



**HAL**  
open science

## pygamma-agreement: Gamma $\gamma$ measure for inter/intra-annotator agreement in Python

Hadrien Titeux, Rachid Riad

► **To cite this version:**

Hadrien Titeux, Rachid Riad. pygamma-agreement: Gamma  $\gamma$  measure for inter/intra-annotator agreement in Python. 2021. hal-03144116v1

**HAL Id: hal-03144116**

**<https://hal.science/hal-03144116v1>**

Preprint submitted on 18 Feb 2021 (v1), last revised 12 Jun 2021 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License




# pygamma-agreement: Gamma $\gamma$ measure for inter/intra-annotator agreement in Python

Hadrien Titeux<sup>1</sup> and Rachid Riad<sup>1, 2</sup>

<sup>1</sup> LSCP/ENS/CNRS/EHESS/INRIA/PSL Research University, Paris, France  
<sup>2</sup> NPI/ENS/INSERM/UPEC/PSL Research University, Créteil, France

DOI: [DOIunavailable](#)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Pending Editor](#) 

## Reviewers:

- [@Pending Reviewers](#)

Submitted: N/A

Published: N/A

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Introduction

In the last decades, it became easier to collect large audio recordings in naturalistic conditions and large corpora of text from the Internet. This broadens the scope of questions that can be addressed regarding speech and language.

Scientists need to challenge their hypotheses and quantify the observed phenomena on speech and language; that is why researchers add different layers of annotations on top of speech and text data. Some type of human intervention is used to reliably describe events contained in the corpus's content (ex: Wikipedia articles, conversations, child babbling, animal vocalizations, or even just environmental sounds). These events can either be tagged at a particular point in time, or over a stretch of time. It is also commonplace to provide a categorical annotation or - in the case of speech - even precise transcriptions (Serratrice, 2000) for these events. Depending on the difficulty of the annotation task and the eventual expertise of the annotators, the annotations they produce can include a certain degree of interpretation. A common strategy when building annotated corpora is to have small parts of a corpus annotated by several annotators, to be able to quantify their consensus on that reduced subset of the corpus. If that consensus is deemed robust (i.e., agreement is high), we infer that the annotation task is well defined, less prone to interpretation, and that annotations that cover the rest of the corpus are reliable (Gwet, 2012). An objective measure of the agreement (and subsequent disagreement) between annotators is thus desirable.

## Statement of Need

The Gamma ( $\gamma$ ) Inter-Annotator Agreement Measure was proposed by (Mathet et al., 2015) as a way to quantify inter-rater agreement for sequences of annotations. The  $\gamma$ -agreement measure solves a number of the shortcomings of other pre-existing measures. This quantification will have to satisfy some constraints: segmentation, unitizing, categorization, weighted categorization and the support for any number of annotators. They should also provide a chance-corrected value. Measures, such as the  $\kappa$  (Carletta, 1996) or Krippendorff's  $\alpha$ 's (Krippendorff, 2011), have existed for some time to deal with these constraints, but never could address all of them at once. A detailed comparison between metrics is available in (Mathet et al., 2015). Furthermore, the authors of (Mathet et al., 2015) [provided a Java freeware](#) GUI implementation along with their paper.

Linguist and automated speech researchers today use analysis pipelines that are either Python or shell scripts. To this day, no open-source implementation allows for the  $\gamma$ -agreement to be computed in a programmatic way, and researchers that are already proficient in Python and willing to automate their work might be hindered by the graphical nature of the original Java implementation. Moreover, the original  $\gamma$ -agreement algorithm has several parameters that are determinant in its computation and cannot be configured as of now. For this reason, we thought it would greatly benefit the speech and linguistic scientific community if we could

provide them with a fully open-source Python implementation of the original algorithm. In that end, we made sure that our implementation had several key features. It is comparatively as fast as the original implementation, taking about 10s to compute a high-confidence  $\gamma$ -agreement measure on a middle-range processor. The `pygamma-agreement` package is modular and users can easily extend one of the modules without the burden of rebuilding an optimized code base from scratch (Ex: Users can easily add a new dissimilarity measure). Our code allows for fine-grained constructions of an annotation Continuum and an advanced configurability of the  $\gamma$ -agreement's different parameters. We also made sure it eases the interpretability of the input (i.e. the annotation continuum) and the output (i.e. the alignments) data structures, as both can be visualized in a Jupyter Notebook (see Figure 1). Finally, we support most of the commonly used data formats and analysis pipelines in the speech and linguistic fields.

