# Decomposition of Muscle Activity for Sensorimotor Neuroscience Final Report

Daniel King, Jasmine Ortega, Rada Rudyak, Rowan Sivanandam

Mentor: Dr. Alexi Rodríguez-Arelis
Partner: Dr. Jean-Sébastien Blouin, Sensorimotor Physiology Laboratory, UBC

2022-06-29

## Contents



### Executive Summary

The decomposition of EMG signals into individual motor units is a critical technique for understanding the relationship between the nervous and muscular systems. The UBC Sensorimotor Physiology Laboratory's research currently decomposes raw EMG signals using a free software from OT Bioelettronica, whose code is based on a paper from Negro et al. (2016). As this software has several shortcomings, we have replicated the decomposition algorithm and packaged it in an open-source Python package called `EMGdecomPy`, over the course of the capstone project. Additionally, we have included a reproducible tool to visualize the decomposed output of the algorithm.

The decomposition algorithm described by Negro et al. (2016) was broken up into individual functions. These functions were individually tested and extensively documented for the sake of transparency. Finally, per the partner's request, `EMGdecomPy` was made flexible enough to run experiments of an arbitrary. Preliminary qualitative results show that `EMGdecomPy` identifies 3 out of 5 of the same motor units as Hug et al. (2021) when run on their raw EMG data obtained from the *Gastrocnemius lateralis* muscle with 10% contraction intensity.

While the sequential steps to decompose raw EMG signals were given by Negro et al. (2016), there was room for interpretation and improvement in the implementation of each step. Future recommendations for improvement include, decreasing the algorithm's run-time through optimization, increasing decomposition accuracy through domain knowledge, and implementing a relearning feature into the algorithm.

### Introduction

*The sensorimotor system*, or the bodily process that links physical movement to outside sensory information, is a key component of human motion (Forbes, Little, and Candow 2012). A healthy sensorimotor system is critical, as it is required to communicate, interact, and navigate through one's environment. Muscle

movement, a key component of motion and the motor system's purpose, is the carefully coordinated result between the nervous and muscular systems.

The Sensorimotor Physiology Lab at the UBC School of Kinesiology studies the nervous system's role in muscle movement. Dr. Jean-Sébastien Blouin's research is targeted at understanding the sensorimotor system's role in human balance, as well as expanding on the known neurological-muscular mechanisms involved in human motion. By studying the link between the nervous system and movement, researchers can further understand the effects of neuromuscular damage. Findings can be used to establish preventative measures, as well as to create more effective therapeutic treatments for those with chronic neuromuscular disorders and age-related neural degeneration (Purves 2018).

The brain, spinal cord, and attached nerves form the nervous system, which is responsible for initiating and propagating electrical movement signals to the correct muscles. These electrical bursts, known as action potential, are initially fired off from the brain. Action potential propagates throughout the body via neurons, which are cells specialized in transmitting electrical signals across long distances.

Motor neurons are a specific subclass of neurons that create a junction between the central nervous system and the muscular system. Motor neurons attach to fibers in the muscle, forming an entity known as a motor unit, as seen in **figure 1**. When struck by action potential, the motor unit generates motor unit action potential (MUAP), a burst of electrical activity specific to the motor unit that produced it. The MUAP signal causes the muscle fibers attached to the motor unit to contract (Purves 2018).

