returned, but the genes from the groups are appended to the profile at the bottom to view the entire group as a whole.
1d_group_based	The genes in the profile are sorted based on the one-dimensional uMap sorting, and genes from the groups are appended to the profile at the bottom.

2. uGeneCore.py

The uGeneCore.py is the heart of the tool and is also fully controlled by command line inputs. It gets a CSV-File and produces a cluster.csv file, which contains additional columns with cluster results. Cluster result columns are named by their job name. The following arguments are usable with main.py.

Argument	Optional	Format	Value example	Explanation
--help, -h	no	void		Show a small help.
--file, -f	no	String	test.csv	Filename with path to a .csv file of data
--tasks, -t	no	List	[{'x_axis':'geneID', 'y_axis':'ncbiID', 'values':'FAS_F','jobs':'tax'}]	Main task is a list with containing tasks. One task has to be a dictionary within specific parameters. Read below how to write one task.

How to write a task. Remember, one task has to be a dictionary and can hold serval cluster jobs. To cluster the molten dataset, it needs to get pivoted for technical reasons. Therefore an x_axis and a y_axis must be chosen, as well as the values to calculate a data matrix. All containing cluster jobs get calculated based on this matrix. One task has to hold at least one job. All these parameters x_axis, y_axis, values and jobs have to be defined. It is allowed to put just a string or a list of strings in each of these parameters. How to write a Job One Job is a dictionary with serval options on how the uMap clustering has been calculated. A job produces depending on the number of n_components new columns in the resulting data frame. Not all parameters have to be defined, but keep in mind that an undefined parameter always gets replaced by standard parameters. A list with the most common parameter follows, but there are still a lot more on umap-wiki.

Parameter	Standard value	Explanations
job_name	unnamedJob	Names the column in the resulting data frame. Names will contain a tail like 2D_x related to the dimension kind of information
n_components	1	Set how many dimensions the resulting cluster should have.
n_neighbors	15	Size of training sets, effects local vs global structure research.

Parameter	Standard value	Explanations
min_dist	0.1	Means the minimal distance to points possibly can be close to together.
metric	Euclidean	Describes the method of how the distance of two points will be calculated.

Data mining with the uGene pipeline

We have adopted a completely new approach by utilizing the uMap dimension reduction technique for genetic profiles, and it is crucial to demonstrate its effectiveness. In addition to the development of uGene, we have undertaken a data mining project focused on identifying certain genes that are lost in rodents (LROD). Information regarding the genes of interest can be found here. You can find details about the investigations and outcomes of using uGene with this dataset here. In this project, we successfully clustered a dataset consisting of 17056 genes and 169 taxa efficiently. Furthermore, we were able to confirm a predefined group with a loss in rodents. Additionally, this group was expanded by 172 genes. The Lost in Rodents project (LROD-Project) demonstrated the versatile application of the uGene software in the context of phylogenetic profiles.

To underline the evidence of your results, we decide to establish a benchmark based on known GO-Terms. Then we decided to use GO annotations from UniProt and the tool GOGO developed by Chenguang Zhao. The core results of this benchmark underline that the result of uGene not being ended up in random clusters with no significant information. Click here to read more about the GOGO-benchmark project: GOGO-Benchmark

Known issues

Issues with the all_human_across_mammalia.phyloprofile dataset

This error appears due to the newest pynndescent version 0.5.10 dependency. If you do not receive the message: 'assign slice from input of different size,' then you don't need to worry about this error. It's only the configuration and specific values within these particular datasets that lead to this error.

Full uGeneCore.py error output:

```
ERROR:root:cannot assign slice from input of different size
ERROR:root:cannot assign slice from input of different size
ERROR:root:cannot assign slice from input of different size
ERROR:root:No cluster data is produced in total.
```

Current working solution:

```
pip uninstall pynndescent
pip install pynndescent==0.5.8
```

To encounter and investigate this error, we are compiling statistics about the cases in which it occurs. If you have experienced this error with your database, please contribute by adding new datasets to our investigation. Leave the name of the dataset as well as the name of the author so that we can analyze the issue comprehensively.

List of known issues with datasets:

```
all_human_across_mammalia.phyloprofile (FelixLangschied)
```

Issues opening and saving files

The uGeneGUI application has an issue with opening and saving files. When a click event on a button to save or open files is activated, nothing happens other than the pending symbol being displayed next to it.

