

SUPPLEMENTARY INFORMATION

Supplementary Note 1: Dependencies

DiNiro is an online tool and the source code and the files required to install it are provided in GitLab Repository of the tool developer. But, all the files there, are old and haven't been maintained like the live version of DiNiro.

The requirements.txt files have old packages versions and some dependencies that doesn't work properly. That's why, some of dependencies and packages needed to be updated to be able to build the docker file.

Solution:

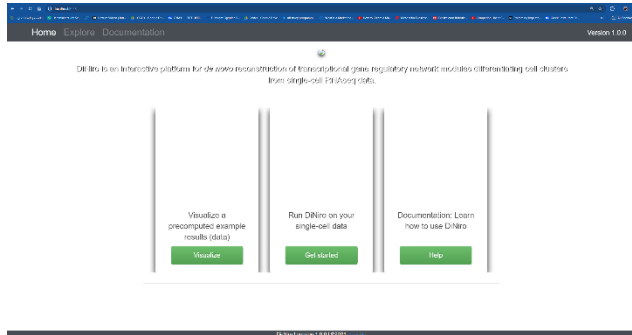
Change the following packages and dependencies versions in requirements.txt file before build up the docker file.

List of changed upgraded packages:

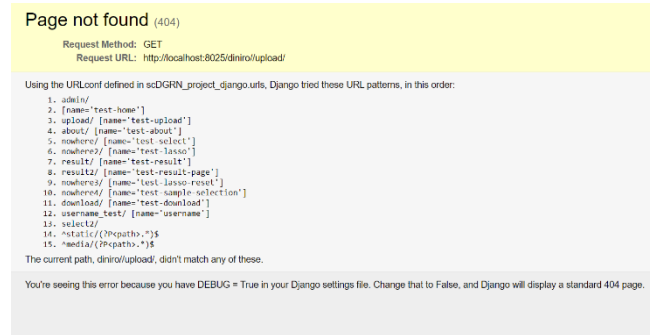
```
gevent==20.9.0 → gevent==24.2.1
greenlet==0.4.17 → greenlet==2.0.0
bioservices==1.7.9 → bioservices==1.11.2
easydev==0.10.1 → easydev==0.10.1
joblib==0.15.1 → joblib==1.2.0
numba==0.51.2 → numba==0.59.1
numexpr==2.7.1 → numexpr==2.10.0
numpy → numpy==1.23.4 #downgraded to deal with deprecation problem
packaging==20.4 → packaging==21.3
patsy==0.5.1 → patsy==0.5.4
scikit-learn==0.23.1 → scikit-learn==1.4.2
statsmodels==0.11.1 → statsmodels==0.14.1
```

Supplementary Note 2: Accessing the local version

In the process of DiNiro locally setup, accessing the tool through the local host <http://0.0.0.0:8025/> is not possible. The tool is unable to access the frontend and backend files and any button returns 404 Error Page as in the screenshots.



<http://localhost:8025>



<http://localhost:8025/diniro/upload/>

- Frontend files were stored in: `/test_interface/static/`
- Backend files were stored in: `/test_interface/tasks/`
- Both of them were already in the right location but the tool wasn't able to access them.

Solution:

rtags.py: is a file responsible for redirecting the buttons in the page and the CSS files to the correct location but it needed to be edited to be executed right.

The Original File:

```
from django.urls import reverse
from django import template

register = template.Library()

@register.simple_tag(name='rurl')
def rurl(url, *args, **kwargs):
    return "/diniro/" + reverse(url)

@register.simple_tag(name='rstatic')
def rstatic(file, *args, **kwargs):
    return "/diniro/static/" + file
```

The Edited File:

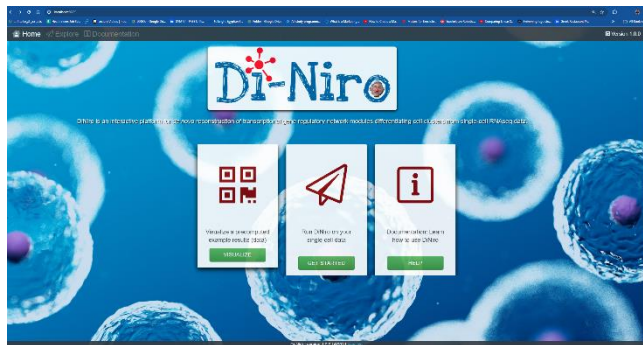
```
from django.urls import reverse
from django import template

register = template.Library()

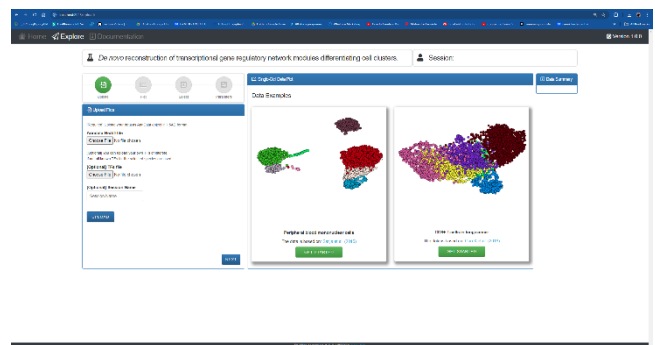
@register.simple_tag(name='rurl')
def rurl(url, *args, **kwargs):
    return reverse(url) # removed /diniro/

@register.simple_tag(name='rstatic')
def rstatic(file, *args, **kwargs):
    return "/static/" + file # removed /diniro/
```

Screenshots of the tool local web page after solving the problem:



<http://localhost:8025>



<http://localhost:8025/diniro//upload/>

Supplementary Note 3: Input

DiNiro input is an Anndata file with the H5AD file format so if you want to upload your dataset you have to transfer it first from CSV or TSV file formats to H5AD file format.

Solution:

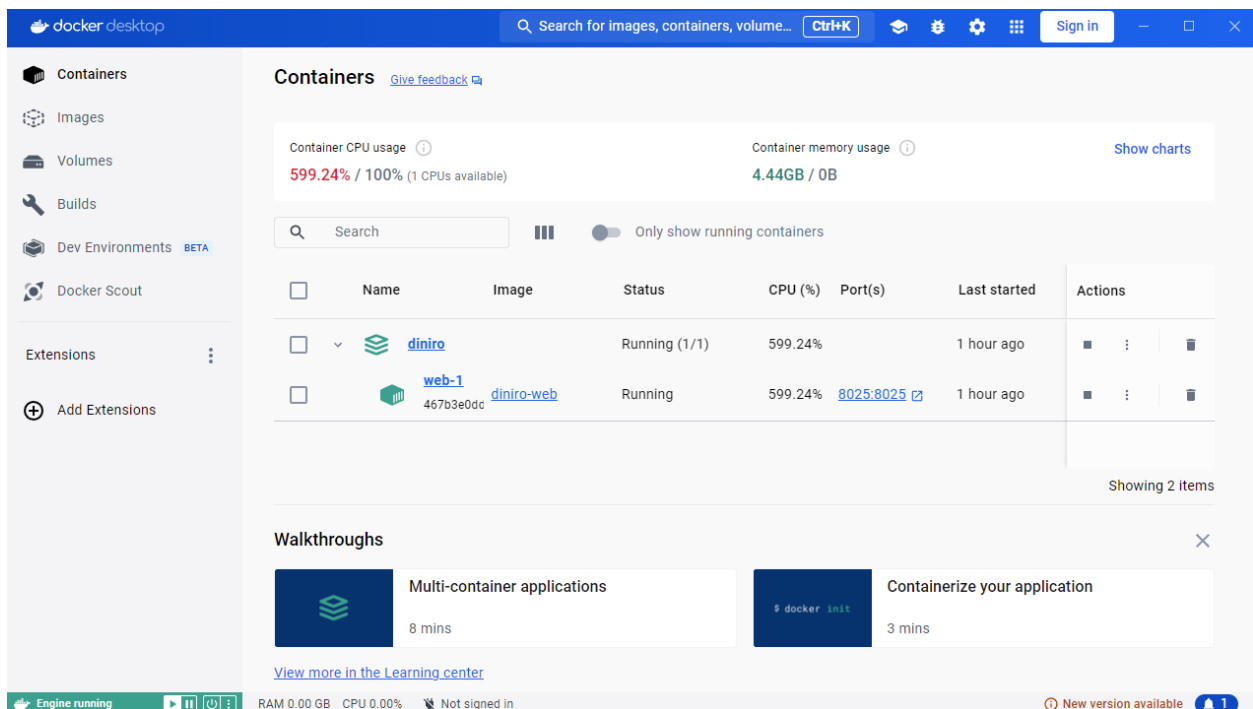
Created script to transfer the data under the name: **Transfer Data to H5AD Script.ipynb**.