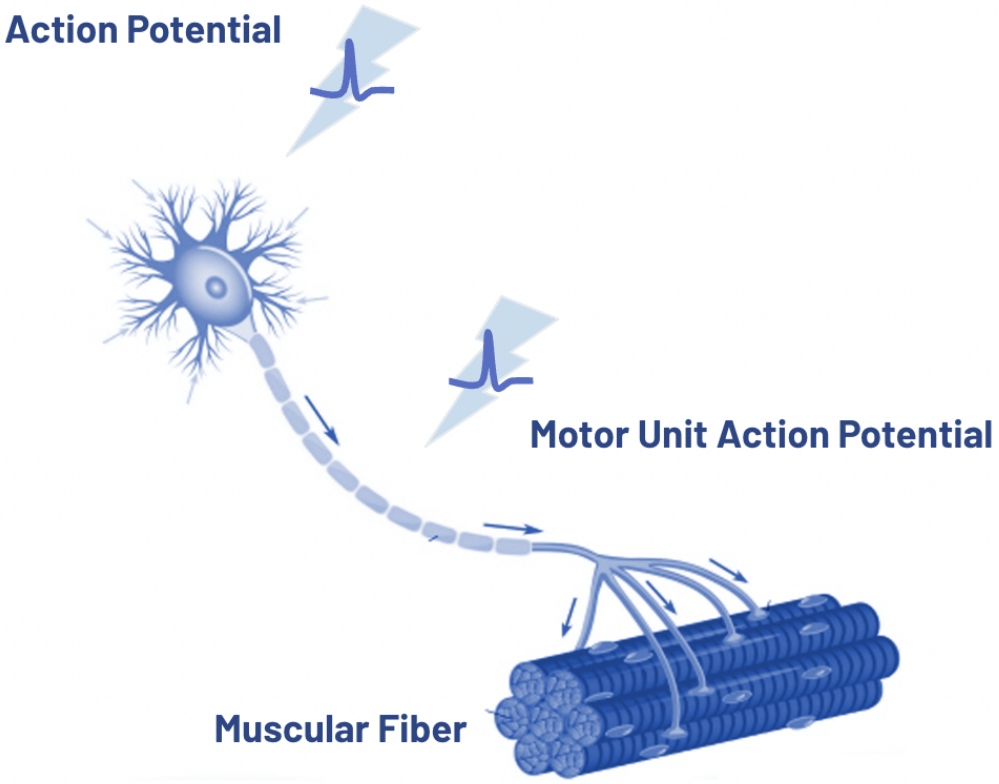


Figure 1: Diagram of a single motor unit. Modified from McLaughlin (2020).

The activation of several motor units (and their accompanying MUAPs) are required to move a single muscle. Therefore, the number of motor units involved in an isolated muscle movement can be determined by measuring the net electrical charge of a single muscle.

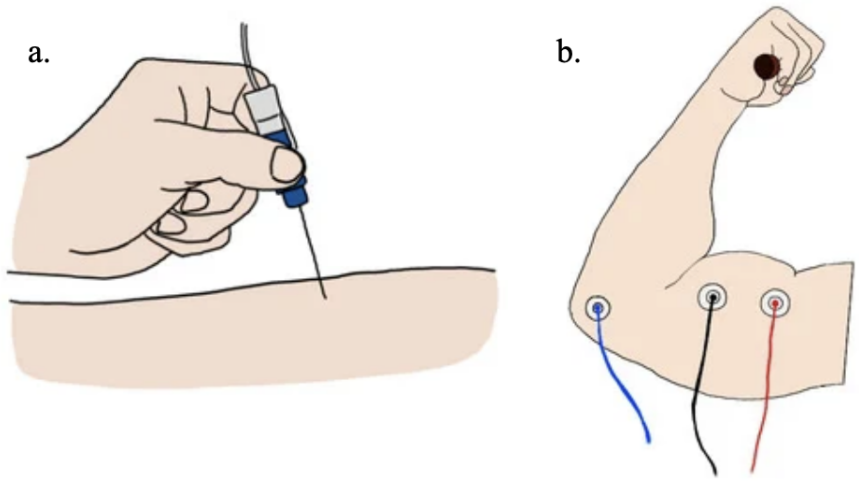


Figure 2: Obtaining electromyographs with a. surface electrodes and b. needle electrodes. Modified from Nam, Cha, and Park (2021).

The partner uses non-invasive electromyography (EMG) to measure the net firing of MUAPs across a single muscle, seen in **figure 2a** as opposed to invasive EMG, seen **figure 2b**. Surface electrodes are placed on the skin and voltage is measured as participants flex and relax a muscle ("EMG Decomposition Tutorial" 2006). The partner uses a grid of 64 surface electrodes, as seen in **figure 3**, allowing the collection of 64 streams of MUAP data across a single muscle. The raw signal collected by EMG is the result of many MUAP peaks constructively and destructively interfering with each other.

A blind source separation algorithm can decompose raw EMG signals into several individual electrical signals that can be ascribed to singular muscle units. The partner currently decomposes EMG signals using a free software from OT Bioelettronica, and its graphical user interface (GUI) can be seen in **figure 4**. This software determines the individual MUAP spike trains using a closed-source algorithm based off a paper published by Negro et al. (2016).

The decomposition of EMG signals is a critical technique for the partner's research. Decomposition allows researchers to gain a deeper understanding of which motor units are responsible for which parts of movement in a muscle. As is, the OT Bioelettronica product is an unideal solution for decomposing EMG signals for the partner's needs.

