

taXaminer

Whole genome shotgun data are a great resource for reconstructing the genome of any target species. At the same time, there is a rich source of contaminations, i.e. reads from taxa other than the one you had in mind. Possible contaminations are

- taxa living in close association with your target species. Most prominent examples are bacteria of the gut or skin microbiome, or of symbiotic partners.
- reads representing the genome of the person who handled the data, i.e. human contamination
- contaminated reagents used for extracting or sequencing the DNA
- contamination in the sequencer

There are two main levels to detect such contaminations, either on the **level of the genome assembly**, e.g. with the help of BlobTools, or on the **gene set level**. We will focus on the latter, since it allows to investigate the nature of the contaminations.

We will use the software **taXaminer** (Fig. 1) to characterize the gene set of *C. hominis*. Next to performing the taxonomic assignment using **Diamond searches against the NCBI nrProt database**, taXaminer determines **values for a number of other gene features**, such as read coverage, standard deviation of read coverage from contig mean, gene length, position of the gene (terminal in contig or not), etc. **taXaminer runs then a PCA** on these feature vectors and returns, next to other information, a **3D plot of the taxonomically labeled PCA** in html format. To make full use of the taXaminer output, we have developed the tX-dashboard that you can install locally on your computer.

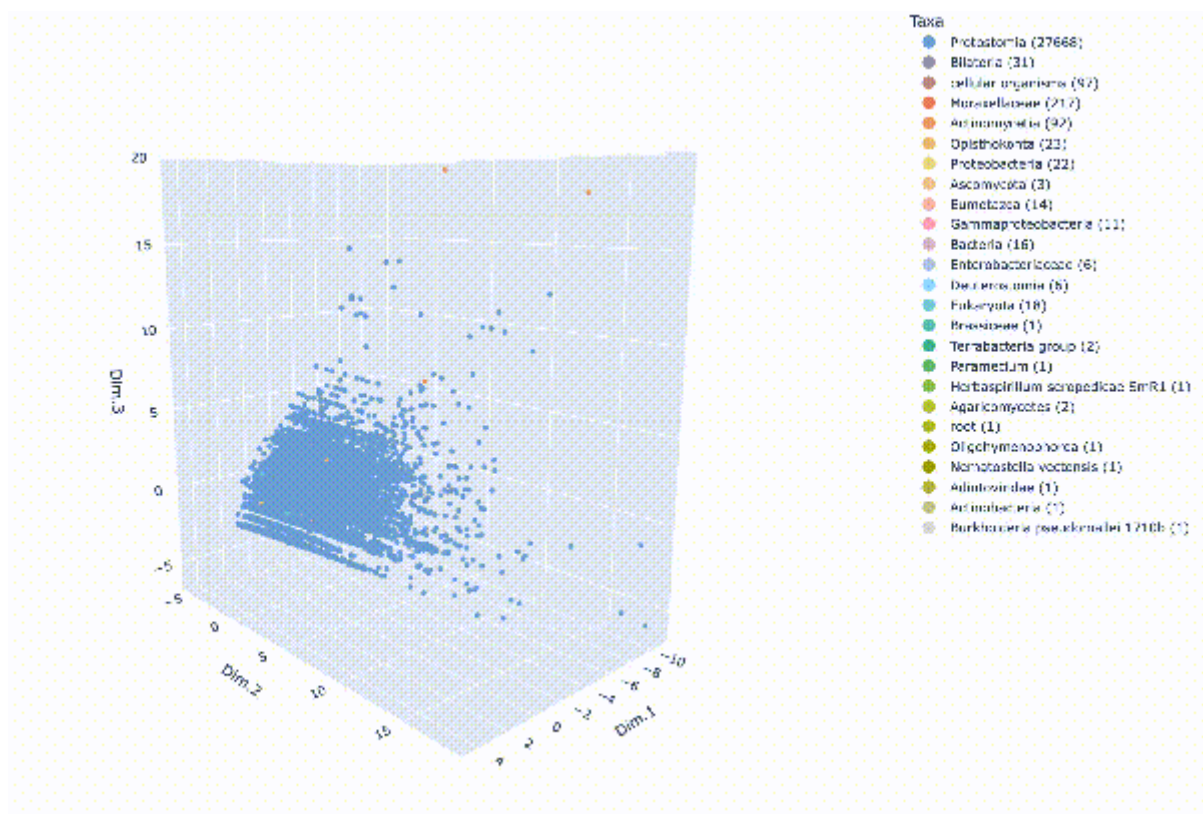


Figure 1: Result of a taxaminer analysis on the gene set of a nematode. Each dot in the PCA represent


a gene that is annotated in the nematode genome. Each gene is annotated with a number of features such as 'numbers of genes on contig', 'position on contig', 'GC content' and the like and a PCA was performed to project the multi-dimensional vectors into 3D space. The dot color represents the taxonomic assignment. If you click on the image, the GIF will be animated

taXaminer analysis

What you need

1. The genome sequence in fasta format. We will be using
/home/ubuntu/Share/Assemblies/crypto_BCM2021_v2.fasta
2. the annotation file in gff3. We will be using

```
/home/ubuntu/Share/Analysis/taxaminer/results/metaeuk/Crypto_Metaeuk.sorted.gff3
```

3. **optionally:** the protein fasta file.  taXaminer will extract the protein sequences from the gff file if not provided.
4. **optionally:** read mapping information: One BAM file per library /home/ubuntu/fritz/sv-detection/for_ingo/illumina_pairs.mapped.sort.bam
5. **optionally** a local installation of the [taxaminer-dashboard](#).

