# pybreathe: a python package for respiratory airflow rates analysis

Thibaut Coustillet [1,2]

1 Sorbonne Université, CNRS, Inserm, Development, Adaptation and Ageing, Dev2A, F-75005 Paris, France ROR **2** Sorbonne Université, CNRS, Inserm, Institut de Biologie Paris-Seine, IBPS, F-75005 Paris, France ROR

Summary

Breathing is the vital function that enables air to flow into the lungs during inhalation and out during exhalation. Inhalation delivers (di)oxygen to the tissues, while exhalation flushes out carbon dioxide. In physiology, breathing is widely studied to investigate a whole range of respiratory diseases as well as physical abilities. A consistent and robust analytical framework is essential to cope with the large volume of data and to address the often time-consuming nature of routine analysis procedures. pybreathe is a python package that allows breathing to be formally analysed. It lets users to extract features such as the volume inhaled and exhaled, the inspiratory and expiratory times, and the breathing frequency. The package has been designed to be user-friendly, relying solely on a user-supplied discretised air flow rate considered as a time series (instantaneous flow rate).

Statement of need

In 2017, half a billion people worldwide lived with a chronic respiratory disease (Soriano et al., 2020). Concurrently, the role of breathing has garnered growing attention in research focused on both athletic/sport performance (Contreras-Briceño et al., 2024; Harbour et al., 2022) and overall well-being (Fincham et al., 2023). Recently, new algorithms have been developed to structure analysis and open up new insights. Such tools complement commercial analysis software used historically (Lusk et al., 2023).

On the one hand, most of the open source software relies heavily on peak/hollow detection (Bishop et al., 2022; Brammer, 2020; Makowski et al., 2021) to extract features such as amplitude and breathing period. However, such local extrema detection approaches often require human correction or complementary algorithms to guarantee the accuracy of the results (Vanegas et al., 2020). Besides, they are more suited for the characterization of instantaneous volume than of instantaneous flow. Although the former can be deduced from the latter (Figure 1), it is generally flow rates, themselves derived from pressure differences that are supplied (Criée et al., 2011). In some cases, however, signals and related algorithms may also originate from chest/abdominal belts (Holm et al., 2024).
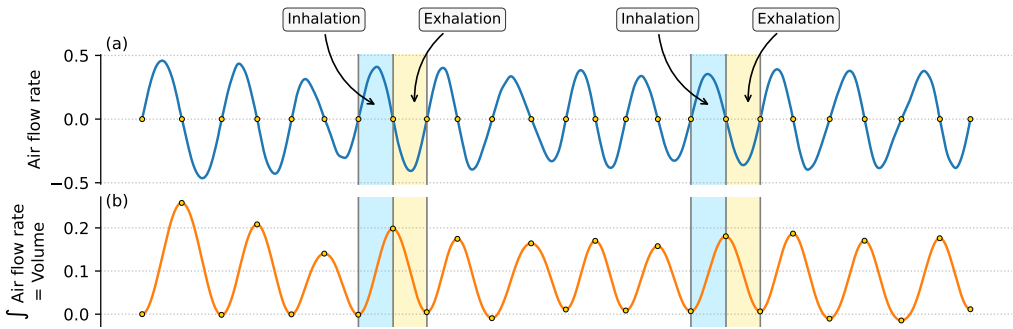
Figure 1: Relationship between instantaneous flow rate (a) and instantaneous volume (b). The volume is obtained by integrating the flow rate over time. Thus, when the flow rate is positive (inhalation; blue areas), the volume increases, whereas when the flow rate is negative (exhalation; yellow areas), the volume decreases.

On the other hand, some advanced algorithms use cutting-edge clustering methods to detect respiratory patterns that go beyond the features mentioned above (Germain et al., 2023). While offering deeper physiological insights, they require advanced programming skills and knowledge and may be complicated to set up in practice for non-computer users.

Here, we sought to implement an easy-to-use framework specially designed to facilitate respiratory analyses derived from instantaneous air flow rates (recorded by plethysmography). Users can acquire their respiratory data using any standard software. A necessary and sufficient condition for using pybreathe is to export the data and discretise it into a classic text format (*e.g.*, .txt, .csv) as shown in Table 1.

Table 1: Example of a two-column table depicting the instantaneous discretised air flow rate (time series) required for the use of pybreathe.

time	values
0.0	0.0650
0.004	0.0660
0.008	0.0671
0.012	0.0681
0.016	0.0692
...	...

