# A neural network-based web tool for the quick selection of isotachophoresis electrolytes.

Amit Jangra

today

**Abstract**

We present a neural network-based web tool to predict whether the selected electrolyte system can focus given analyte. The tool is called Neural Network-based Application for Fast Electrolyte Selection in Isotachophoresis (NAFES). NAFES uses cross-browser compatibility to provide an easy-to-use tool to select an electrolyte system. The code uses the trained neural network for prediction and provides a satisfactory result, requiring no prior installation and compilation. NAFES also offer a large database of commonly used monovalent species and their relevant physiochemical properties. We present a validation of prediction from NAFES by comparing them to experimental data of a well-controlled ITP process. The network yields an accurate estimate for selecting electrolyte systems for both anionic and cationic. The tool is available at `https://amit2745.pythonanywhere.com/`

***Keywords:***Machine learning, isotachophoresis, neural network.

## 1 Introduction

Isotachophoresis(ITP) [1] is an electrophoresis process which is used for the separation and focusing of ionic species, including inorganic ions [2], organic acids and bases [3], proteins and nucleic acids [4]. In ITP, the analytes separate according to the difference in their electrophoretic mobility under an applied electric field. For the separation, the sample containing these analytes is introduced into a capillary between the leading electrolyte (LE) and trailing electrolyte (TE). These two electrolyte solutions contain a common counter ion and co-ions as the sample ions. The co-ions of LE and TE have relatively high and low effective mobility, respectively. When an electric field is applied, ions start migrating according to mobility. These migrating ionic species arrange themselves into the individual zone. The counter-ion migrates from LE to TE zone. Sample ions with effective mobility more than the TE co-ion but less than LE co-ion focus between zones of LE and TE. Adjacent zones in ITP are separated by sharp zone boundaries, which result from a balance between electromigration and diffusive fluxes.

In ITP, sample ions depending on initial concentration can focus in two modalities. Analytes present in sufficient amount focus in 'plateau mode'. This mode is characterized by relatively long zones of locally uniform concentration separated by thinner zone boundaries. For trace quantities, analyte species may not develop into such plateaus. Instead, multiple trace analyte species bounded by LE and TE focus into completely overlapping peaks whose widths are governed by diffuse TE-to-LE interface. This regime is termed as 'peak mode'. Peak mode ITP is used to simultaneously purify and pre-concentrate, followed by either

direct on-chip detection or downstream analysis. On the other hand, plateau mode ITP allows the separation and detection of multiple sample ions.

The success of ITP analysis requires that ionic species should have sufficient differences in their effective mobility. Several parameters such as acid dissociation constant ($pK_a$), temperature and pH in the zone influence the mobility of analytes, and a good choice of electrolyte system( LE and TE) makes the separation possible. The application involving the detection and quantitation of ionic species from a sample mixture requires an electrolyte system that allows separation and preconcentration. In trace analysis from the samples with complex matrix, it is necessary to choose the electrolyte system to focus the analyte of interest while excluding others selectively. The same applies to the application where preconcentration is desired for accelerating reactions [5]. In any case, the selection of LE and TE is always of paramount importance.

The selection of an electrolyte system depends on several factors, such as the choice of (i) solvent, (ii) LE and TE co-ions, (iii) buffering counter ion, and (iv) pH of LE. The commonly used solvents are water and methanol, with water being a preferable solvent due to its superior solubility. The recommended choice of co-ions of LE and TE in the separation of anionic species are the anions of the strong and weak acid, respectively [6–11]. Similarly, in the separation of cationic species, the LE ion is the cation of a strong base, and TE ion is the cation of a weak base. The buffering counter-ion is selected such that LE co-ion has high and TE co-ion has low effective electrophoretic mobility. Therefore, we must choose a system in such a way that ionic species to be separated have the maximum difference in effective mobility.

