
imaging-transcriptomics

Release 1.1.0

Alessio Giacomel, Daniel Martins

Jan 11, 2022

CONTENTS:

1	Getting started	1
2	What is imaging transcriptomics?	3
2.1	Allen Human Brain Atlas	4
2.2	Desikan-Killiany Atlas	4
2.3	Partial least squares	4
3	Installation	7
4	Script usage	9
5	Usage as python library	11
5.1	ImagingTranscriptomics Class	11
5.2	Other Functions of Imaging_Transcriptomics	12
6	What to do with your results	13
7	Contributing	15
7.1	General guidelines for contributing	15
8	Contributor Covenant Code of Conduct	17
8.1	Our Pledge	17
8.2	Our Standards	17
8.3	Our Responsibilities	17
8.4	Scope	18
8.5	Enforcement	18
8.6	Attribution	18
9	How to cite and get in touch	19
9.1	Contact us	19
9.2	Cite our work	19
10	FAQ	21
11	Indices and tables	23
	Bibliography	25

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 -n 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.

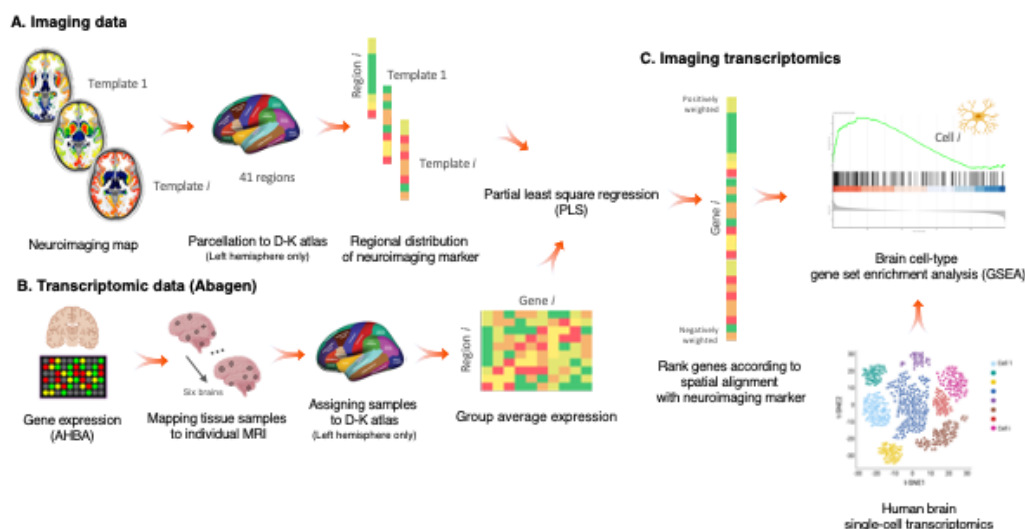
For more information on the use have a look at the [usage](#) page. You can also have a deeper look at the [methods](#) and on [what to do with the results from the script](#).

For more advanced use, or to integrate it in your python workflow, you can use the [python module](#).

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](https://doi.org/10.1101/2021.06.18.448872). 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>

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 [Shein2012].

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 remaining 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 [Martins2021].

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 [Desikan2006]

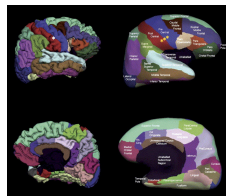


Fig. 1: Representation of the pial and inflated view of the cortical regions from the Desikan-Killiany atlas. Image from the [original paper](#) [Desikan2006]

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

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/pypls.git/#egg=pypls
```

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

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

Once this package 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 --upgragde imaging-transcriptomics`.

SCRIPT USAGE

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

