The Journal of Open Source Software

EMGFlow: A Python package for pre-processing and feature extraction of electromyographic signals

William L. Conley [1¶] and **Steven R. Livingstone** [1]

1 Department of Computer Science, Ontario Tech University, Oshawa, Canada ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

The use of surface electromyography (sEMG) as a measure of human physiology and behaviour has grown recently, supported by developments in deep learning and wearable computing. Here, we present *EMGFlow*, an open-source Python package for preprocessing and extracting features from sEMG signals. *EMGFlow* has been designed to facilitate the analysis of large datasets through batch processing of signal files, a common requirement in machine learning. The package extracts an extensive set of features from both time and frequency domains. Regular expression matching provides additional flexibility in mapping files for selective preprocessing and extraction. The use of Pandas DataFrame throughout allows users to mix and match elements of the processing pipeline, supporting interoperability with other packages. An interactive dashboard supports human decision processes through a visual comparison of signals at each stage of preprocessing. *EMGFlow* is released under the GNU General Public License (v3.0) and can be installed from PyPI. Source code, documentation, and examples are accessible on GitHub (https://github.com/Willson/EMGFlow-Python-Package).

Statement of Need

Although several packages exist for processing physiological and neurological signals, support for sEMG has remained limited. Many packages lack a comprehensive set of features that can be extracted from sEMG data, leaving researchers to use a patchwork of tools. Other packages are orientated around event detection in individual recordings and use a GUI-based workflow that requires more manual intervention. While this design works well for processing unedited continuous recordings of a single participant, it complicates the extraction of features from large datasets common to machine learning (Abadi et al., 2015; Chen et al., 2022; Koelstra et al., 2012; Schmidt et al., 2018; Sharma et al., 2019; Zhang et al., 2016).

EMGFlow, a portmanteau of EMG and Workflow, fills this gap by providing a flexible pipeline for extracting a wide range of sEMG features, with a scalable design suited for large datasets.

Comparison to Other Packages

Compared to other toolkits, *EMGFlow* extracts a comprehensive set of 32 statistical features from sEMG signals (Bota et al., 2024; Makowski et al., 2021; Sjak-Shie, 2022; Soleymani et al., 2017). An interactive dashboard visualizes batch processed files rather than individual recordings, allowing the operator to efficiently view the effects of preprocessing stages across all files. Adjustable filter settings and smoothing functions support cleaning of data collected in North America or internationally (50 vs 60 HZ mains AC), a subtle difference overlooked in some packages.

Features

Processing Pipeline

Extracting features from large datasets is a common task in machine learning and quantitative domains. *EMGFlow* supports this need through batch-processing, allowing users to either semi- or fully automate the treatment of sEMG recordings. To demonstrate, we use data from PeakAffectDS (Greene et al., 2022), a collection of physiological signals that includes two channels of facial sEMG, labelled Zyg and Cor, capturing Zygomaticus major and Corrugator supercilii muscle activity respectively. We begin by defining the path to the directory containing our raw, uncleaned files stored in plaintext (.csv) format. We then apply a notch filter to remove the AC mains noise introduced by the recording system's power source, a common initial step in preprocessing raw sEMG signals.

```python
import EMGFlow

# Paths for sEMG files
raw_path = 'Data/01_Raw'
notch_path = 'Data/02_Notch'

# Sampling rate
sr = 2000

# Columns containing sEMG
cols = ['EMG_zyg', 'EMG_cor']

# Notch filter parameters
notch_vals = [(50,5)]

# Apply notch filter to raw sEMG files
EMGFlow.NotchFilterSignals(raw_path, notch_path, sr, notch_vals, cols)
```

Additional arguments allow users to customize which files are selected and how they are processed. Filtering functions accept an optional regex argument, allowing users to apply filters to specific files. Most functions use common sense defaults, which can be modified task-wide or for select cases. For example, in North America, mains electricity is nominally supplied at 120 VAC 60 Hz, while other countries may supply power at 200-240 VAC 50Hz. This variation in frequency requires different notch filter settings depending on where the data were recorded. *EMGFlow* accommodates this need by allowing the user to specify the frequency and quality factor of the applied filter. Extending our first example, we now apply an additional notch filter to a subset of files exhibiting noise at 150 Hz, the 3rd harmonic of the mains source.

```python
# Filter parameters for files that start with "08" or "11"
notch_vals_extra = [(150,25)]
reg_pat = '^(08|11)'

# Apply notch filter to file subset
EMGFlow.NotchFilterSignals(notch_path, notch_path, sr, notch_vals_extra, cols,
                           expression=reg_pat, exp_copy=True)
```

