imaging-transcriptomics

Release 1.1.8

Alessio Giacomel, Daniel Martins

Mar 17, 2022

CONTENTS:

CHAPTER

ONE

GETTING STARTED

Once the tool is installed you can run the analysis by calling the script from the terminal as:

```
imagingtranscriptomics -i path-to-your-file.nii --no-gsea pls --ncomp 1
```

This is the most simple way to run the script and will permorm the analysis with 1 PLS component on your file and save the results in a folder named *Imt_file_name* in the same path as the original scan file. It might be that running this will not hold much of the total variance of the scan, however this can be used as a "first quick estimation". In the resulting path there will be a plot with the variance explained by the first 15 components independently and cumulatively, that can be used to tune consequent analyses, if needed.

For more information on the use have a look at the *usage* page. You can also have a deeper look at the *methods* and on *what to do with the results from the script*.

For more advanced use, or to integrate it in your python workflow, you can use the *python module*.

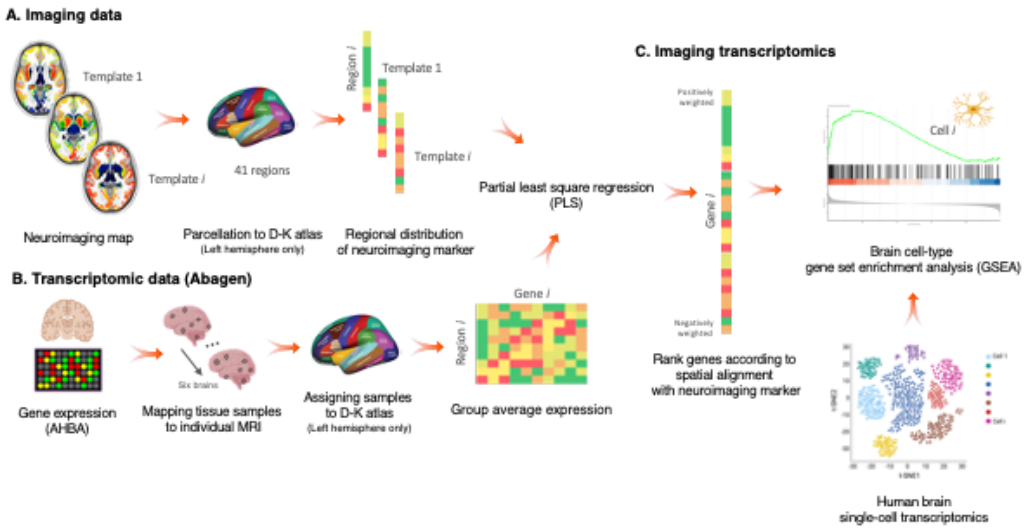
CHAPTER

TWO

WHAT IS IMAGING TRANSCRIPTOMICS?

Imaging transcriptomics is a methodology that allows to identify patterns of correlation between gene expression and some property of brain structure or function as measured by neuroimaging (e.g., MRI, fMRI, PET).

An overview of the methodology can be seen in the figure below.



In brief, average values of the scan are extracted from 41 brain regions as defined by the Desikan-Killiany (DK) atlas. Regional values are then used to perform partial least squares (PLS) regression with gene expression data from the Allen Human Brain Atlas (AHBA) mapped to the DK atlas, in the left hemisphere only.

As a result of the PLS regression we obtain the ranked genes list according to the spatial alignment with the neuroimaging marker of interest.

See also:

For a more comprehensive dive into the methodology have a look at our paper: Imaging transcriptomics: Convergent cellular, transcriptomic, and molecular neuroimaging signatures in the healthy adult human brain. *Daniel Martins, Alessio Giacomel, Steven CR Williams, Federico Turkheimer, Ottavia Dipasquale, Mattia Veronese, PET templates working group*. Cell Reports; doi: https://doi.org/10.1016/j.celrep.2021.110173

2.1 Allen Human Brain Atlas

The Allen Human Brain Atlas (AHBA) freely available multimodal atlas of gene expression and anatomy comprising a comprehensive 'all genes–all structures' array-based dataset of gene expression and complementary *in situ hybridization* (ISH) gene expression studies targeting selected genes in specific brain regions. Available via the Allen Brain Atlas data portal (www.brain-map.org), the Atlas integrates structure, function, and gene expression data to accelerate basic and clinical research of the human brain in normal and disease states.

