Jun 19, 2024

LIANA+_Integrating_MultiOmics_Data

DOI
dx.doi.org/10.17504/protocols.io.eq2lyw6orvx9/v1

Daniel Dimitrov[1], Philipp Sven Lars Schäfer[1], Elias Farr[1], Pablo Rodriguez-Mier[1], Sebastian Lobentanzer[1], Pau Badia-i-Mompel[1,2], Aurelien Dugourd[1], Jovan Tanevski[1], Ricardo Omar Ramirez Flores[1], Julio Saez-Rodriguez[1]

[1]Heidelberg University, Faculty of Medicine, and Heidelberg University Hospital, Institute for Computational Biomedicine, Heidelberg, Germany;
[2]GSK, Cellzome, Heidelberg, Germany

Julio Saez-Rodriguez: pub.saez@uni-heidelberg.de;

D Daniel Dimitrov
Heidelberg University

DOI: dx.doi.org/10.17504/protocols.io.eq2lyw6orvx9/v1

External link: https://liana-py.readthedocs.io/en/latest/notebooks/sma.html

Protocol Citation: Daniel Dimitrov, Philipp Sven Lars Schäfer, Elias Farr, Pablo Rodriguez-Mier, Sebastian Lobentanzer, Pau Badia-i-Mompel, Aurelien Dugourd, Jovan Tanevski, Ricardo Omar Ramirez Flores, Julio Saez-Rodriguez 2024. LIANA+_Integrating_MultiOmics_Data. **protocols.io https://dx.doi.org/10.17504/protocols.io.eq2lyw6orvx9/v1**

Manuscript citation:

Protocol status: Working
We use this protocol and it's working

Created: June 17, 2024

Last Modified: June 19, 2024

Protocol Integer ID: 101923

Keywords: multi-omics, ccc, python

Funders Acknowledgement:
ITN Strategy-CKD
Grant ID: 860329
ITN DECIDER
Grant ID: 965193
ITN PerMedCoE
Grant ID: 951773
Deutsche Forschungsgemeinschaft
Grant ID: SPP 2395
Deutsche Forschungsgemeinschaft
Grant ID: CRC 1550

Abstract

A Protocol describing the application of LIANA+ on a spatially-resolved metabolite-transcriptome dataset from a recent murine Parkinson's disease model **Vicari et al., 2023**.

We demonstrate LIANA+'s utility in harmonizing spatially-resolved transcriptomics and MALDI-MSI data to unravel metabolite-mediated interactions and the molecular mechanisms of dopamine regulation in the striatum.

Two particular challenges with this data are: - The unaligned spatial locations of the two omics technologies and the untargeted nature of the MALDI-MSI data, which results in a large number of features with unknown identities. Only few of which were previously identified as specific metabolites.

Here, we show untargeted modelling of known and unknown metabolite peaks and their spatial relationships with transcriptomics data. Specifically, we use a multi-view modelling strategy **MISTy** to decipher global spatial relationships of metabolite peaks with cell types and brain-specific receptors. Then, we use LIANA+'s **local metrics** to pinpoint the subregions of interaction. We also show strategies to enable spatial multi-omics analysis from diverse omics technologies with unaligned locations and observations.

This protocol is associated with our Nature Cell Biology paper describing LIANA+.

Materials

Hardware:
A desktop or laptop computer with at least 8GB RAM.

Software:
Python version 3.8 or newer.
liana-py (installable via pip or github) version 1.2.0 or newer.
numpy version < 2.0

Safety warnings

Please ensure numpy <= 2.0 is installed in any new environment, as some packages might misbehave due to the recent update.

Before start

Install *Python*. We recommend using conda, *miniconda, or mamba* for this.

Install LIANA+

1
```
# In command line / or jupyter notebook
pip install liana[common]
pip install adjustText==0.8
```

Import Required Packages

2
```
import numpy as np
import liana as li
import mudata as mu
import scanpy as sc
from matplotlib import pyplot as plt
from adjustText import adjust_text
```

3
```
# save figure parameters
kwargs = {'frameon':False, 'size':1.5, 'img_key':'lowres'}
```

Obtain and Examine the Data

4
```
# let's download the data
rna = sc.read("sma_rna.h5ad", backup_url="https://figshare.com/ndownloader/files/44624974?
private_link=4744950f8768d5c8f68c")

msi = sc.read("sma_msi.h5ad",
backup_url="https://figshare.com/ndownloader/files/44624971?
private_link=4744950f8768d5c8f68c")

ct = sc.read("sma_ct.h5ad", backup_url="https://figshare.com/ndownloader/files/44624968?
private_link=4744950f8768d5c8f68c")
```

5
```
# and create a MuData object
mdata = mu.MuData({'rna':rna, 'msi':msi, 'ct':ct})
# examine the object
mdata
```

6
```
# examine data
fig, axes = plt.subplots(1, 3, figsize=(12, 3))

sc.pl.spatial(rna, color='log1p_total_counts', ax=axes[0], **kwargs, show=False)
sc.pl.spatial(msi, color='Dopamine', cmap='magma', ax=axes[1], **kwargs, show=False)
sc.pl.spatial(ct, color='MSN1', cmap='viridis', ax=axes[2], **kwargs, show=False)
```

```
fig.subplots_adjust(wspace=0, hspace=0)
fig.tight_layout()
```

7
```
# examine experimental design
sc.pl.spatial(rna, color=['lesion', 'region'], **kwargs, wspace=0.25)
```