The creation of a custom Python package, `EMGdecomPy`, can address the issues with the OT Bioelettronica software, and contains some custom functionalities specific to the partner's use. OT Bioelettronica is closed-source, which obscures how the partner's experimental results are derived. By making `EMGdecomPy` open-source, the algorithm's code can be easily inspected, circumventing this problem. Additionally, the OT Bioelettronica software also has a 100 second time limit on experiments, which hinders the partner as they run experiments up to 5 minutes in length. In `EMGdecomPy`, there is no hard-coded time limit on experiments that can be run.

The transparency of `EMGdecomPy`'s code and the open-source licensing supports other parties expanding on and improving the existing functionalities. While made for the partner, it is our hope that `EMGdecomPy` will be a useful package for other researchers to explore the mechanisms underlying the brain's interactions with the body's muscles.

## Data Science Methods

The partner wants us to replicate the blind source separation algorithm used in Negro et al. (2016) to decompose raw EMG signals into their constituent motor unit spike trains. This algorithm provides advantages

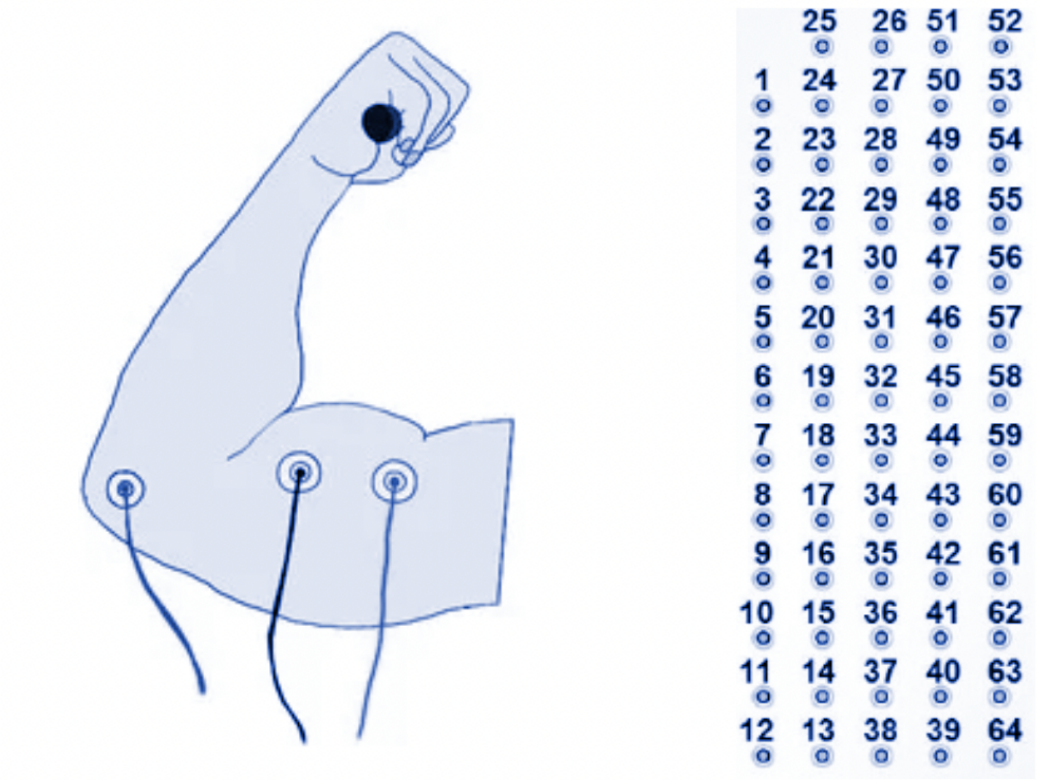


Figure 3: Diagram of surface EMG with 3 electrodes, and the 64 channel template used by the Sensorimotor Physiology Lab.