## The `pygamma-agreement` Package

The `pygamma-agreement` package provides users with two ways to compute the  $\gamma$ -agreement for a corpus of annotations. The first one is to use the package's Python API.

```
import pygamma_agreement as pa
continuum = pa.Continuum.from_csv("data/PaulAlexSuzann.csv")
dissimilarity = pa.Dissimilarity(categories=list(continuum.categories))
gamma_results = continuum.compute_gamma(dissimilarity, confidence_level=0.02)
print(f"Gamma is {gamma_results.gamma}")
```

The most important primitives from our API (the Continuum Figure 1 and Alignment Figure 2 classes) can be displayed using the `matplotlib.pyplot` backend if the user is working in a Jupyter notebook.

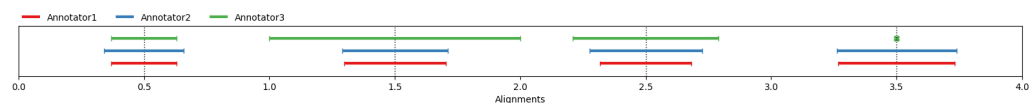


Figure 1: Displaying a Continuum in a jupyter notebook.

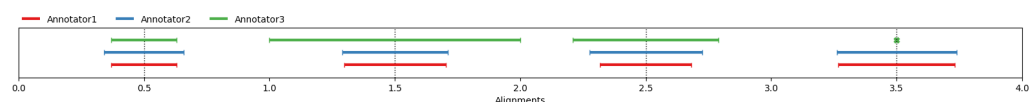


Figure 2: Displaying an Alignment in a jupyter notebook.

The second one is a command-line application that can be invoked directly from the shell, for those who prefer to use shell scripts for corpus processing:

```
pygamma-agreement corpus/*.csv --confidence_level 0.02 --output_csv results.csv
```

We support a variety of commonly used annotation formats among speech researchers and linguists : RTTM, ELAN, TextGrid, CSV and `pyannotate.core.Annotation` objects.

Computing the  $\gamma$ -agreement requires both array manipulation and the solving of multiple optimization problem formulated as Mixed-Integer Programming (MIP) problems. We thus used the *de facto* standard for all of our basic array operations, NumPy (Harris et al., 2020). Since some parts of the algorithm are fairly demanding, we made sure that these parts were heavily optimized using `numba` (Lam et al., 2015). We used `cvxpy`'s (Diamond & Boyd, 2016) MIP-solving framework to solve the optimization problem. For time-based annotations, we rely on primitives from `pyannotate.core` (Bredin et al., 2020). We made sure that it is robustly tested using the widely-adopted `pytest` testing framework. We also made sure that

pygamma-agreement's outputs matched both the theoretical values from the original paper and those of the Java freeware. We set-up an automated Travis CI to use these tests to ensure our package's quality. Most of our package's code is type-hinted and has descriptive docstrings, both of which can be leveraged by IDEs to ease the use of our API.

We provide a user [documentation](#) as well as an example Jupyter notebook in our package's repository. Additionally, we have used and tested pygamma-agreement in conjunction with the development of our own custom-built annotation platform, Seshat (Titeux et al., 2020). In [Table](#) , we present two use cases for our implementation of the  $\gamma$ -agreement measure on two corpora. These two corpora, ranging from medical interviews to child recording, allowed us to evaluate the performance of the  $\gamma$ -agreement on a wide panel of annotation types.

