


APPROVAL SHEET

Title of Dissertation: Bi-potential Pulsed Electrochemical Detection of Copper

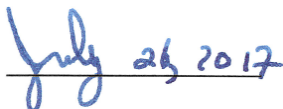
Name of Candidate: William M. Cuning
Doctor of Philosophy, 2017

Dissertation and Abstract Approved: _____



Dr. William R. LaCourse
Professor
Dept. of Chemistry and Biochemistry

Date Approved: _____



ABSTRACT

Title of Document: BI-POTENTIAL PULSED ELECTRO-
CHEMICAL DETECTION OF COPPER

William M. Cuning, Ph.D. 2017

Directed By: Professor William R. LaCourse, Department of
Chemistry and Biochemistry

Copper in its pure form is a lustrous red-orange metal that is valued for its electrical, mechanical and aesthetic properties. Aqueous copper is often found in either the +1 or +2 oxidation state, and is used in a variety of applications including: as a dietary supplement, as a biocidal marine paint additive, as an algal and fungal control agent, as an antimicrobial agent, and as the treatment for the protozoan *Cryptocaryon irritans* (a.k.a. Marine “Ich”). The level of copper in drinking water is strictly monitored by the US EPA, with a maximum contaminant limit of 1.3 parts per million.

Literature is replete with analytical methods for the analysis of aqueous copper, with research on this topic dating back centuries. New research seeks to simplify the

analysis and broaden matrix compatibility to include beverages and seawater, but reported methods often require extensive sample preparation and preconcentration or, in the case of electrochemical methods, complex chemical electrode modification to achieve the reported sensitivity and selectivity.

This research explores the application of bi-potential pulsed electrochemical detection (BPED) to the analysis of copper and other metals after ion chromatography separation. Copper was successfully separated and quantified in the presence of cadmium and lead in fresh water samples. Additionally, copper was quantified in various dietary supplements used to treat copper deficiency in humans. Additional work is presented that shows BPED is capable of being developed into a stand-alone analytical technique for copper at gold and glassy carbon electrodes – with emphasis on waveform development and optimization techniques. Directions for future work are also explored.

BI-POTENTIAL PULSED ELECTROCHEMICAL DETECTION OF COPPER

By

William M. Cuning

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

2017

© Copyright by
William M. Cuning
2017

Dedication

I dedicate this work first and foremost to my wife, Jacquelyn. You have always known just what to say to keep me on track and focused on my goals. Without you, I would never have stood a chance. You are my greatest blessing. Thank you.

Mema – You were the one who first told me about chemistry, and gave me my first microscope. You were the one who told me that water was one atom of oxygen and two atoms of hydrogen. You encouraged me to always seek out answers to my questions. The love of learning and the curiosity you fostered in me has guided me through my entire academic career.

Finally, I would like to dedicate this work to Mr. Richard T. Hamilton, Sr. His words of wisdom, encouragement, and support will always be etched on my heart. I hope he would be proud of what I have accomplished.

Acknowledgements

I would first and foremost like to thank my advisor, Dr. William R. LaCourse. Thank you for all the guidance and assistance, even when it meant coming over and troubleshooting with me step by step. Thank you for the infinite patience you've shown over the years. You have taught me so much over the course of my time here, not only about chemistry, but about life.

I would also like to thank my committee members: Dr. Brian Cullum, Dr. Marie-Christine Daniel-Onuta, Dr. Paul J. Smith, Dr. Ryan White, and Dr. Yordan Kostov for their assistance and guidance throughout this process. Without their input, I would not be half the writer or scientist I am now.

I would like to also acknowledge the assistance of my fellow lab members: Sarah Bass, Jacquelyn Cunning, Dr. Jennifer Fedorowski, Dr. Andrea Gray, DeLauren McCauley, and Dr. Gregory Winter. Additional thanks go to Josh Wilhide, Ian Shaffer, and Maggie LaCourse of the MCAC for all their help over the years looking over my work and helping me polish it up. All of you have made me understand just how supported and cared for I have been these past years.

Finally, I would like to thank my family and my friends who have shown me support and, when necessary, reminded me why I went through all this.

A former professor told me that the word awesome should be interchangeable with apocalyptic – as that is the power of the word awesome; therefore, I conclude, you're all awesome.

Table of Contents

Dedication	ii
Acknowledgements	iii
Table of Contents	iv
List of Tables	viii
List of Figures	ix
Chapter 1: Introduction	1
Importance of Metals	1
General Remarks	1
Copper	4
Review of Established Methods	5
General Reviews	5
US EPA Standard Methods	6
Analytical Separation Techniques	7
Detection Methods	9
Optical Detection Methods	10
Mass Spectrometric Methods	11
Electrochemical Detection Methods	11
Bi-potential Pulsed Electrochemical Detection (BPED)	20

Research Objectives	22
Chapter 2: Experimental	23
Instrumentation	23
IC-BPED Instrumentation.....	23
Electrochemical Instrumentation	24
Spectrophotometric Instrumentation.....	25
Reagents and Solutions	26
Chemicals and Reagents	26
Preparation of Chromatographic Mobile Phases and Solutions	26
Procedures and Sample Handling	28
Polishing of Electrodes	28
Cyclic Voltammetry and Anodic Stripping Voltammetry	29
Initial BPED Waveform Development	31
Optimization of BPED for IC Analysis	31
Optimization of BPED Waveforms for Benchtop Applications.....	31
Data Analysis and Handling	32
Sample Storage Recommendations.....	32
Safety Considerations	33
Chapter 3: IC-BPED analysis of Copper, Lead and Cadmium.....	35
Introduction.....	35
Voltammetric Studies of Copper Redox system.....	36
Cyclic Voltammetry Results	37