The `imaging-transcriptomics` script uses a modified version of the AHBA gene data parcellated onto 83 regions from the DK atlas obtained using the abagen toolbox. In brief, probes that cannot be reliably matched to genes were discarded and filtered based on their intensity compared to the background noise level. The remainig probes were pooled retaining only the one with the highest differential stability to represent each gene, resulting in 15,633 probes each representing an unique gene. The genes were then assigned to brain regions based on their corrected MNI coordinates.

More details on the processing of the transcriptomic data are available in the methods section of the paper .

2.2 Desikan-Killiany Atlas

The DK atlas is a parcellation atlas of the human brain, which includes both cortical and subcortical regions.

This atlas is derived from a dataset of 40 MRI scans where 34 cortical ROIs were manually delineated for each of the individual hemispheres. More details on the ROIs of the atlas or methods to derive it refer to the original paper.

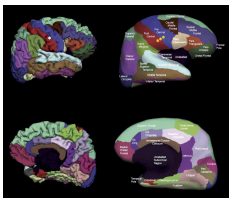


Fig. 1: Representation of the pial and inflated view of the cortical regions from the Desikan-Killiany atlas. Image from the orignal paper

2.3 Partial least squares

The goal of any regression is to model the relationship between a target variable and multiple explanatory variables. The standard approach is to use Ordinary Least Squares (OLS), but in order to use OLS the assumptions of linear regression have to be met. The assumptions of linear regression are:

- Independence of observations
- No hidden or missing variables
- Linear relationship
- Normality of the residuals
- No or little multicollinearity
- Homoscedasticity
- All independent variables are uncorrelated with the error term
- Observations of the error term are uncorrelated with each other

In some cases it can be that we have a lot of independent variables, many of which are correlated with other independent variables, violating thus the assumption of no multicollinearity. In this case instead of using OLS a more appropriate method is to use Partial Least Squares (PLS) Regression. This method allows to reduce the dimensionality of correlated variables and model the underlying information shared.

References

Imaging transcriptomics: Convergent cellular, transcriptomic, and molecular neuroimaging signatures in the healthy adult human brain. *Daniel Martins, Alessio Giacomel, Steven CR Williams, Federico Turkheimer, Ottavia Dipasquale, Mattia Veronese, PET templates working group*. bioRxiv 2021.06.18.448872; doi: https://doi.org/10.1101/2021.06.18.448872

The Allen Human Brain Atlas: Comprehensive gene expression mapping of the human brain. *Elaine H. Shein, Caroline C. Overly, Allan R. Jones*, Trends in Neuroscience vol. 35, issue 12, December 2012; doi: https://doi.org/10.1016/j.tins.2012.09.005

An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *Rahul S.Desikan, Florent Ségonne, Bruce Fischl, Brian T. Quinn, Bradford C. Dickerson, Deborah Blacker, Randy L. Buckner, Anders M. Dale, R. Paul Maguire, Bradley T. Hyman, Marilyn S. Albert, Ronald J. Killiany*, NeuroImage, Volume 31, Issue 3, July 2006; doi: https://doi.org/10.1016/j.neuroimage.2006.01.021

An Introduction to Partial Least Squares Regression. *R. Tobias*, https://stats.idre.ucla.edu/wp-content/uploads/2016/02/pls.pdf

Comparison of prediction methods for multicollinear data, *T. Naes and H. Martens*, Communications in Statistics, Simulation and Computation, 14(3), 545-576.

SIMPLS: An alternative approach to partial least squares regression. *Sijmen de Jong*, Chemometrics and Intelligent Laboratory Systems, March 1993, doi: https://doi.org/10.1016/0169-7439(93)85002-X

Gene transcription profiles associated with inter-modular hubs and connection distance in human functional magnetic resonance imaging networks. *Petra E. Vértes, Timothy Rittman, Kirstie J. Whitaker, Rafael Romero-Garcia, František Váša, Manfred G. Kitzbichler, Konrad Wagstyl, Peter Fonagy, Raymond J. Dolan, Peter B. Jones, Ian M. Goodyer, the NSPN Consortium and Edward T. Bullmore*, Philosophical Transactions of the Royal Society B, October 2016, doi: https://doi.org/10.1098/rstb.2015.0362

CHAPTER

THREE

INSTALLATION

To install the `imaging-transcriptomics` Python package you must first of all have Python `v3.6+` installed on your system along with the `pip` package manager.