In general, there are two possible approaches to choose a suitable electrolyte system for qualitative and quantitative analysis of components in a matrix. Either a suitable electrolyte system has already been described in the literature and can be applied with slight changes depending on the accessible instrumentation, or a new electrolyte system has to be designed. In the latter case, the LE and TE is compiled with respect to the physiochemical properties of both analytes and other components of the sample to ensure sufficient ionization of analytes and necessary difference in their mobilities. The principle quantities used in the considerations are the values of dissociation constants ($pK_a$) and ionic mobilities. These quantities can be found in tables, and relation enabling their calculation from other quantities or experimental values are also known.

The advent of various simulation tools such as SIMUL [12], SPRESSO [13], SPYCE [14], and CAFES [15] have simplified the procedure to select suitable electrolyte systems. All these simulation tools are based on the solution of coupled mass conservation equations for ionic species in an electrolyte using numerical methods. While these simulations can accurately predict the dynamics of ITP in a few minutes, running these simulations requires a basic understanding of numerical methods parameters such as time steps, grid points, and initial conditions. However, the selection of suitable electrolyte systems has relied to a large extent on empirical guidelines and experience.

In this contribution, we present a simple-to-use tool called Neural Network-based Application for Fast Electrolyte Selection in Isotachophoresis (NAFES). It is capable to recommend appropriate electrolyte system for ITP by employing machine learning methods. NAFES can be used through various web browsers on devices running any operating system, including mobile devices. It includes a database of LE co-ion, TE co-ion, and counter ions in addition to custom user-defined species. The trained neural network enables accurate and fast selection of an electrolyte system with the relatively minor trade-off of offering only for monovalent species. Therefore, this study aims to investigate the novel application of neural networks for the quick estimate of LE and TE for preconcentration and separation. We only considered the monovalent species for anionic as well as cationic. We also present the experimental validation of NAFES using the data from a well-controlled ITP process.

# 2 Material and Methods

Machine learning is a subset of artificial intelligence that can learn from the training data and predict the future outcomes. The learning in which the training data comprises inputs with and without any corresponding output is known as supervised learning and unsupervised learning, respectively. Based on the output, supervised learning has two main categories (i) classification, where the output values are a finite set of classes, and (ii) regression, where the output values are the real number. While generating the inputs and outputs is often a laborious task, supervised learning is easy to understand, and their performance is easily measurable. Therefore we formulate the problem, the selection of an electrolyte system, as supervised learning and collect the data set of input/output pair. We categorized the electrolyte system into two classes based on sample ion (focusing or not-focusing), which makes it a binary classification problem.

## 2.1 Database preparation

In any machine learning problem, building the predictive model requires data collection. The predictive model would be as good as the data, so good data collection is of the utmost importance in developing high-performing models. The collected data set comprises multiple data points where each data point represents an entity to be analysed. So data points can be anything; in our case, the data point consists of ionic species of leading ion (L), trailing ion(T), counter ion (C), and sample ion (S). To create the data set, one has to measure and collect a number of features that describe the properties of data points. Therefore, we selected the limiting ionic mobilities ($\mu$) and acid dissociation constant (p$K_a$) of each ion to create the eight features where each feature represents one dimension in the feature space. Next, one must decide the number of data points to train the learning algorithm. The neural network requires a lot of data to train, so we collect around 10,000 data points.

We created the input data set for anionic and cationic ITP by choosing the value of each input feature from the uniform distribution in the range mentioned in the supplementary. Each data set contains 10,000 combinations of L, T, C, and S. The corresponding target value, either focusing or not-focusing, was obtained by running the diffusion-free model [16] simulation without any ionic strength effect. The diffusion-free model takes into account detailed chemical equilibrium calculations and is well suited for unsteady and steady-state plateau mode ITP problems. It can handle both an arbitrary number of weak and strong electrolytes. The concentration of ionic species used for the simulation has given in Table 1. The concentration of a counter ion in LE was twice that of the leading ion.