In contrast to other respiratory algorithms, pybreathe operates on ventilatory flow rather than volume. The fundamental feature of a respiratory signal being the tidal volume (*i.e.*, the volume passing through the lungs during a single breath), peak/hollow analysis cannot be applied in the case of air flow rates because the amplitude (*i.e.*, height) depends on the 'speed' at which the air flows in and out: for the same exhaled or inhaled volume, the faster the airflow, the greater the amplitude (Figure 2).
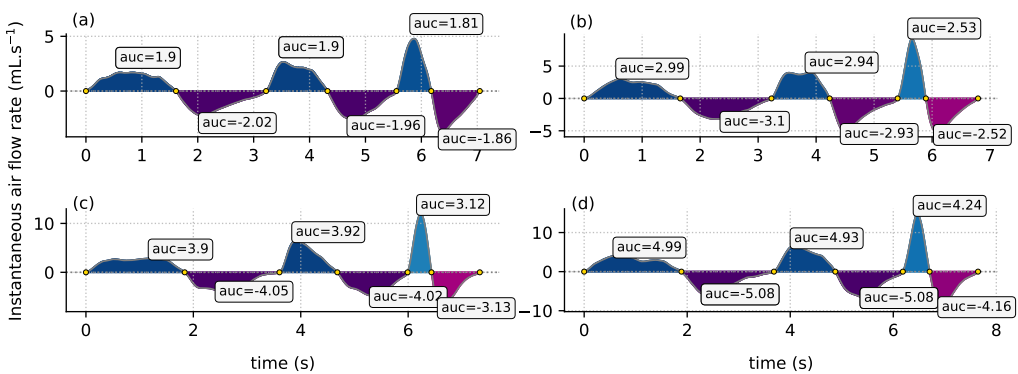
Figure 2: Manual injection (blue) and aspiration (purple) of different quantities of air into a chamber with a syringe at three different speeds. (a) 2 mL; (b) 3 mL; (c) 4 mL; (d) 5 mL.

In this situation, to really grasp the tidal volume, we need to get the Area Under the Curve (AUC) instead of the amplitude (Table 2).

Table 2: Comparison of the integral (Area Under the Curve) and amplitude (height) of several volumes of air manually injected/aspirated into a chamber.

actual volume	speed	positive integral	negative integral	positive amplitude	negative amplitude
≈ 2 mL	slow	1.90	- 2.02	1.70	- 2.26
≈ 2 mL	moderate	1.90	- 1.96	2.66	- 2.54
≈ 2 mL	fast	1.81	- 1.86	4.79	- 3.75
≈ 3 mL	slow	2.99	- 3.10	3.04	- 3.19
≈ 3 mL	moderate	2.94	- 2.93	3.98	- 5.07
≈ 3 mL	fast	2.53	- 2.52	9.24	- 5.2
≈ 4 mL	slow	3.90	- 4.05	2.91	- 3.71
≈ 4 mL	moderate	3.92	- 4.02	6.46	- 4.81
≈ 4 mL	fast	3.12	- 3.13	12.03	- 7.09
≈ 5 mL	slow	4.99	- 5.08	4.21	- 5.76
≈ 5 mL	moderate	4.93	- 5.08	6.58	- 6.70
≈ 5 mL	fast	4.24	- 4.16	14.88	- 9.04

pybreathe fundamentals

For a given respiratory signal, pybreathe detects zero-crossings (Figure 1a). AUC (integral) of these zero-separated segments therefore corresponds to the volume inhaled or exhaled (depending on the configuration of the primary acquisition software). The duration of these segments (time between two zeros) corresponds to the inspiratory or expiratory time.

The pybreathe package is accompagnied by a *BreathingFlow* object which is the core of the algorithm. Users should instantiate a *BreathingFlow* object with their own data (*e.g.*, Table 1).

```python
from pybreathe import BreathingFlow

my_signal = BreathingFlow.from_file(
    filename="path_to_your_discretised_flow.txt",
```

```
        identifier="my data"
)
```

58 The package comes with an example script based on simulated breathing signals. It explains
59 the milestones involved in carrying out a complete analysis and it supply useful documentation.
60 To get started with pybreathe API, users are strongly advised to refer to this script.

Proof

62 To ensure that the package worked correctly, we checked the volumes extracted by pybreathe
63 when we injected/aspirated known quantities of air (Figure 2 and Table 2). The AUC
64 values matched well with the corresponding injected volumes. However, because the manual
65 experiment can be imprecise due to the experimenter and the precision of the syringe (especially
66 at high speed), we also created an artifical respiratory signal corresponding to the sine function
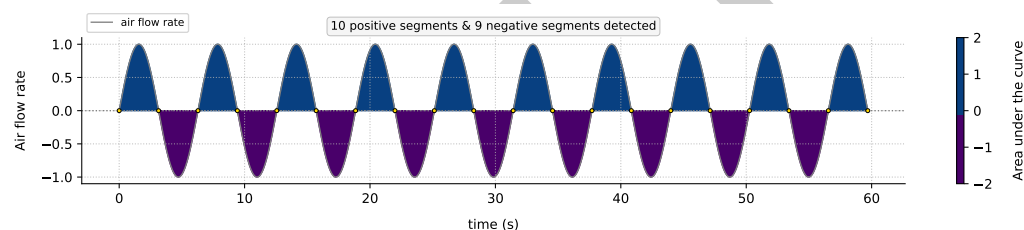67 (Figure 3).