```
imagingtranscriptomics -i path_to_file -n 2
```

The script has some parameters that can be tuned according to the necessity of the analysis, which can be viewed by calling the help function from the script as `imagingtranscriptomics -h` or `imagingtranscriptomics --help`.

Here we will describe in more details all the parameters of the script. The parameters of the script are:

`-i (--input)` Scan on which you want to perform the analysis. It is *recommended* that you provide an absolute path to your scan (e.g., `~/Desktop/myscan.nii.gz`) instead of a relative one (e.g., `myscan.nii.gz`) to avoid errors. The input file *must* be an imaging file in NIfTI format (both `.nii` and `.nii.gz` formats are supported), be in MNI152 space and have a matrix dimension of 182x218x182. If your image has a different matrix size you can reslice it to match this dimension with your preferred method. (A quick method is to use `fslview` and `reslice` to match the dimension of the included *MNI152_1mm* brain atlas).

Warning: The input scan must have a predefined dimension (**182x218x182**) and be in **MNI152** space. If the input scan is not in the required dimension the script will throw an error. You should always check the dimension before running the script and eventually reslice or spatially normalise your image to the matching dimensions with your preferred method (e.g., SPM, FSL, ANTS).

`-n (--ncomp)` Number of PLS components to use for the analysis. The parameter *must* be an **integer** between 1 and 15, otherwise an error will occur. *Please note* that in PLS regression the first component is not necessarily the component explaining the most amount of variance, as in PCA. Example: running `imaging-transcriptomics -i path_to_file -n 2` will run the script on your imaging file selecting the two first components for the analysis.

`-v (--variance)` Total amount of variance you want your components to explain. The code will automatically select the number of components that explain at least the variance you specify. The parameter *must* be an **integer** between 10 and 100, which represents the percentage of explained variance. Example: if you run `imaging-transcriptomics -i path_to_file -v 30` and the first 3 components explain 10%, 25% and 3%, respectively, of the total variance the script will use 2 components, even if they explain 35% (which is a bit more than specified) of the total variance.

Warning: Please note that the `-v` and `-n` parameters are mutually exclusive and only one has to be provided as argument for the script, otherwise an error will occur.

`--corr` Allows to run the analysis using regression instead of PLS regression.

Note: If you use the `--corr` option you can avoid specifying all other parameters (e.g., `-v`, `-n`) as the script will ignore them anyway.

Optional additional parameters that can be provided are:

-o (--out) Path where you want to save the results, if no path is provided the results will be saved in the same path as the input scan. When the code is finished running the results will be saved in a folder named *Imt_myscannname* and will contain all the results (.csv files, .pdf report and images in .png format). If you run the script multiple times you will have more than one results folder with a trailing number for the run (e.g., *Imt_myscannname* for the first run and *Imt_myscannname_1* for the second run).

--verbose Sets the output logging level to debug mode and shows all debug values and steps

--suppress Sets the logging level to warning and will display only eventual warning messages.

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 start using the class the first step is to initialise it. A way to do this is:

```
my_analysis = imt.ImagingTranscriptomics(my_data, n_components=1)
```

The `my_data` is an array that contains the data you want to analyse (e.g., the average values from the scan). This vector has to have some characteristics, mainly:

- it has to be a `numpy.array` with length 41, which corresponds to the number of regions in the left hemisphere of the Desikan-Killiany atlas.
- it has to contain the values you want to analyse but not the `zscore` of the values as this is computed automatically during the initialisation.

Alternatively to initialise the class with the number of desired components, you can initialise the class by specifying the amount of variance that you want the components to explain. The software will then select the number of components that explains *at least* the specified amount (e.g., you specify a 60% of variance and one component explains 58% while the first two components combined explain 70%, two components will be selected).

```
my_analysis = imt.ImagingTranscriptomics(my_data, var=0.6)
# The amount of variance can be expressed in different ways and gets converted
# internally.
# The following will produce the same results as the above
my_analysis = imt.ImagingTranscriptomics(my_data, var=60)
```

Once the class is initialised, you can run the analysis by invoking the `.run()` method.

```
my_analysis.run()
```

There are currently two methods to run the analysis, the first uses PLS regression while the other uses Spearman correlation. The PLS analysis is the default method to analyse the data is PLS, while if you want to run the analysis with correlation you can run the command:

```
my_analysis.run(method="corr")
```

Note: Please be aware that running the correlation method is currently much slower than the PLS method. This is due the number of correlation that have to be ran during the permutation analysis. The code running these analysis is leveraging multiprocessing of the processor, by using as many cores of the CPU as possible, but even doing this times of *20min* are not uncommon.

Once the analysis is completed you can check you results by accessing the attributes of the class.

5.2 Other Functions of Imaging_Transcriptomics

The `imaging_transcriptomics` library contains several helpful functions, like:

- `read_scan`: that allows to read a NIfTI file and returns the data matrix (without any of the header information).
- `extract_average`: that allows to extract the average value from the left hemisphere of the scan.

WHAT TO DO WITH YOUR RESULTS

Once the script is finished running you can use the results to perform Gene Set Enrichment Analysis (GSEA). An example of tool that can be used to perform this analysis is the online tool [WEB-based GENE SeT AnaLysis Toolkit \(WEBGESTALT\)](#), which is the tool used for the analysis in our paper.

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 **SHOULD** also make a test using the `pytest` library and add it to the `tests/` folder.
- 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.

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](https://www.contributor-covenant.org/version/1/4/code-of-conduct.html), version 1.4, available at [<https://www.contributor-covenant.org/version/1/4/code-of-conduct.html>](https://www.contributor-covenant.org/version/1/4/code-of-conduct.html).

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>

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 it is less time consuming.
4. **Can I run the `ImaginTranscriptomics` analysis on just the cortical areas without the subcortical areas?** The short answer is maybe. We are currently working on an update that will allow the user to select if to use the cortical, subcortical or both. For now we are using both areas.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

BIBLIOGRAPHY

- [Martins2021] **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>
- [Shein2012] **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>
- [Desikan2006] **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>
- [Tobias1996] **An Introduction to Partial Least Squares Regression.** *R. Tobias,* <https://stats.idre.ucla.edu/wp-content/uploads/2016/02/pls.pdf>
- [Naers1985] **Comparison of prediction methods for multicollinear data,** *T. Naes and H. Martens,* Communications in Statistics, Simulation and Computation, 14(3), 545-576.
- [deJong1993] **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](https://doi.org/10.1016/0169-7439(93)85002-X)
- [Vertes2016] **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>