Corpus	Annotation	# Classes	Mean of $\gamma$
Clinical Interviews	Turn-Takings	3	0.64
Clinical Interviews	Utterances	1	0.61
Child Recordings	Speech Activity	1	0.46
Child Recordings	Child/Adult-directed speech	2	0.27

Table :  $\gamma$  Inter-rater agreement for clinical interviews (16 samples) and child-centered day-long recordings (20 samples).

We have uploaded our package to the [Pypi repository](#), thus, pygamma-agreement can be installed using pip.

## Future Work

Our implementation of the  $\gamma$ -agreement opens the path for a number of potential extensions:

- An obvious improvement is to add support for the “ $\gamma$ -cat” metric, a complement measure (Mathet, 2017) for the  $\gamma$ -agreement.
- The  $\gamma$ -agreement's theoretical framework allows for some useful improvements such as:
  - the implementation of a sequence-based dissimilarity, based on the Levenshtein distance. This would however require a numba re-implementation of the latter. This could, for instance, be used to compare annotated strings of phonemes.
  - for categorical annotations, the support for an undefined set of categories, with annotators using different sets of categories. This would be solved using an adapted implementation of the [Hungarian Algorithm](#). This could be useful in unconstrained diarization annotation tasks.

## Acknowledgements

We are thankful to Yann Mathet for his help on understanding his work on the  $\gamma$ -agreement. We also thank Xuan-Nga Cao, Anne-Catherine Bachoux-Lévy and Emmanuel Dupoux for their advice, as well as Julien Karadayi for helpful discussions and feedbacks. This work is funded in part by the Agence Nationale pour la Recherche (ANR-17-EURE-0017Frontcog, ANR-10-IDEX-0001-02 PSL\*, ANR-19-P3IA-0001PRAIRIE 3IA Institute) and Grants from Neuratris, from Facebook AI Research (Research Gift), Google (Faculty Research Award), Microsoft Research (Azure Credits and Grant), and Amazon Web Service (AWS Research Credits).

## References

Bredin, H., Yin, R., Coria, J. M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., & Gill, M. (2020). Pyannote.audio: Neural building blocks for speaker

- diarization. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7124–7128. <https://doi.org/10.1109/ICASSP40776.2020.9052974>
- Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2), 249–254. <https://www.aclweb.org/anthology/J96-2004>
- Diamond, S., & Boyd, S. (2016). CVXPY: A python-embedded modeling language for convex optimization. *J. Mach. Learn. Res.*, 17(1), 2909–2913.
- Gwet, K. Li. (2012). *Handbook of inter-rater reliability*. 6–7.
- Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., R'io, J. F. del, Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Krippendorff, K. (2011). *Computing krippendorff's alpha-reliability*. [https://repository.upenn.edu/asc\\_papers/43/](https://repository.upenn.edu/asc_papers/43/)
- Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. <https://doi.org/10.1145/2833157.2833162>
- Mathet, Y. (2017). The Agreement Measure Gamma-Cat : a Complement to Gamma Focused on Categorization of a Continuum. *Computational Linguistics*, 43(3), 661–681. [https://doi.org/10.1162/COLI\\_a\\_00296](https://doi.org/10.1162/COLI_a_00296)
- Mathet, Y., Widlöcher, A., & Métivier, J.-P. (2015). The unified and holistic method gamma ( $\gamma$ ) for inter-annotator agreement measure and alignment. *Computational Linguistics*, 41(3), 437–479. [https://doi.org/10.1162/COLI\\_a\\_00227](https://doi.org/10.1162/COLI_a_00227)
- Serratrice, L. (2000). Book reviews : The CHILDES project: Tools for analyzing talk, 3rd edition. *First Language*, 20(60), 331–337. <https://doi.org/10.1177/014272370002006006>
- Titeux, H., Riad, R., Cao, X.-N., Hamilakis, N., Madden, K., Cristia, A., Bachoud-Lévi, A.-C., & Dupoux, E. (2020, May). Seshat: A tool for managing and verifying annotation campaigns of audio data. *LREC 2020 - 12th Language Resources and Evaluation Conference*. <https://hal.archives-ouvertes.fr/hal-02496041>