**Table 1:** Concentration of ionic species used in the simulation

| ITP | leading ion (mM) | trailing ion (mM) | counter ion (mM) | sample ion (mM) |
|---|---|---|---|---|
| Anionic | 10 | 5 | 20 | 1 |
| Cationic | 10 | 5 | 20 | 1 |

The condition for focusing a sample ion X between LE and TE zone is indicated in Eqs.(1) - (6) [17].

$$|\bar{\mu}_L^{LE}| > |\bar{\mu}_S^{LE}| \tag{1}$$

$$|\bar{\mu}_L^{LE}| > |\bar{\mu}_T^{LE}| \tag{2}$$

$$|\bar{\mu}_L^{S}| > |\bar{\mu}_S^{S}| \tag{3}$$

$$|\bar{\mu}_S^{S}| > |\bar{\mu}_T^{S}| \tag{4}$$

$$|\bar{\mu}_L^{TE}| > |\bar{\mu}_T^{TE}| \tag{5}$$

$$|\bar{\mu}_S^{TE}| > |\bar{\mu}_T^{TE}| \tag{6}$$

We used absolute values to account for both anionic and cationic ITP. The subscript(L) and superscript(LE) in the absolute effective mobility $|\bar{\mu}_L^{LE}|$ represent ionic species and zone, respectively. In the LE zone, as in Equations (1) and (2), the absolute effective mobility of the leading ion is more than the trailing ion and sample ion. Similarly, Eqs. (3,4) and (5,6) represent the condition in the sample zone and adjusted TE zone, respectively. When all the above conditions were satisfied, we labelled the input with focusing; otherwise not-focusing.

The final data set of anionic contain 7217 data points from class 0 and 2783 data point from class 1, whereas, in the cationic data set, 6653 data points belong to class 0 and 3347 to class 1. In any machine learning task to learn from data, it is necessary to convert the categorical output into a numeric value, so we assigned the labels not-focusing and focusing with the value of 0 and 1, respectively. An example of a final anionic database is shown in Table 2, and a similar database holds for cationic.

**Table 2:** An example of the database used by neural network

| Sr No | mobility ($10^{-9}m^2V^{-1}s^{-1}$) | | | | p$K_a$ | | | | Output |
|---|---|---|---|---|---|---|---|---|---|
| | LE | TE | BG | Analyte | LE | TE | BG | Analyte | |
| 1 | -31.3 | -31.3 | 34.9 | -24.7 | 4.018 | 4.018 | 6.75 | 3.437 | 0 |
| 2 | -79.1 | -28.0 | 29.5 | -42.4 | -2.0 | 6.095 | 8.076 | 4.756 | 1 |

## 2.2 Tools and software

We used Python for simulation and machine learning. The NN algorithm is implemented in the machine learning library Scikit-learn [18] under MLPClassifier [19, 20]. Scikit-learn library GridSearchCV was used for cross-validation and hyperparameter tuning of the neural network. The final model is deployed on a web-based application.

## 2.3 Neural Network algorithm

Neural network (NN) or artificial neural network(ANN) is a classification and pattern recognition technique that can be used to learn highly non-linear functions [20]. As shown in Figure 1(a), Multilayer perceptron is one form of NN representing a non-linear mapping between input and output.