What you get

1. a taxonomic assignment for each gene based on a modified version of the DIAMOND Last Common Ancestor algorithm¹⁾
2. a file with feature vectors for each gene in CSV
3. a html-file with the PCA as a 3D plotly plot
4. a file with the proteins encoded by the annotated genes
5. a text file with the diamond hits

Running taXaminer

1. Check for the presence of taXaminer on your system. To do so:
 1. activate the conda environment:

```
conda activate /home/ubuntu/miniconda3/envs/taxaminer
```

2. issue the following command to test if you can run taxaminer:

```
taxaminer.run -h
```

1. if it installed, proceed with the next steps
2. **if it is not:**

- install taXaminer from the [GitHub page](#) according to the guidelines.
- download the NCBI nonredundant protein database from NCBI: `wget ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/nr.gz`

2. **Optionally** Install the [taXaminer-dashboard](#) on your local computer²⁾
3. create a working directory for the analysis:

```
mkdir -p $HOME/Analysis/taxaminer
```

4. change into the working directory:

```
cd $HOME/Analysis/taxaminer
```

5. copy or soft-link the following files into the working directory
 - The genome sequence in fasta file
 - The genome annotation in gff3 format
 - any read mapping information in BAM format
6. we will be using the unref50 database for the Diamond search.
\$HOME/Share/DBs/uniref50/db.dmnd
7. edit the config-script according to your needs

```
## Input and output options ##
# this section is the minimum information that is required and must be
# stated
fasta_path: "AddYourInputDataHere" # path to assembly FASTA
gff_path: "AddYourInputDataHere" # path to GFF
output_path: "AddYourInputDataHere" # directory to save results to
taxon_id: "AddYourInputDataHere" # NCBI Taxon ID of query species.
# Cryptosporidium parvum has the taxon id 5807
database_path: "$HOME/Share/DBs/uniref50/db.dmnd"

#####
##### from here onwards, the info is optional #####
## Coverage options ##
# state one of the following files to include coverage information
bam_path_1: "" # path to BAM; file; omit to use default location in
# output directory
bam_path_2: ""
# to add further coverage sets duplicate the parameter you need and
# increase the number in the suffix

## Taxonomic assignment options ##
taxon_exclude: "TRUE" # exclude query taxon from taxonomic assignment
# [TRUE/FALSE]
assignment_mode: "exhaustive" # mode for taxonomic assignment
# [exhaustive/quick]; see Documentation for details

## PCA options ##
# gene descriptors to be used in the PCA; see Documentation for details
# on options
input_variables:
"\"c_name,c_num_of_genes,c_len,c_genelenm,c_genelensd,g_len,g_lendev_c,g_
abspos,g_terminal,c_cov,c_covsd,g_cov,g_covsd,g_covdev_c,c_pearson_r,g_
pearson_r_o,g_pearson_r_c\""

## Plot output options ##
```

```
update_plots: "FALSE" # only update the plots (use if you changed
settings below) [TRUE/FALSE]
num_groups_plot: "25" # number of taxa to display in plot (taxa are
automatically merged at higher ranks) [X/all]
merging_labels: [] # influence the merging of taxa; see Documentation
for details on options
```

8. run taXaminer by issuing the following command³⁾

```
taxaminer.run config.yml
```



Make sure that you are either in the directory where the config.yml is located, or provide the path.

Once, the taXaminer run has completed, you can [download the information](#) to your local computer. Then you can either open the 3D_plot.html directly in a web browser, or you use the [taXaminer-dashboard](#) to first import the output folder and then load the data.



1. Are the results in line with your expectations



2. Do you find anything suspicious?

- each dot in the PCA represents one protein coding gene as it was annotated by MetaEuk2 in the *Cryptosporidium parvum* genome assembly CryPa_BCM2021a.fasta
- the position of the dot is determined by a multidimensional feature vector capturing for each gene information such as position in the assembly, word frequency, GC content, #of genes on contig, etc.
- the color code informs about the taxonomic assignment of the gene
- the plot is interactive such that detailed information can be retrieved and downloaded for each gene

[Check out the taXaminer-dashboard:](#)

example input

- [Physalia main page](#)
- [Physalia gene set characterization page](#)
- [Proceed with BUSCO](#)

¹⁾

We modified it a bit because we have a strong prior from which organism the gene should come from

²⁾

This is not necessary, but it makes more fun to look at the data via the dashboard



3)

make sure that the correct conda environment is active

From:

<https://applbio.biologie.uni-frankfurt.de/teaching/wiki/> - **Teaching**

Permanent link:

<https://applbio.biologie.uni-frankfurt.de/teaching/wiki/doku.php?id=general:bioseqanalysis:genesetanalysis:taxaminer>

Last update: **2025/04/08 17:11**