Visualization of Preprocessing Stages

The application of a bandpass filter is often the second stage in preprocessing sEMG signals, as it isolates the frequency spectrum of human muscle activity. Signals are commonly filtered to the 10-500 Hz range (Livingstone et al., 2016; McManus et al., 2020; Sato et al., 2021; Tamietto et al., 2009), though precise filter corner frequencies vary by research domain

The Journal of Open Source Software

and approach (Abadi et al., 2015). After filtering, data can be further smoothed to remove high-frequency noise and outliers in preparation for the extraction of temporal features. The default smoother is RMS, equal to the square root of the total power in the sEMG signal and commonly used to estimate signal amplitude (McManus et al., 2020). Additional filter options are provided, including boxcar, Gaussian, and LOESS.

EMGFlow provides an interactive Shiny dashboard to visualize the effects of preprocessing on sEMG signals. Preprocessing stages can be displayed simultaneously or shown individually with options for Notch, Bandpass, and Smoothing steps. Users can select the file for visualization using the Files dropdown box. The dashboard is generated from a list of file paths containing files at different stages of preprocessing. Here, our example shows how signals are further bandpass filtered and smoothed, with results visualized using the dashboard.

```
# Paths for sEMG files
band_path = 'Data/03_Bandpass'
smooth_path = 'Data/04_Smoothed'

# Filter and smoothing parameters
band_low = 20
band_high = 450
win_length = 50

# Apply bandpass and smoothing filters
EMGFlow.BandpassFilterSignals(notch_path, band_path, sr, band_low, band_high,
                              cols)
EMGFlow.SmoothFilterSignals(band_path, smooth_path, sr, win_length, cols)

# Paths for dashboard generation
in_paths = [smooth_path, band_path, notch_path]
labels = ['Smooth', 'Bandpass', 'Notch']

# Column to visualize, and units of measurement
show_col = 'EMG_zyg'
units = 'mV'

# Generate dashboard
EMGFlow.GenPlotDash(in_paths, sampling_rate, show_col, units, labels)
```

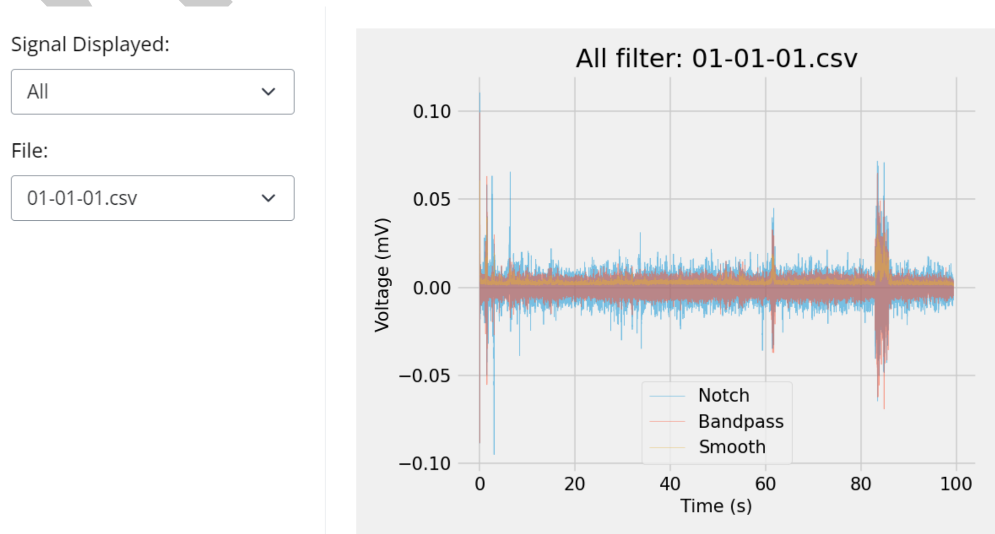

Figure 1: *EMGFlow*'s interactive dashboard visualizing effects of different preprocessing stages

on batch processed files.

The nature of electromyographic recordings

To better understand the range of features extracted by *EMGFlow*, we begin with a review of surface electromyography as a recording instrument. Nearly all body movement occurs by muscle contraction. During contraction, nerve impulses sent from motoneurons cause muscle fibers innervated by the axon to discharge, creating a motor unit action potential (McManus et al., 2020). The speed at which action potentials propogate down the fibre is called muscle fiber conduction velocity. Each motor unit firing results in a force twitch. The superposition of these twiches over time produces a sustained force that enables functional muscle activity, such as lifting or smiling (De Luca, 2008).

Surface electromyography measures voltage difference across muscle fibers generated by action potentials, producing a voltage timeseries that quantifies muscle activity (Fridlund & Cacioppo, 1986). It is from this voltage timeseries that statistical features are extracted.