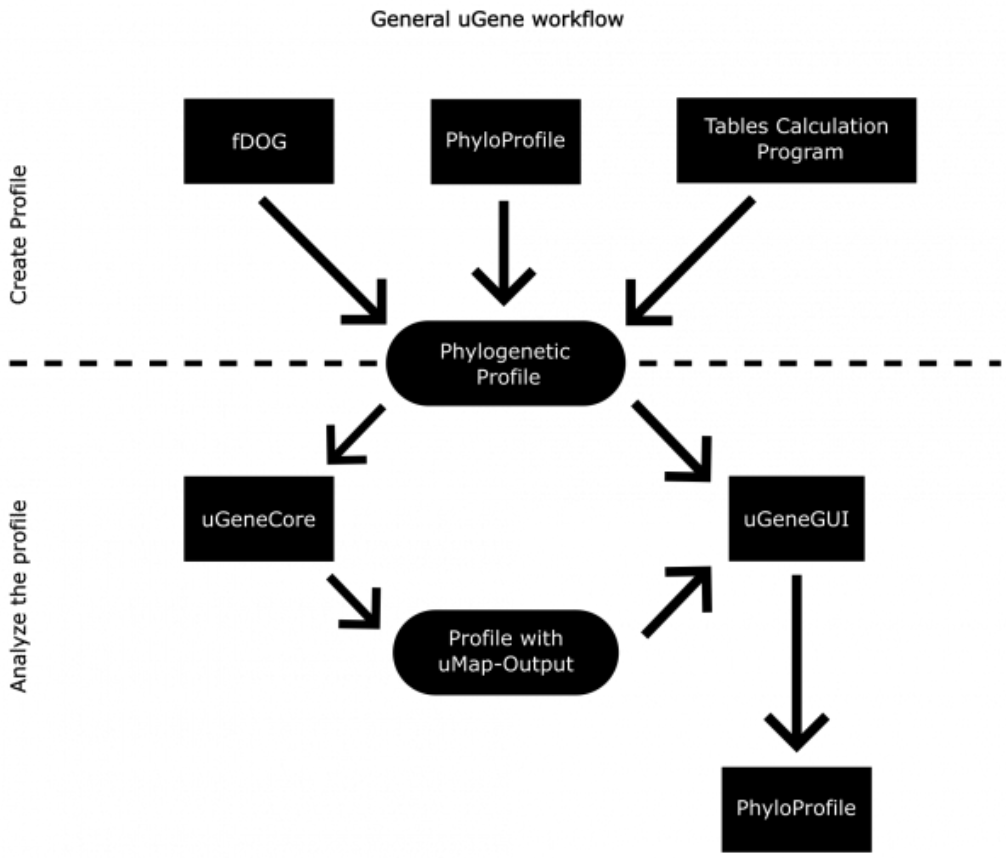
The root of the problem is that the dialog box erroneously opened in the background. One must switch to the background opened window with the file dialog using Alt + Tab.

The Python package tkinter is used for the file dialog, which behaves differently depending on the operating system. The behavior on different operating systems should be compiled in this list. If your operating system is not listed or the reaction differs from the property listed below, please add your experience to improve the software.

Operating System	Error Occurs	Optional Description
Windows 10 Home x64	Yes	
Debian GNU/Linux 11	No	

Project Milestones

With the start of the project, uGeneCore.py was created, which allows phylogenetic profiles to be clustered using the uMap-algorithm. Efforts were then made to directly demonstrate that the uMap results have clusters which are particularly significant for protein-protein-interaction analysis. To improve user experience, the dashboard was further developed. To investigate the displayed clusters, a statistical evaluation, a hypergeometric test, as well as the PhyloProfile representation of the clusters were made possible. The current pipeline can extract sub-profiles from a phylogenetic profile using uMap clustering. Here is an overview of the



current workflow.

Outcome and Summary

Finally, it can be summarized that uGene is a tool that is supportive in the analysis of phylogenetics. It provides the possibility to display phylogenetic relationships of genes in clusters. We were able to show evidence that suggests the displayed information has biological relevance and is not due to random arrangements. However, it should be noted that a susceptibility to artifacts in the data cannot be ruled out, which is why strong signals in the clusters need to be scrutinized. Extra caution is required when there is a particularly ambiguous presence/absence pattern of a gene and its orthologs. uGene cannot replace a thorough analysis of the genes but is very useful for quickly understanding the available data and identifying genes with interesting phylogenetic properties.

Example Datasets

Within the "example" directory in the file location, you'll find two sets of data samples. The first set showcases simulated data, while the second features genuine data. Open the .phyloprofile or the respective .cluster.csv files. Use the graphical interface of uGenGUI to do so.

- A .phyloprofile file that contains only the phylogenetic profile and no cluster results.
- A .cluster.csv file that contains already processed uMap results.
- An additional statistics file with the .csv extension, tailored for specific use cases, counting various properties.

As mentioned before, the first example is pretty small and based on a random distribution. Start with this for your first attempts. The expectation for a random distribution, in 3-dimensional space, is one cluster that resembles a full harmonic ball. Find this example here.

The second example is based on real data. From this data, you should expect real clusters. These clusters should be more than one and not perfectly round. Find this example here.

Useful Links

- uGene
- PhyloProfile external GitHub resource
- PhyloProfile intern wiki page
- uMap learn wiki
- Example Data mining Project with uGene