Tip: We suggest installing the package in a dedicated python environment using venv or conda depending on your personal choice. The installation on a dedicated environment avoids the possible clashes of dependencies after or during installation.

Note: All following steps assume that, if you have created a dedicated environment, this is currently active. If you are unsure you can check with `which python` from your terminal or activate your environment via the `source activate` (for conda managed environments) or `source venv/bin/activate` (for venv managed environments).

Before installing the `imaging-transcriptomics` package we need to install a package that is not available through PyPi but from GitHub only. This package is pypls and is used in the script to perform all PLS regressions. In order to install it you can run the following command from your terminal

```
pip install -e git+https://github.com/netneurolab/pypyls.git/#egg=pyls
```

This will install install the GitHub repository directly using pip and it will make it available with the name `pyls`.

Warning: Do not install pyls directly from pip with the command `pip install pyls` as this is a completely different package!

A second package to install for the full functionalities of the imaging-transcriptomics toolbox is the ENIGMA toolbox . To install this we'll follow the instructions of the developers. In brief, install this by running the commands:

```
git clone https://github.com/MICA-MNI/ENIGMA.git
cd ENIGMA
python setup.py install
```

Once these packages is installed you can install the `imaging-transcriptomics` package by running:

```
pip install imaging-transcriptomics
```

Once you get the message that the installation has completed you are set to go!

Note: The version `v1.0.0` and `v1.0.1`, can cause some issues on the installation due to compatibility issues of some packages. In version `v1.0.2+` this issue has been resolved during installation. If you have one of

the older versions installed you might want to update the version using the command `pip install --upgrade imaging-transcriptomics`.

Note: From version `v1.1.0` has the possibility of running directly from the toolbox also the gene set enrichment analysis (GSEA). Version `v1.1.8` has a major speedup in the correlation analyses, reducing the overall time needed to run the analysis.

CHAPTER

FOUR

SCRIPT USAGE

Once you have installed the package you can run the analysis script as:

```
imagingtranscriptomics --input /path-to-your-in-file [options] {corr|pls [options]}
```

The script has some options that allow the user to tune the analysis to their specific application. The options are as follows:

- `--input` (`-i`, **mandatory**): path to the input data. This can be either a neuroimaging scan (*i.e.*, .nii[.gz]) or a text file (*i.e.*, .txt).

  Warning: If the input scan is a neuroimaging scan (*i.e.*, .nii, .nii.gz) this is expected to be in the same resolution as the Desikan-Killiany (DK) atlas used which is 1mm isotropic (matrix size 182x218x182). On the other hand if the input is a text file, this must be a text file with one column and no headers, with the rows containing the values of interest in the same order as the DK atlas used.

- `--output` (`-o`, **optional**): path to the output directory. If none is provided the results will be saved in the same folder as the input scan.
- `--regions` (`-r`, **optional**): regions to use for the analysis, can be *cort+sub* (or equivalently *all*) which specifies that all the regions are used, or, alternatively, *cort* for the cortical regions only. The latter is useful with some certain types of data, where the subcortical regions might not be available (*e.g.*, EEG).
- `--no-gsea` (**optional**): specifies whether or not Gene Set Enrihment Analysis should be performed.
- `--geneset` (**optional**): specifies the name of the gene set or the path to the file to use for Gene Set Enrichment Analysis.

  Warning: The `--geneset` argument will be ignored if you also specify the `--no-gsea` flag. If the GSEA analysis is performed, the name of the gene set, or a path to a custom made gene set, should be given. To lookup the name of the available gene sets or on how to create a custom one refer to the GSEA section.

After the selection of the above options, you can now specify the type of analysis to perform. The available analyses are:

- `corr`: to perform mass univariate correlation analysis using Spearman's rank correlation.
- `pls`: to perform partial least squares (PLS) analysis. If you select this option you must additionally specify either the number of components to use in the analysis, with the `--ncomp` option, or the amount of variance to retain from the data, with the `--var` option.

Tip: All paths given as input should be given as absolute paths instead of relative paths to avoid any errors in reading the file.