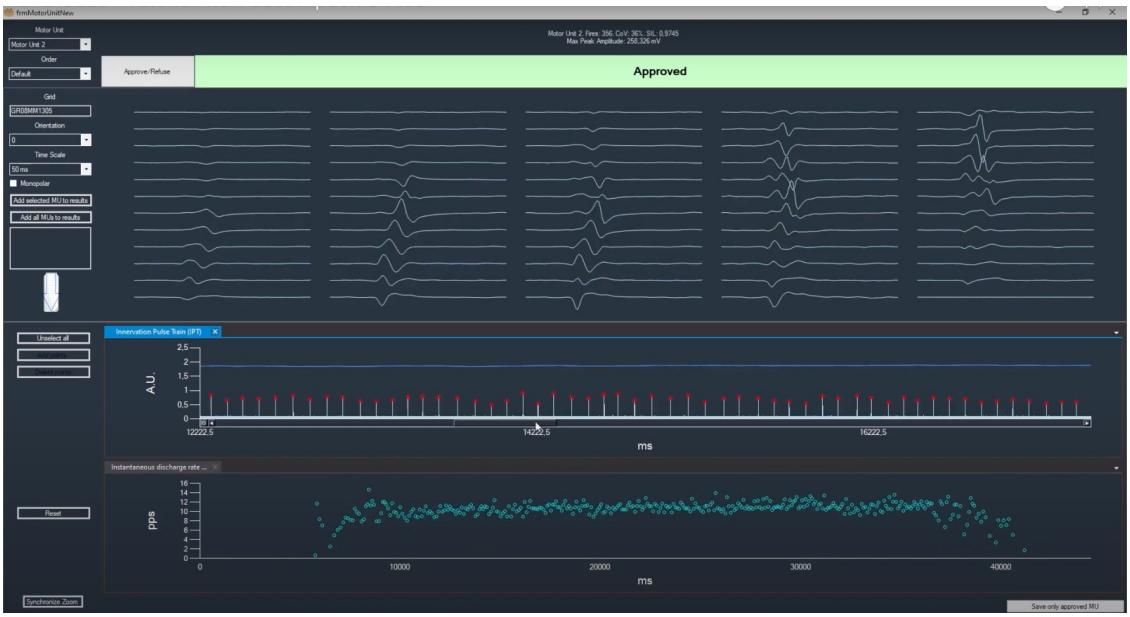


Figure 4: Graphical user interface of the OT Biolettronica software.

to other blind source separation algorithms in that it is an experimentally validated algorithm that allows for the decomposition of multi-channel invasive EMG and non-invasive EMG data.

Essentially, the blind source separation algorithm is an unsupervised machine learning model that iteratively extracts separation vectors from the data. Separation vectors are vectors that, when applied to the pre-processed data, separate a single motor unit spike train from unwanted noise and other motor unit spike trains. A motor unit spike train contains the firing times of a single motor unit, and it is the mixture of these that compose the raw EMG signal.

Separation vectors are extracted through broadly three steps:

1. The data is pre-processed.
2. Latent component analysis (LCA) is performed to estimate a separation vector.
3. The refinement process is used to increase the quality of the estimated separation vector.

Steps two and three are repeated for a predetermined amount of iterations of the overall blind source separation algorithm.

**Step 1: Pre-Processing.**

The first step of the algorithm is data pre-processing and consists of four sub-steps:

**a.** First, the data is *band-pass filtered.* That removes significant amount of noise from the data.

**b.** Then, the data is *centered* by subtracting the mean of each channel per channel. This process is essencial for LCA to extract all of the separation vectors.

**c.** Then each channel is *extended* by a certain number of time-delayed versions of that channel. This process converts the mixture of signals from a convolutive mixture to a linear instantaneous mixture. This is also a critical step in order to ready the data for LCA (Negro et al. 2016; Weenink 2012).

**d.** Finally, the data is *whitened*, so that the covariance matrix of the extended channels is equal to the identity matrix. This improves performance by computationally simplifying the convergence of the LCA step (Hyvärinen, Karhunen, and Oja 2001).

**Step 2: Latent Component Analysis.**

The pre-processed data then goes through the LCA step. The LCA step is based off of independent component analysis, where the separation vectors would be obtained by maximizing their statistical independence from each other. However, since the motor unit spike trains are extended along with the observations, they cannot be fully independent. This is why LCA extracts separation vectors by maximizing sparsity instead of independence. Intuitively, this is done because a singular motor unit spike train will be more sparse than the combination of multiple motor unit spike trains (Negro et al. 2016).

**a.** First, the separation vector is *initialized* by using a time instance of high activity in the whitened data, as these time instances are likely to correspond to multiple motor unit firings.

**b.** Using a contrast function that measures sparsity of the motor unit spike train, the estimated separation vector is *iteratively updated.* In each iteration, the estimated separation vectors are orthogonalized against previously accepted separation vectors and normalized, to increase the number of wxtracted unique motor unit spike trains.