Feature Extraction Routines

Following data preprocessing, the signal files are ready for feature extraction. *EMGFlow* extracts 32 features that capture information in both time and frequency domains. The set of 18 time-domain features capture standard statistical moments, including mean, variance, skew, and kurtosis, along with sEMG-specific measures. These include features such as Willison amplitude, an indicator of motor unit firing calculated as the number of times the sEMG amplitude exceeds a threshold, and log-detector, an estimate of the exerted muscle force (Tkach et al., 2010).

A set of 12 frequency-domain features are also extracted, providing information on the shape and distribution of the signal's power spectrum. Measures such as median frequency (Phinyomark et al., 2009) provide insight into changes in muscle fibre conduction velocity and are used in the assessment of muscle fatigue (Lindstrom et al., 1977; McManus et al., 2020; Van Boxtel et al., 1983). Standard frequency measures include spectral centroid, flatness, entropy, and roll-off. One novel sEMG feature introduced here is Twitch Ratio, an adaptation of Alpha Ratio from speech signal analysis (Eyben et al., 2016). Twitch Ratio is defined as the ratio of energy contained in the upper versus lower power spectrum, with a threshold of 60 Hz to delineate slow- and fast-twitch muscles fibres (Hegedus et al., 2020).

Here, we demonstrate feature extraction in *EMGFlow*. After specifying locations of preprocessed files, features are summarized into a single CSV file, containing rows for each file analyzed, as shown below.

```
# Path where feature table will be written to disk
feature_path = 'Data/05_Feature'

# Extracts features
df = EMGFlow.ExtractFeatures(band_path, smooth_path, feature_path, sr, cols)

# Print first few rows of extracted features table. The “File_ID” column
# contains the names of the files extracted, and the additional columns take
# the format “[Column name]_[Feature name]”.
df.head()
"""
File_ID column contains

        File_ID  EMG_zyg_Min  ...  EMG_cor_Spec_Rolloff  EMG_cor_Spec_Bandwidth
0  01-01-01.csv     0.000826  ...              0.040222             1424.933862
1  01-01-02.csv     0.000740  ...              0.019559             2651.987804
```

```
2  01-01-03.csv     0.000780  ...             0.065183            2021.345274
3  01-01-04.csv     0.000660  ...             0.087384            1755.834836
4  01-01-05.csv     0.000697  ...             0.057368            1174.562467

[5 rows x 61 columns]
"""
```

Community Guidelines

We welcome contributions to the project. These can be initiated through the project's issue tracker or via a pull request. Suggestions for feature enhancements, tips, as well as general questions and concerns, can also be expressed through direct interaction with contributors and developers.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

During the preparation of this work, the authors used GPT-4o to edit a final draft of the manuscript for flow, tone, and grammatical correctness. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), (#2023-03786), and from the Faculty of Science, Ontario Tech University.

Author contributions

S.R.L. conceptualised the project. W.L.C. and S.R.L. designed the toolbox functionality. W.L.C. wrote the toolbox code. W.L.C. created and maintained the Github repository. W.L.C. prepared figures for manuscript and Github repository. W.L.C. and S.R.L prepared the manuscript and approved the final version of the manuscript for submission.

References

Abadi, M. K., Subramanian, R., Kia, S. M., Avesani, P., Patras, I., & Sebe, N. (2015). DECAF: MEG-Based Multimodal Database for Decoding Affective Physiological Responses. *IEEE Transactions on Affective Computing*, *6*(3), 209–222. https://doi.org/10.1109/TAFFC.2015.2392932

Bota, P., Silva, R., Carreiras, C., Fred, A., & Silva, H. P. da. (2024). BioSPPy: A Python toolbox for physiological signal processing. *SoftwareX*, *26*, 101712. https://doi.org/10.1016/j.softx.2024.101712

Chen, J., Ro, T., & Zhu, Z. (2022). Emotion Recognition With Audio, Video, EEG, and EMG: A Dataset and Baseline Approaches. *IEEE Access*, *10*, 13229–13242. https://doi.org/10.1109/ACCESS.2022.3146729

De Luca, C. J. (2008). A practicum on the use of sEMG signals in movement sciences. *Delsys Inc.*

Eyben, F., Scherer, K. R., Schuller, B. W., Sundberg, J., André, E., Busso, C., Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S., & Truong, K. P. (2016). The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. *IEEE Transactions on Affective Computing*, *7*(2), 190–202. https://doi.org/10.1109/TAFFC.2015.2457417