Supplementary Note 4: Tool Computational Tasks fails

After the frontend worked successfully. I have tried to upload the transferred dataset to DiNiro web page. It was uploaded successfully and the metadata like plot type and plot color worked well. But after I start the tool analysis, a lot of errors started to show up because the code wasn't maintained.

1) Processor fails:

When trying on the full datasets and the Docker freezes:



Solution: work with 500 Genes datasets

2) Missing files:

When the tool starts the analysis, it start with the script compute_GRN.py script and I got this error:

```
web-1| FileNotFoundError: [Errno 2] No such file or directory:
'/code2/test_interface/static/GRN_files/human/motifs-v9-nr.hgnc-m0.001-o0.0.tbl'
web-1| [20/May/2024 20:23:24] "POST /result/ HTTP/1.1" 500 185202
```

Solution: find this motifs files “motifs-v9-nr.hgnc-m0.001-o0.0.tbl” and add it to the right directory. So, I have checked the author GitLab Repo but I couldn’t find it so I started searching online and I found it provided in this link: <https://resources.aertslab.org>

Then I got this error:

```
web-1| df2regulons
web-1| Internal Server Error: /result/
web-1| Traceback (most recent call last):
web-1| File "/usr/local/lib/python3.9/site-packages/django/core/handlers/exception.py", line 47, in inner
web-1| response = get_response(request)
web-1| File "/usr/local/lib/python3.9/site-packages/django/core/handlers/base.py", line 179, in
web-1| _get_response
web-1| response = wrapped_callback(request, *callback_args, **callback_kwargs)
web-1| File "/code2/test_interface/views.py", line 414, in result
web-1| ComputeGrn.get_matrices(userrr, n_sub, s_size, TFs_type)
web-1| File "/code2/test_interface/tasks/compute_GRN.py", line 187, in get_matrices
web-1| PathsG.paths_generator(sample_1_data_holder, data_folder_sample_1, path_to_TFs, path_to_feather,
web-1| f_motif_path)
web-1| File "/code2/test_interface/tasks/scGRN.py", line 121, in paths_generator
web-1| res.append(scGRN.GRN_function(path_to_TFs, path_to_data, path_output_file_csv, path_to_matrix_csv,
web-1| File "/code2/test_interface/tasks/scGRN.py", line 92, in GRN_function
web-1| regulons = df2regulons(df)
web-1| File "/usr/local/lib/python3.9/site-packages/pyscenic/transform.py", line 398, in df2regulons
web-1| assert not df.empty, 'Signatures dataframe is empty!'
web-1| AssertionError: Signatures dataframe is empty!
```

First, the problem is with df2regulons which is the function responsible of extracting regulons and it get an empty data frame which lead to this error but after some debugging, there was another missing file which is:

“hg38__refseq-r80__10kb_up_and_down_tss.mc9nr.genes_vs_motifs.rankings.feather”

Solution: find this file and add it to the right directory.

3) Copula Function:

After solving the last problem, DiNiro continues the calculations until it faces another error in the function_two.py script and p_value.py script.

```
web-1| ValueError: errors in input dimensions:
web-1|         expected data to have dimension 2 but got 0 instead
```

This error was resulting from the line code:

```
web-1| cop1 = EmpiricalCopula(df1.shape[1], pseudo_obs(df1), ties= 'max')
```

And after a lot of debugging, I found that the input data has the right dimension which is 2 not 0 but the problem was resulting because the EmpiricalCopula takes a matrix as a first attribute but the first attribute was “df1.shape[1]” which will be an integer value.

Solution: remove the first attribute in the EmpiricalCopula function.

The problem solved but I got another error which is:

```
web-1| ValueError: Input array must have the same dimension as copula
```

This error was resulting from the line code:

```
web-1| ec1 = cop1.cdf(pseudo_obs(df1))
```

After a lot of debugging, the arrays had the same dimension as copula but the problem results when comparing tf with gene that have the same name and same values so it consider it as 1 dimension instead of 2.

The only way to solve it is to skip the comparison where tf = gene but it affects the results. So, I contacted the tool developer and asked for the last maintained code but I have no answer from him.

Then I have re-implemented the EmpiricalCopula and CDF function to solve this problem.