**c.** The LCA step converges. This happens when the separation vector no longer changes, within a tolerance.

**Step 3: Refinement.**

After the separation vector is extracted, it goes through the refinement step. This is done because the latent component analysis may converge to unreliable estimates and through refinement the quality of the estimate is increased. The refinement step is an iterative algorithm that maximizes the regularity of the motor unit spike train. This process is carried out under the assumption that singular motor units fire off action potentials at a much more regular rate than combinations of motor units

$$@negro_muceli_castronovo_holobar_farina_2016$$

.

**a.** First, the motor unit spike train is estimated by applying the separation vector to the pre-processed data.

**b.** Then, the firing times are determined by applying the peak-finding algorithm from Virtanen et al. (2020). Of these firing times, the instances that correspond to small peaks in the spike train are separated away from the large peaks using the KMeans algorithm from Pedregosa et al. (2011). The firing times corresponding to small peaks are discarded as they likely correspond to the firing occurrence of more than one motor unit

$$@negro_{m}uceli_{c}astronovo_{h}olobar_{f}arina_{2}016$$

. The information from the accepted firing times are used to update the separation vector.

**c.** The iterative refinement process converges once the coefficient of variation of the time between firings increases.

Once the refinement process is done, the refined separation vector is accepted based on a user-defined threshold of either the silhouette score between the signal and the noise or the pulse-to-noise ratio. The accepted separation vectors correspond to the motor units that the blind source separation algorithm extracts from the raw signal.

A further improvement to the algorithm that we did not have time to implement would be a re-learning feature. The user would run the algorithm on a sample of the data, and based on that sample identify firing times that are inaccurate. Then they would run the algorithm on the rest of the data and based on the firing times identified as inaccurate, the algorithm would no longer make similar mistakes in the rest of the decomposition. Implementing this feature would be quite complex because it is unclear how this would be done. One idea was that we somehow change the initialization of the separation vectors so that they no longer identify the false firing times when applied to the pre-processed data. However, since the separation vector changes throughout the LCA and refinement processes, it would be hard to control the effect that it actually has on the estimated firing times. Another approach would be to influence the KMeans algorithm so that the threshold for the small peaks cluster includes the peaks that were as false firings, in the hopes that future peaks of similar size are also false firings. The downside to this approach would be that we may increase the amount of incorrect identifications of large peaks as small peaks, which are discarded.

The stakeholders affected by our blind source separation algorithm are researchers and those that would be affected by their research. This is why it is important our algorithm works properly so that researchers results are accurate and do not affect the general public adversely down the line. For example, if someone uses `EMGdecomPy` and obtains inaccurate results, and these results are used to inform a neuromuscular diagnosis down the line, it could greatly affect someone's life. Periodically, the results of our algorithm should be compared to others to obtain a second opinion on the decomposition of the EMG signal. We have not had the chance to thoroughly validate our algorithm, as the debugging process took a great deal of time. We have only received qualitative results, obtained by visually comparing MUAP shapes identified by `EMGdecomPy` and those Hug et al. (2021) identified using the `DEMUSE` tool. There are concerns with this approach as the `DEMUSE` software uses a similar but different algorithm than Negro et al. (2016). The `DEMUSE` software is a highly validated software in comparison to `OT Bioelettronica`, and therefore the partner wishes to compare our results to theirs.

## Data Product and Results

Our final data product consists of two elements. The first element is the Python package, `EMGdecomPy`, and its associated GitHub repository containing the replicated Negro et al. (2016) blind source separation algorithm. The second element is a Jupyter notebook containing a workflow leading up to an interactive dashboard visualizing the decomposition results.

Creating a Python package allows anyone who would like to decompose EMG signals into constituent motor units to simply download the package off of pip and run the `decomposition` function with the desired hyper-parameters on their data. Using a Python package to deliver our blind source separation algorithm not only increases accessibility but also readability of the code by breaking the blind source separation

algorithm into many different functions with informative docstrings. By giving the partner access to the GitHub repository for `EMGdecomPy`, we allow them to view the source code and even edit it if they desire to add more functionality in the future. These attributes of our data product satisfy the partner's desires to replicate the blind source separation algorithm from Negro et al. (2016) without experimental duration limits, while providing them with easy-to-understand open source code which means they will understand how their experimental results are derived.

The Jupyter notebook provides the partner with an interface in which to use `EMGdecomPy`. This notebook further facilitates our package's ease-of-use, as loading in the data, running the algorithm, and visualizing the data, will be laid out in a step-by-step process. Essentially, the Jupyter notebook will eliminate all confusion in how to use the package, and can be re-used to analyze different data, or to change hyper-parameters.