Figure 3: Graph of the sine function on the interval $[0, 10\pi]$.

68 Based on the sine, we checked that the signal features obtained with pybreathe were indeed
69 the same as those obtained *mathematically*. Users can access these demonstrations in the
70 validation notebook.

71 For example, pybreathe exactly identified that the duration of each of the segments in which
72 the sine function is positive is exactly π, and that the AUC of these same segments is exactly
73 2.

Acknowledgements

References

80 Bishop, M., Weinhold, M., Turk, A. Z., Adeck, A., & SheikhBahaei, S. (2022). An open-source
81   tool for automated analysis of breathing behaviors in common marmosets and rodents.
82   *eLife*, *11*, e71647. https://doi.org/10.7554/eLife.71647

83 Brammer, J. (2020). Biopeaks: A graphical user interface for feature extraction from heart-
84   and breathing biosignals. *Journal of Open Source Software*, *5*(54), 2621. https://doi.org/
85   10.21105/joss.02621

86 Contreras-Briceño, F., Cancino, J., Espinosa-Ramírez, M., Fernández, G., Johnson, V.,
87   & Hurtado, D. E. (2024). Estimation of ventilatory thresholds during exercise using

88 respiratory wearable sensors. *Npj Digital Medicine*, *7*(1). https://doi.org/10.1038/
89 s41746-024-01191-9

90 Criée, C. P., Sorichter, S., Smith, H. J., Kardos, P., Merget, R., Heise, D., Berdel, D., Köhler,
91 D., Magnussen, H., Marek, W., Mitfessel, H., Rasche, K., Rolke, M., Worth, H., & Jörres,
92 R. A. (2011). Body plethysmography – its principles and clinical use. *Respiratory Medicine*,
93 *105*(7), 959–971. https://doi.org/10.1016/j.rmed.2011.02.006

94 Fincham, G. W., Strauss, C., Montero-Marin, J., & Cavanagh, K. (2023). Effect of breathwork
95 on stress and mental health: A meta-analysis of randomised-controlled trials. *Scientific
96 Reports*, *13*(1). https://doi.org/10.1038/s41598-022-27247-y

97 Germain, T., Truong, C., Oudre, L., & Krejci, E. (2023). Unsupervised classification of
98 plethysmography signals with advanced visual representations. *Frontiers in Physiology*, *14*,
99 1154328. https://doi.org/10.3389/fphys.2023.1154328

100 Harbour, E., Stöggl, T., Schwameder, H., & Finkenzeller, T. (2022). Breath tools: A synthesis
101 of evidence-based breathing strategies to enhance human running. *Frontiers in Physiology*,
102 *13*. https://doi.org/10.3389/fphys.2022.813243

103 Holm, B., Borsky, M., Arnardottir, E. S., Serwatko, M., Mallett, J., Islind, A. S., & Óskarsdóttir,
104 M. (2024). BreathFinder: A method for non-invasive isolation of respiratory cycles utilizing
105 the thoracic respiratory inductance plethysmography signal. *Nature and Science of Sleep*,
106 *16*, 1253–1266. https://doi.org/10.2147/NSS.S468431

107 Lusk, S., Ward, C. S., Chang, A., Twitchell-Heyne, A., Fattig, S., Allen, G., Jankowsky, J. L.,
108 & Ray, R. S. (2023). An automated respiratory data pipeline for waveform characteristic
109 analysis. *The Journal of Physiology*, *601*(21), 4767–4806. https://doi.org/10.1113/
110 JP284363

111 Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., Schölzel,
112 C., & Chen, S. H. A. (2021). NeuroKit2: A python toolbox for neurophysiological signal
113 processing. *Behavior Research Methods*, *53*(4), 1689–1696. https://doi.org/10.3758/
114 s13428-020-01516-y

115 Soriano, J. B., Kendrick, P. J., Paulson, K. R., Gupta, V., Abrams, E. M., Adedoyin, R.
116 A., Adhikari, T. B., Advani, S. M., Agrawal, A., Ahmadian, E., Alahdab, F., Aljunid,
117 S. M., Altirkawi, K. A., Alvis-Guzman, N., Anber, N. H., Andrei, C. L., Anjomshoa,
118 M., Ansari, F., Antó, J. M., … Vos, T. (2020). Prevalence and attributable health
119 burden of chronic respiratory diseases, 1990–2017: A systematic analysis for the global
120 burden of disease study 2017. *The Lancet Respiratory Medicine*, *8*(6), 585–596. https:
121 //doi.org/10.1016/S2213-2600(20)30105-3

122 Vanegas, E., Igual, R., & Plaza, I. (2020). Sensing systems for respiration monitoring: A
123 technical systematic review. *Sensors (Basel, Switzerland)*, *20*(18), 5446. https://doi.org/
124 10.3390/s20185446