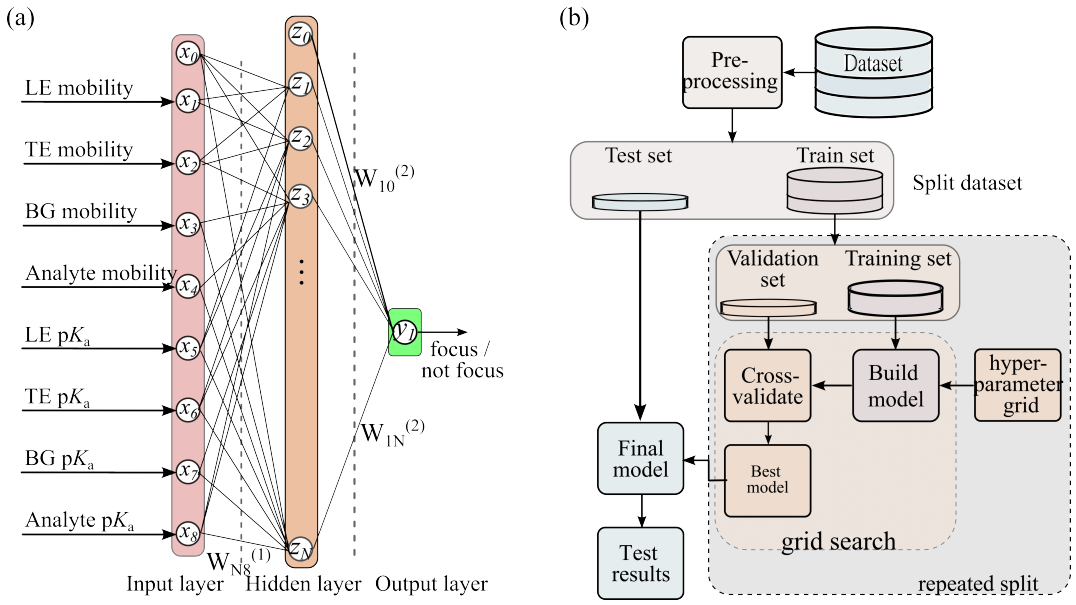


**Figure 1:** (a) Schematic of neural network where input, hidden and output variable are represented by nodes and weights are represented by links in which bias parameters are denoted by links coming from $x_0$ and $z_0$ (b) Schematic illustrating the detailed work flow of model training and validation with the tuning of hyper-parameters of neural network.

Here, each node $x_i$ in the input layer represents the input feature such as mobilities and p$K_a$ along with bias node $x_0$. The bias node value is always 1. These input features were transformed into N linear combination of the form

$$a_j = \sum_{i=1}^{8} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \tag{7}$$

where $j = 1, ....., N$, and the superscript(1) indicate that parameters are in the first 'layer'. The parameters $w_{ji}^{(1)}$ and $w_{j0}^{(1)}$ are weights and biases from input $i$ to unit $j$. The quantities $a_j$ are known as activations. Each of these parameters is then transformed using a non-linear activation function as

$$z_j = h(a_j), \tag{8}$$

where

$$h(a) = max(0, a), \tag{9}$$

is rectifying nonlinearity, also know as a rectified linear unit (relu). Each quantity $z_j$ represents the hidden node, and the layer containing it is called the hidden layer since the activation values are not directly accessible from outside the network. Following 7, these values are again linearly combined to give output unit activations

$$a_k = \sum_{i=1}^{N} w_{kj}^{(2)} z_j + w_{k0}^{(2)}, \tag{10}$$

where $k = 1, ...., K$ and K is the total number of outputs. This transformation corresponds to the second layer of the network. Finally, the output unit activation is transformed using an appropriate activation function to give a set of network output $y_k$. In our case, for the binary classification problem, we transformed using a logistic sigmoid function so that

$$y_k = \sigma(a_k), \tag{11}$$

where

$$\sigma(a) = \frac{1}{1 + e^{-a}}. \tag{12}$$

The output of the sigmoid function is between 0 and 1, representing the probability of output belonging to class 1. The default threshold limit of activation is set up at 0.5 when a = 0. If the sigmoid output is larger than or equal to 0.5, it belongs to class 1; if the output is smaller than 0.5, it belongs to class 0. All the above equations, Eqs.(7)-(11) can be combined together to give

$$y_k = \sigma(\sum_{i=1}^{N} w_{kj}^{(2)} h(\sum_{i=1}^{8} w_{ji}^{(1)} x_i + w_{j0}^{(1)}) + w_{k0}^{(2)}). \tag{13}$$

These weight parameters as in Equation (13) are learned during training in backpropagation [20]. The training begin by assigning the random values to these weights. The predicted output from Equation (13) is compared with the actual output and the error is calculated. The backpropagation training algorithm then take this calculated error and update the weights backwards from output layer to input layer. Now, the update weights are used to predict the output and this process repeats untill the error is within acceptable limits.

## 2.4 Model training and validation

Neural networks must be trained and tested before it is deployed. The whole dataset is usually split into two sets to provide training and testing sets. Training the neural network consists of running the neural network over the training set until the neural network learns to recognize the training set with a sufficiently low error rate. Testing occurs when the neural network's results are evaluated. We trained two neural networks, one for anionic and the other for cationic. The training of both networks is done separately using individual data sets, and the general procedure is described below.