As mentioned above, we have not had the chance to thoroughly validate our algorithm. Preliminary results look promising however, as 3 out of 5 of the MUAP shapes identified by our algorithm were also identified by Hug et al. (2021) for the *Gastrocnemius lateralis* muscle with 10% contraction intensity. In the future, we would like to quantitatively validate our results on more muscle groups, using the metric recommended by Negro et al. (2016), the rate of agreement:

$$rate\ of\ agreement = \frac{c_j}{c_j + A_j + B_j}$$

Where $c_j$ is the number of discharges for the j-th motor unit that were identified by both algorithms, $A_j$ is the number of discharges identified by only one of the algorithms, and $B_j$ is the number identified by the other algorithm.

## Conclusions and Recommendations

Muscle movement is the result of coordination between the body's nervous and muscular system. To move a muscle, the brain releases action potential, a short burst of electrical activity. Action potential propagates via the nervous system, triggering a motor unit. The motor unit generates a specific type of action potential, known as motor unit action potential (MUAP), which causes the attached muscle fibers to contract.

A technique known as electromyography (EMG), uses electrodes to measure the electrical activity across a single muscle. The raw signal is the result of many motor unit action potential peaks constructively and destructively interfering with each other. A blind source separation algorithm, as described by Negro et al. (2016), can decompose raw EMG signals into several individual electrical signals that can be ascribed to singular muscle units.

Our final data product presented is an open-source Python package, `EMGdecomPy`, containing two main elements. The first component is the reconstructed decomposition algorithm as described in the pseudocode of Negro et al. (2016). The algorithm has been slightly adjusted and validated using data from Hug et al. (2021). The second component is a Jupyter notebook that provides a template for users to interactively view the decomposed output of the algorithm.

Although `EMGdecomPy` has not been thoroughly validated yet, but preliminary qualitative results show that the package identifies 3 out of 5 of the same motor units as the DEMUSE software when run on the same *Gastrocnemius lateralis* muscle with 10% contraction intensity data obtained from Hug et al. (2021). Future work includes expanding the qualitative validation to other muscle groups and contraction intensities provided by Hug et al. (2021), and performing quantitative analysis using the rate of agreement.

## References

"EMG Decomposition Tutorial." 2006. *EMG Decomposition Tutorial.* http://www.emglab.net/emglab/Tutorials/EMGDECOMP.html.

Forbes, Scott C., Jonathan P. Little, and Darren G. Candow. 2012. "Exercise and Nutritional Interventions for Improving Aging Muscle Health." *Endocrine* 42 (1): 29–38. https://doi.org/10.1007/s12020-012-9676-1.

Hug, Francois, Simon avrillon, Alessandro Del Vecchio, Andrea Casolo, Jaime Ibáñez, Stefano Nuccio, Julien Rossato, Aleš Holobar, and Dario Farina. 2021. "Analysis of motor unit spike trains estimated from high-density surface electromyography is highly reliable across operators," February. https://doi.org/10.6084/m9.figshare.13695937.v1.

Hyvärinen, Aapo, Juha Karhunen, and Erkki Oja. 2001. "What Is Independent Component Analysis?" In *Independent Component Analysis*, 158–61. Wiley.

McLaughlin, Katy. 2020. "Motor Neuron - the Definitive Guide." *Biology Dictionary*. https://biologydictionary.net/motor-neuron/.

Nam, Dahyun, Jae Min Cha, and Kiwon Park. 2021. "Next-Generation Wearable Biosensors Developed with Flexible Bio-Chips." *Micromachines* 12 (1): 64. https://doi.org/10.3390/mi12010064.

Negro, Francesco, Silvia Muceli, Anna Margherita Castronovo, Ales Holobar, and Dario Farina. 2016. "Multi-Channel Intramuscular and Surface EMG Decomposition by Convolutive Blind Source Separation." *Journal of Neural Engineering* 13 (2): 026027. https://doi.org/10.1088/1741-2560/13/2/026027.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.

Purves, Dale. 2018. "Lower Motor Neuron Circuits and Motor Control." In *Neuroscience*, 357–80. Sinauer Associates, an imprint of Oxford University Press.

Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, et al. 2020. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." *Nature Methods* 17: 261–72. https://doi.org/10.1038/s41592-019-0686-2.

Weenink, David. 2012. "Blind Source Separation." *Blind Source Separation*. https://www.fon.hum.uva.nl/praat/manual/blind_source_separation.html.