Remove Features with little-to-no variation

8
```
sc.pp.highly_variable_genes(rna, flavor='cell_ranger', n_top_genes=5000)
sc.pp.highly_variable_genes(msi, flavor='cell_ranger', n_top_genes=150)
ct.var['cv'] = ct.X.A.var(axis=0) / ct.X.A.mean(axis=0)
ct.var['highly_variable'] = ct.var['cv'] > np.percentile(ct.var['cv'], 20)

msi = msi[:, msi.var['highly_variable']]
rna = rna[:, rna.var['highly_variable']]
ct = ct[:, ct.var['highly_variable']]

# scale msi intensities
sc.pp.scale(msi, max_value=5)
```

Obtain brain-specific metabolite-receptor interactions

9
```
metalinks = li.rs.get_metalinks(tissue_location='Brain',
                                biospecimen_location='Cerebrospinal Fluid (CSF)',
                                source=['CellPhoneDB', 'NeuronChat']
                                )
metalinks.head()
```

10
```
# Obtain Human to Murine gene symbols Map
map_df = li.rs.get_hcop_orthologs(columns=['human_symbol', 'mouse_symbol'],
                                  min_evidence=3
                                  ).rename(columns={'human_symbol':'source',
                                                    'mouse_symbol':'target'})
```

11
```
metalinks = li.rs.translate_column(resource=metalinks,
                                   map_df=map_df,
                                   column='gene_symbol',
                                   one_to_many=1)
metalinks.head()
```

12
```
# Intersect the RNA modality with the receptors
receptors = np.intersect1d(metalinks['gene_symbol'].unique(), rna.var_names)
rec = rna[:, receptors].copy()
```

Compute Spatial Proximies for the Multi-view Model

13
```
# We use the metabolite modality as a reference to which we align the other modalities.
# We use the `spatial_neighbors` function to compute the spatial proximity from cell types and brain-specific receptors to the metabolite intensities.

plot, _ = li.ut.query_bandwidth(coordinates=rna.obsm['spatial'], start=0, end=1500)
plot
```

14
```
# choose bandwidth
bandwidth = 500
cutoff = 0.1
# distances of metabolties to RNA
reference = mdata.mod["msi"].obsm["spatial"]
```

15
```
# Compute proximities to the reference for each modality
li.ut.spatial_neighbors(ct, bandwidth=bandwidth, cutoff=cutoff, spatial_key="spatial", reference=reference, set_diag=False, standardize=False)
li.ut.spatial_neighbors(rec, bandwidth=bandwidth, cutoff=cutoff, spatial_key="spatial", reference=reference, set_diag=False, standardize=False)
```

Construct and Run the Multi-view model

16
```
# MISTy
mdata.update_obs()
misty = li.mt.MistyData({"intra": msi, "receptor": rec, "ct": ct}, enforce_obs=False, obs=mdata.obs)
misty
```

17
```
# Learn the relationships between intra and the extra views
misty(model=li.mt.sp.LinearModel, verbose=True, bypass_intra=True, maskby='lesion')
```

Examine Results

18
```
li.pl.target_metrics(misty, stat='multi_R2', return_fig=True, top_n=20, filter_fun=lambda x: x['intra_group']=='intact')
```

19
```
interactions = misty.uns['interactions']

interactions = interactions[(interactions['intra_group'] == 'intact') & (interactions['target'] == 'Dopamine')]
# Create scatter plot
plt.figure(figsize=(5, 4))
# rank rank by abs importances
```

```
interactions['rank'] = interactions['importances'].rank(ascending=False)
plt.scatter(interactions['rank'], interactions['importances'], s=11,
        c=interactions['view'].map({'ct': '#008B8B', 'receptor': '#a11838'}))

# add for top 10
top_n = interactions[interactions['rank'] <= 10]
texts = []
for i, row in top_n.iterrows():
   texts.append(plt.text(row['rank'], row['importances'], row['predictor'], fontsize=10))
adjust_text(texts, arrowprops=dict(arrowstyle="->", color='grey', lw=1.5))
plt.tight_layout()
```

Identifying Local Interactions

20
```
# Focusing on Dopamine, we can next use LIANA+'s local metrics to identify the subregions of interactions with MSN1/2 cells.
# to do so, we need to interpolate one modality to the other, such that they are on the same coordinate system, while this is done internally for the multi-view learning, we need to interpolate it as:
metabs = li.ut.interpolate_adata(target=msi, reference=rna, use_raw=False, spatial_key='spatial')
```

21
```
# let's rebuild this with the update modalities:
mdata = mu.MuData({'msi': metabs, 'rna':rna, 'deconv':ct}, obsm=rna.obsm, obs=rna.obs, uns=rna.uns)
```

22
```
# re-calculate neighbours
li.ut.spatial_neighbors(mdata, bandwidth=bandwidth, cutoff=cutoff, set_diag=True)
```

23
```
# define interactions of interest
interactions = metalinks[['metabolite', 'gene_symbol']].apply(tuple, axis=1).tolist()
```

24
```
# Let's calculate the local metrics for the Dopamine intensities with and Drd1/2 receptors.

li.mt.bivariate(mdata,
        local_name='cosine',
        x_mod='msi',
        y_mod='rna',
        key_added='lr',
        x_use_raw=False,
        y_use_raw=False,
        verbose=True,
        mask_negatives=True,
```

```
    n_perms=1000,
    interactions=interactions,
    x_transform=sc.pp.scale,
    y_transform=sc.pp.scale,
)
```

Plot Bivariate Similarity between Dopamine and Drd1/2 receptors

```
sc.pl.spatial(mdata.mod['lr'],
    color=['Dopamine^Drd1', 'Dopamine^Drd2'],
    cmap='cividis_r', vmax=1, layer='pvals',
    **kwargs)
```

Protocol references

Vicari, M. et al. Spatial multimodal analysis of transcriptomes and metabolomes in tissues. Nat. Biotechnol. (2023) doi:10.1038/s41587-023-01937-y.