We split the data set into two sets, the training set with 80% and the test set with 20%, using the 'train test split' module of scikit-learn. In particular, neural networks require all the features to vary on a similar scale. So we scaled features of training and test sets in the range 0 and 1 using the MinMaxScaler module of scikit-learn. Next, we used the training set to train the neural network and the test set to evaluate the network. We used the 5-fold cross-validation on the training set for internal validation and tuned the hyper-parameters such as the number of the hidden layer, the number of nodes in each hidden layer, and a learning rate to update the weight parameters [19]. We performed the tuning of hyperparameters and cross-validation with the help of the GridSearchCV module. The 'best estimator ' attribute of the GridSearchCV provided the best model. A detailed workflow of analysis is illustrated in Figure 1(b).

The best neural network for anionic comprises three hidden layers with 16 hidden nodes in each hidden layer, whereas the best neural network for cationic comprises three hidden layers with 12 hidden nodes in each hidden layer. The final performance of each network was evaluated using the unseen 20% test set. We declared class 0, the negative and class 1, the positive class. We evaluated the neural network based on several parameters, including accuracy, precision, recall, and f1-score [21]. In Eqs. (14) - (17), *TP, TN, FP*, and *FN* refer to true positive, true negative, false positive, and false negative, respectively.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{14}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{15}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

$$\text{f1} - \text{score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{17}$$

The accuracy represents the fraction of sample ions for which the right electrolyte system was predicted. It is a good measure when all classes are perfectly balanced. But in our case, there was a slight imbalance in the classes, so we also calculated precision and recall. Ideally, these two quantities should be 1. Equations (15) and (16) apply to the positive class only but can be used for the negative class by interchanging positive and negative. Precision measures how many sample ions predicted as focusing are actually focusing. Recall, on the other hand, measures how many of the focusing samples are captured by the focusing predictions. We can combine precision and recall into the f1-score, the geometric mean of these two quantities.

When evaluated on the test set, the neural network achieved an accuracy of 97.35% and 96.8% for anionic and cationic, respectively. It is best to represent the binary classification result in confusion matrices, as shown in Figure 2. In the matrix, the main diagonal indicates correctly classified labels, whereas the counter diagonal shows misclassified labels.

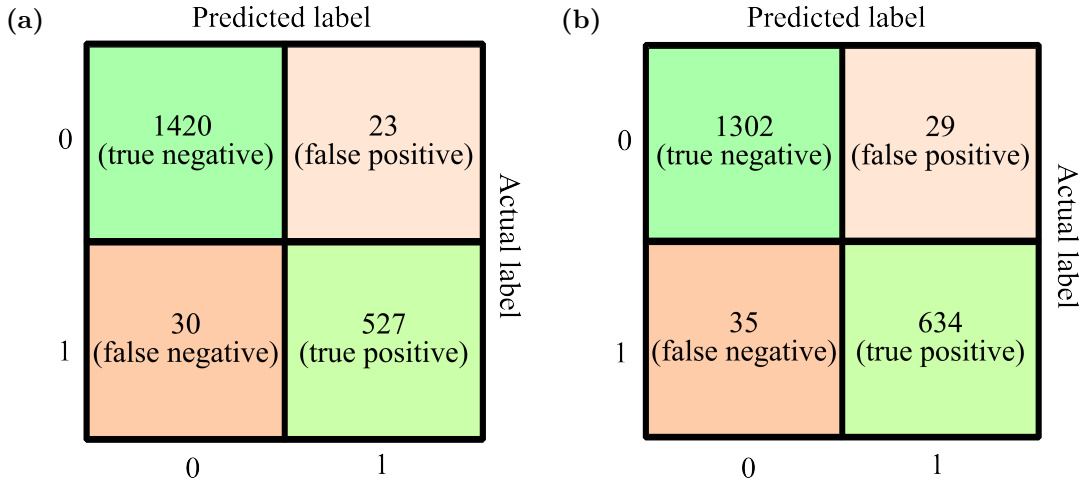


**Figure 2:** Classification result of test set represented as confusion matrix (0 = not focusing and 1 = focusing) for (a) Anionic (b) Cationic. The actual label and predicted label represent the labels from simulation and model respectively.