Fridlund, A. J., & Cacioppo, J. T. (1986). Guidelines for Human Electromyographic Research. *Psychophysiology*, *23*(5), 567–589. https://doi.org/10.1111/j.1469-8986.1986.tb00676.x

Greene, N., Livingstone, S. R., & Szymanski, L. (2022). *PeakAffectDS*. Zenodo. https://doi.org/10.5281/zenodo.6403363

Hegedus, A., Trzaskoma, L., Soldos, P., Tuza, K., Katona, P., Greger, Z., Zsarnoczky-Dulhazi, F., & Kopper, B. (2020). Adaptation of Fatigue Affected Changes in Muscle EMG Frequency Characteristics for the Determination of Training Load in Physical Therapy for Cancer Patients. *Pathology & Oncology Research*, *26*(2), 1129–1135. https://doi.org/10.1007/s12253-019-00668-3

Koelstra, S., Muhl, C., Soleymani, M., Lee, J.-S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., & Patras, I. (2012). DEAP: A Database for Emotion Analysis ;Using Physiological Signals. *IEEE Transactions on Affective Computing*, *3*(1), 18–31. https://doi.org/10.1109/T-AFFC.2011.15

Lindstrom, L., Kadefors, R., & Petersen, I. (1977). An electromyographic index for localized muscle fatigue. *Journal of Applied Physiology*, *43*(4), 750–754. https://doi.org/10.1152/jappl.1977.43.4.750

Livingstone, S. R., Vezer, E., McGarry, L. M., Lang, A. E., & Russo, F. A. (2016). Deficits in the Mimicry of Facial Expressions in Parkinson's Disease. *Frontiers in Psychology*, *7*. https://doi.org/10.3389/fpsyg.2016.00780

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

McManus, L., De Vito, G., & Lowery, M. M. (2020). Analysis and Biophysics of Surface EMG for Physiotherapists and Kinesiologists: Toward a Common Language With Rehabilitation Engineers. *Frontiers in Neurology*, *11*. https://doi.org/10.3389/fneur.2020.576729

Phinyomark, A., Limsakul, C., & Phukpattaranont, P. (2009). A novel feature extraction for robust EMG pattern recognition. *arXiv Preprint arXiv:0912.3973*. https://doi.org/10.48550/arXiv.0912.3973

Sato, W., Murata, K., Uraoka, Y., Shibata, K., Yoshikawa, S., & Furuta, M. (2021). Emotional valence sensing using a wearable facial EMG device. *Scientific Reports*, *11*(1), 5757. https://doi.org/10.1038/s41598-021-85163-z

Schmidt, P., Reiss, A., Duerichen, R., Marberger, C., & Van Laerhoven, K. (2018). Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection. *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, 400–408. https://doi.org/10.1145/3242969.3242985

Sharma, K., Castellini, C., Broek, E. L. van den, Albu-Schaeffer, A., & Schwenker, F. (2019). A dataset of continuous affect annotations and physiological signals for emotion analysis. *Scientific Data*, *6*(1), 196. https://doi.org/10.1038/s41597-019-0209-0

Sjak-Shie. (2022). *PhysioData Toolbox* (Version 0.6.3). https://physiodatatoolbox.leidenuniv.nl/

Soleymani, M., Villaro-Dixon, F., Pun, T., & Chanel, G. (2017). Toolbox for Emotional feature extraction from Physiological signals (TEAP). *Frontiers in ICT*, *4*. https://doi.org/10.

3389/fict.2017.00001

Tamietto, M., Castelli, L., Vighetti, S., Perozzo, P., Geminiani, G., Weiskrantz, L., & Gelder, B. de. (2009). Unseen facial and bodily expressions trigger fast emotional reactions. *Proceedings of the National Academy of Sciences*, *106*(42), 17661–17666. https://doi.org/10.1073/pnas.0908994106

Tkach, D., Huang, H., & Kuiken, T. A. (2010). Study of stability of time-domain features for electromyographic pattern recognition. *Journal of NeuroEngineering and Rehabilitation*, *7*(1), 21. https://doi.org/10.1186/1743-0003-7-21

Van Boxtel, A., Goudswaard, P., Van der Molen, G., & Van Den Bosch, W. (1983). Changes in electromyogram power spectra of facial and jaw-elevator muscles during fatigue. *Journal of Applied Physiology*, *54*(1), 51–58. https://doi.org/10.1152/jappl.1983.54.1.51

Zhang, L., Walter, S., Ma, X., Werner, P., Al-Hamadi, A., Traue, H. C., & Gruss, S. (2016). "BioVid Emo DB": A multimodal database for emotion analyses validated by subjective ratings. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1–6. https://doi.org/10.1109/SSCI.2016.7849931