The `imagingtranscriptomics` script allows the user to specify the options to perform also the GSEA analysis, directly after the correlation analysis. However, it is not uncommon that on the same imaging data a researcher might have different research questions, which results in different gene sets to use for the investigation. For this reason, in the toolbox there is an additional script that, once a first correlation analysis is performed, allows to run directly the GSEA analysis. This script can be invoked as:

```
imt_gsea --input /path-to-your-in-file [options]
```

The running of this script is pretty straightforward, and the options available are:

- `--input` (`-i`, **mandatory**): path to the input data. To run this script you must have already have performed a correlation analysis, either with mass univariate correlation or with PLS, as the input file is one of the output files of the previous step. The required file is located in the output folder and has `.pkl` extension.

Warning: The `--input` argument **MUST** be a `.pkl` file generated by running the `imagingtranscriptomics` script.

In addition to the `--input` argument, the script has the following options:

- `--output` (`-o`, **optional**): path to the output directory. If none is

provided the results will be saved in the same folder as the input file. - `--geneset` (**optional**): specifies the name of the gene set or the path to the file to use for Gene Set Enrichment Analysis. If you want to use one of the provided gene sets you can browse the available ones by running the script with only the `--geneset avail` option.

Tip: To see the gene sets available in the package, run the script with the `--geneset avail` option, i.e. `imt_gsea --geneset avail`.

CHAPTER

FIVE

USAGE AS PYTHON LIBRARY

Once installed the library can be used like any other Python package in custom written analysis pipelines. To the library can be imported by running:

```
import imaging_transcriptomics as imt
```

Once imported the package will contain the core `ImagingTranscriptomics` class, along with other useful functions. To see all the available functions imported in the library run:

```
dir(imt)
```

which will display all the functions and modules imported in the library.

5.1 ImagingTranscriptomics Class

The `ImagingTranscriptomics` class is the core class of the entire package and allows you to run the entire analysis on your data. To use the class you simply need to initialise it and then run the `.run()` method.

To initialise the class, you will need to already have decided the type of correlation analysis to perform, as this will be needed as initialisation keyword for the class. The initialisation of the class can be done as follows:

```
# To initialise the class with PLS analysis
analysis = imt.ImagingTranscriptomics(my_data,
            method="pls",
            n_components=1)

# To initialise the class with mass univariate correlation analysis
analysis = imt.ImagingTranscriptomics(my_data,
            method="corr")
```

In the above code snippets the `my_data` argument is a `numpy.ndarray` vector with the imaging data of interest (e.g. the mean intensity of the ROI). The vector **MUST** be a vector with either 35 or 41 elements, corresponding to the number of ROIs in the left hemisphere of the brain (35 for the cortical regions and the remaining for the subcortical regions).

There are addiotional parameters that can be used for the initialisation of the class, which are:

- `method` (`"pls"` or `"corr"`, **mandatory**): specifies the type of analysis to perform.
- `n_components` (`int`, **optional**): specifies the number of components to use for the PLS analysis.
- `var` (`float`, **optional**): specifies the variance explained threshold to use for the PLS analysis.

- `regions`: specifies if the analysis should be performed on the cortical regions only or on the whole brain. The possible values are: `"cort+sub"` (or `"all"`) or `"cort"`.

Once the class is initialise you can run the analysis by running the `.run()` method.

```
analysis.run()
```

The method has some additional parameters that can be used to run the method. Some of the parameters are:

- `gsea`: `bool` variable to indicate whether the GSEA analysis should be run.
- `gene_set`: `str` variable to indicate the gene set to use for the GSEA analysis.
- `outdir`: `str` variable to indicate the output directory.
- `scan_name`: `str` variable to indicate the name of the scan to use to save the results.
- `save_res`: `bool` variable to indicate whether the results should be saved. Default is `True`.
- `gene_limit`: number of genes to use for the GSEA analysis. Default is `500`.

Once the correlation analysis is completed, the results can be accessed in the `analysis.gene_results` attribute. If you want to perform the GSEA analysis after the correlation, or on a second gene set, you can run the `analysis.gsea()` method. The method has the following parameters:

- `gene_set`: `str` variable to indicate the gene set to use for the GSEA analysis.
- `outdir`: `str` variable to indicate the output directory.
- `gene_limit`: number of genes to use for the GSEA analysis. Default is `500`.