Figure 2(a) shows that the neural network for anionic classified 1947 data points correctly, and only 53 were misclassified out of 2000 data points. For cationic, as shown in Figure 2(b), out of 2000 data points, 1936 were correctly classified, and 64 were misclassified. As discussed above, we calculated the precision and recall of both classes, and the result is shown in Table 3.

**Table 3:** The classification result both for anionic and cationic.

| classes | Anionic | | | Cationic | | |
|---|---|---|---|---|---|---|
| | precision | recall | f1-score | precision | recall | f1-score |
| 0 | 0.98 | 0.98 | 0.98 | 0.97 | 0.98 | 0.98 |
| 1 | 0.96 | 0.95 | 0.95 | 0.96 | 0.95 | 0.95 |

# 3 Result and Discussion

The neural networks-based web tool outputs the predicted result indicating whether the analyte is focusing or not-focusing in the chosen electrolyte system. To verify the applicability of our tool, we compare the predicted result with already published experimental results for anionic and cationic.

We begin by verifying the trained neural network and its implementation in NAFES with the experimental result of anionic ITP provided by D.Chambers et al. [22]. The LE ion was 100 mM MES and TE ion was 100 mM Tricine, and the counter-ion was 200 mM Bistris. Three sample ions, MOPS, HEPES and Alexa Fluor (AF488), were analysed. Both MOPS and HEPES focus between LE and TE, whereas AF488 migrates from TE to LE zone without focusing. NAFES also shows that MOPS and HEPES are focusing and AF488 is not-focusing. We also compared NAFES with the experimental result of Everaerts et al. [1] in which many sample ions were segregated with the help of leading ion (chloride) and trailing ion (MES). Table 4 shows the comparison between the experiment and NAFES.

Next, we compare NAFES for the cationic ITP using experimental data of Garcia-Schwarz et al. [23] to separate amino acids(lysine and arginine). The electrolyte system consisted of LE(100 mM ETA + 200 mM Tricine) and TE (20 mM Tris + 40 mM Tricine), focuses lysine and arginine. The operating pH was such that the sample ions arginine and lysine were monovalent positively charged species, allowing us to use the NAFES. The result from NAFES matches the experimental result, as shown in Table 4.

**Table 4:** The detailed result comparison for anionic and cationic with experimental and NAFES

| ITP | LE | TE | Sample | experimental | NAFES | ref |
|---|---|---|---|---|---|---|
| Anionic | MES + Bistris | Tricine + Bistris | MOPS | Focused | Focused | [22] |
| | | | HEPES | Focused | Focused | |
| | | | AF488 | Not-Focused | Not-Focused | |
| | chloride + Histidine | MES + Histdine | Perchloric acid | Focused | Focused | [1] |
| | | | Formic acid | Focused | Focused | |
| | | | Acetic acid | Focused | Focused | |
| | | | Lactic acid | Focused | Focused | |
| | | | Caproic acid | Focused | Focused | |
| Cationic | ETA + Tricine | Tris + Tricine | lysine | Focused | Focused | [23] |
| | | | arginine | Focused | Focused | |
| | | | R6G | Not-Focused | - | |

# 4 Conclusion

We demonstrated the implementation and experimental validation of NAFES, an NN-based tool to select the LE, TE and counter ions for ITP. This highly interactive tool estimates the electrolyte system accurately and quickly using the neural network. NAFES offer a user-friendly GUI that can be used on any platform and provide the result in just one click. For simplicity and to decrease the prediction time, the code uses a trained model; however, it only applies to monovalent species. We validated the tool using experimental data from both anionic and cationic experiments. NAFES was able to predict the electrolyte system within a few milliseconds. The tool is available for free at `https://amit2745.pythonanywhere.com/` and requires no license or compilation.

# References


[1] F.M. Everaerts, F.E.P. Mikkers, and Th.P.E.M. Verheggen. Isotachophoresis. *Separation and Purification Methods*, 6(2):287–351, 1977.

[2] Pavel Blatný and František Kvasnička. Application of capillary isotachophoresis and capillary zone electrophoresis to the determination of inorganic ions in food and feed samples. *Journal of Chromatography A*, 834(1):419–431, 1999.

[3] Jana Sádecká and Jozef Polonský. Determination of organic acids in tobacco by capillary isotachophoresis. *Journal of Chromatography A*, 988(1):161–165, 2003.

[4] Anita Rogacs, Lewis A. Marshall, and Juan G. Santiago. Purification of nucleic acids using isotachophoresis. *Journal of Chromatography A*, 1335:105–120, 2014. Editors' Choice VIII.

[5] Moran Bercovici, Crystal M. Han, Joseph C. Liao, and Juan G. Santiago. Rapid hybridization of nucleic acids using isotachophoresis. *Proceedings of the National Academy of Sciences*, 109(28):11127–11132, 2012.

[6] Takeshi Hirokawa and Yoshiyuki Kiso. Preparative procedures in isotachophoresis. *Journal of Chromatography A*, 658(2):343–354, 1994.

[7] Ernst Kenndler. Applications of isotachophoresis. *TrAC Trends in Analytical Chemistry*, 2(9):202–206, 1983.

[8] Zdena Malá, Pavla Pantůčková, Petr Gebauer, and Petr Boček. Advanced electrolyte tuning and selectivity enhancement for highly sensitive analysis of cations by capillary itp–esi ms. *ELECTROPHORESIS*, 34(5):777–784, 2013.

[9] Petr Gebauer, Zdena Malá, and Petr Boček. Recent progress in analytical capillary itp. *Electrophoresis*, 30(1):29–35, 2009.

[10] Lihui Wang, Dayu Liu, Hao Chen, and Xiaomian Zhou. A simple and sensitive transient itp method for on-chip analysis of pcr samples. *Electrophoresis*, 29(24):4976–4983, 2008.

[11] Supreet S Bahga, Moran Bercovici, and Juan G Santiago. Ionic strength effects on electrophoretic focusing and separations. *Electrophoresis*, 31(5):910–919, 2010.

[12] Bohuslav Gaš and Petr Bravenec. Simul 6: A fast dynamic simulator of electromigration. *Electrophoresis*, 42(12-13):1291–1299, 2021.

[13] Moran Bercovici, Sanjiva K Lele, and Juan G Santiago. Open source simulation tool for electrophoretic stacking, focusing, and separation. *Journal of Chromatography A*, 1216(6):1008–1018, 2009.

[14] Supreet Singh Bahga and Prateek Gupta. Electrophoresis simulations using chebyshev pseudo-spectral method on a moving mesh. *Electrophoresis*, 43(5-6):688–695, 2022.

[15] Alexandre S Avaro, Yixiao Sun, Kaiying Jiang, Supreet S Bahga, and Juan G Santiago. Web-based open-source tool for isotachophoresis. *Analytical Chemistry*, 93(47):15768–15774, 2021.

[16] Supreet S Bahga, Govind V Kaigala, Moran Bercovici, and Juan G Santiago. High-sensitivity detection using isotachophoresis with variable cross-section geometry. *Electrophoresis*, 32(5):563–572, 2011.

[17] Ashwin Ramachandran and Juan G Santiago. Isotachophoresis: Theory and microfluidic applications. *Chemical Reviews*, 122(15):12904–12976, 2022.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Geoffrey E Hinton. Connectionist learning procedures. In *Machine learning*, pages 555–610. Elsevier, 1990.

[20] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[21] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[22] Robert D Chambers and Juan G Santiago. Imaging and quantification of isotachophoresis zones using nonfocusing fluorescent tracers. *Analytical chemistry*, 81(8):3022–3028, 2009.

[23] Giancarlo Garcia-Schwarz, Anita Rogacs, Supreet S Bahga, and Juan G Santiago. On-chip isotachophoresis for separation of ions and purification of nucleic acids. *JoVE (Journal of Visualized Experiments)*, (61):e3890, 2012.