It is to note that since in most cases the analysis is performed having as inputs either a neuroimaging scan (i.e., a .nii or .nii.gz file) or a txt file with some measure of interest (e.g., measures extracted using Freesurfer), we also included two additional methods to initialise the class which are:

```
analysis = imt.ImagignTranscriptomics.from_scan(my_scan,
            method="corr")
```

to initialise the class from a scan, extracting the average from the regions, and:

```
analysis = imt.ImagignTranscriptomics.from_file(my_txt_file,
            method="corr")
```

These methods allow you to initialise the class from a scan or a txt file respectively. In both cases the input is a path to the file of interest, while the rest of the input parameters are the same as the initialisation of the normal class explained above.

CHAPTER

SIX

EXAMPLES

Coming soon…

CHAPTER

SEVEN

CONTRIBUTING

If you want to contribute to the `imaging_transcriptomics` python package or script you can clone the GitHub repo and change/add whatever you feel appropriate. Once you want to merge your changes to the project you can request a pull request to the `develop` branch. Please note that we only accept pull requests to the `develop` branch.

7.1 General guidelines for contributing

If you want to contribute to the project there are some general guidelines we ask you to follow in order to maintain a certain level of consistency:

- When you write some functionality you MUST document that functionality with docstrings. The docstrings should include a description of the functionality along with a description of the parameters of the function and returns using the `:param:` and `:return:` parameters.
- All your code SHOULD be compliant with the PEP8 python styling guide.

CHAPTER

EIGHT

CONTRIBUTOR COVENANT CODE OF CONDUCT

8.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to make participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

8.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

8.3 Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

8.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

8.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at hs@ox.cx. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

8.6 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at <https://www.contributor-covenant.org/version/1/4/code-of-conduct.html>.

CHAPTER

NINE

HOW TO CITE AND GET IN TOUCH

9.1 Contact us

We are happy to answer any questions you might have about the methods and/or problems/suggestions with the software.

For any questions regarding the methodology you can contact Dr Daniel Martins, or the senior authors of the paper Dr Ottavia Dipasquale and Dr Mattia Veronese.

For questions about the software you can contact Alessio Giacomel or any of the authors above.

See also:

For questions regarding the software you can also check out our *FAQ* section or open a new issue on GitHub.

9.2 Cite our work

If you use our software or methods in your research please cite our work:

- **Imaging transcriptomics: Convergent cellular, transcriptomic, and molecular neuroimaging signatures in the healthy adult human brain.** *Daniel Martins, Alessio Giacomel, Steven CR Williams, Federico Turkheimer, Ottavia Dipasquale, Mattia Veronese, PET templates working group*. bioRxiv 2021.06.18.448872; doi: https://doi.org/10.1101/2021.06.18.448872
- **Imaging-transcriptomics: python package (v1.0.0).** Alessio Giacomel, Daniel Martins. Zenodo 2021. https://doi.org/10.5281/zenodo.5507506

For more information about ongoing research please visit our website at: https://molecular-neuroimaging.com

CHAPTER

TEN

FAQ

1. **How can I install the imaging transcriptomics package?** The short answer is: you can install it via `pip`. For more details on how to install refer to the *installation section.*
2. **Why does the analysis use only the left hemisphere?** The analysis relies on the left hemisphere only due to the genetic data used. The Allen Human Brain Atlas (AHBA) has a discrepancy in data acquisition between left and right hemisphere resulting in a lot of missing data in the right hemisphere. Given that the brain is not symmetrical, we decided to not mirror data from one hemisphere to the other and constrain the analysis to this hemisphere only.
3. **Why did you use the pypls library instead of some more maintained PLS library, e.g., sklearn?** We used pypls instead of sklearn because the latter one, and most of the other available, are implemented using the NIPALS algorithm, while pypls uses the SIMPLS. One of the main advantages of the SIMPLS algorithm in respect to the NIPALS is that is is less time consuming.
4. **Can I run the ImaginTranscriptomics analysis on just the cortical areas without the subcortical areas?** Yes, check out the main page on the use to get an idea on how to do this.

CHAPTER

ELEVEN

INDICES AND TABLES

- genindex
- modindex
- search