Anodic Stripping Voltammetry.....	40
BPED after Ion Chromatography.....	42
Initial BPED Parameters	42
Initial IC-BPED Results.....	42
Effects of [KNO ₃] on Chromatographic Parameters	43
Optimization Results.....	44
Method Validation and Analytical Figures of Merit.....	47
Analytical Figures of Merit.....	47
Method Validation – Blind Samples.....	49
Analysis of Copper Dietary Supplements.....	51
Liquid Copper Supplement Analysis	51
Additional Dietary Supplements and Tap Water Analysis	52
BPED after Ion Chromatography at Glassy Carbon	54
Conclusions.....	55
Chapter 4: BPED at the Benchtop	56
Introduction.....	56
Benchtop BPED at Gold	58
Initial BPED Parameters	58
MATLAB Functions for Data Analysis.....	59
Waveform Optimization	60
Analytical Figures of Merit.....	66
Summary of results	67

Voltammetric Studies of Copper at Glassy Carbon	68
Anodic Stripping Voltammetry.....	69
Benchtop BPED at Glassy Carbon	71
Calibration of Electrode Response and Application to a copper standard.....	72
Conclusions.....	73
Chapter 5: Summary, Impact, and Future Directions	75
Summary and Overall Impact	75
Future Directions	78
Appendix.....	80
Bibliography	117

List of Tables

Table 1: Summary of copper detection limits for EPA standard methods

Table 2: Summary of DPASV and SWASV methods at unmodified electrodes

Table 3: Summary of DPASV and SWASV using modified working electrodes

Table 4: Effect of KNO_3 concentration on copper retention time and peak asymmetry (10% peak height)

Table 5: Summary of optimization process for IC-BPED waveform

Table 6: Optimized IC-BPED values

Table 7: Analytical figures of merit for the IC-BPED method determined for Cu, Cd, and Pb

Table 8: Initial parameters for benchtop analysis of copper on Au electrodes

Table 9: Optimized analysis parameters for benchtop BPED copper detection at the Au RDE in 250 mM KNO_3 , 15.0 mM HNO_3

Table 10: Analytical figures of merit for benchtop analysis of copper at the Au RDE

Table 11: Benchtop BPED waveform parameters for glassy carbon ($d = 5$ mm) RDE in marine water

List of Figures

Figure 1: Generic Potential vs. Time Waveforms for Four Common Types of Voltammetry

Figure 2: Generic BPED Waveform

Figure 3: IC system block diagram

Figure 4: Simplified 3 electrode cell

Figure 5: Background CV Results

Figure 6: Effects of dissolved oxygen on CV response

Figure 7: Copper CV results

Figure 8: ASV Response at Au

Figure 9: Initial IC-BPED waveform results for 1.00 ppm copper solution

Figure 10: Example IC-BPED waveform optimization plot

Figure 11: Calibration series run on the IC-BPED system

Figure 12: Chromatographic reproducibility of the IC-BPED method

Figure 13: Tunability of BPED Waveform

Figure 14: Chromatogram for copper citrate supplement

Figure 15: Chromatogram of 5 ppm Cu, Cu, and Pb using glassy carbon working electrode

Figure 16: Tunability of the BPED waveform

Figure 17: Application of BPED waveform at Au working electrode to analysis of copper in 250.0 mM KNO₃, 15.0 mM HNO₃

Figure 18: S/N plot for Au benchtop BPED reduction potential optimization

Figure 19: S/N plot of reduction time optimization for benchtop BPED at Au RDE

Figure 20: Optimization of oxidation potential for benchtop BPED on Au electrode

Figure 21: Delay time optimization results for benchtop BPED on Au

Figure 22: Integration time optimization for benchtop BPED on Au electrode

Figure 23: Calibration plot for analysis of copper by the described benchtop method at an Au RDE

Figure 24: CV results for Cu in artificial seawater at the glassy carbon RDE ($d = 5$ mm)

Figure 25: ASV study for copper at glassy carbon RDE ($d = 5$ mm) in marine water

Figure 26: Batch BPED at glassy carbon with initial waveform parameters

Figure 27: Standard additions plot for copper in marine water on glassy carbon electrodes using the benchtop BPED waveform

Chapter 1: Introduction

Importance of Metals

General Remarks

The discovery of metals and metallurgy is a major turning point in human history, and is often remarked upon as a key to the development of modern society. It is thought that the earliest use of metals occurred in the shaping of copper metal, followed by silver, gold, tin, lead, and finally iron.¹⁻⁴ Arguably one of the greatest achievements of ancient humanity is the discovery of metallurgy and the mixture of copper and tin – called bronze. With the advent of metallurgy and the discovery of smelting processes, iron became the primary metal for tools and weapons for humankind, which has been refined into modern steel through the addition of metals such as nickel, titanium, vanadium, chromium, and manganese to name a few.^{3,4} Modern metal pieces are often made from lightweight, yet strong, aluminum, which can be found in pop cans, motor vehicle bodies, computers, and aircraft and spacecraft fuselages among others. Metals are not only used in their pure forms for tools, but are utilized in biology for their chemical properties as well.

Some metals, such as zinc, copper, iron, magnesium, calcium, sodium, potassium, and others are essential nutrients in both plants and animals – depending on the levels of their exposure. Other metals, such as cadmium, lead, beryllium, mercury, and arsenic are highly toxic elements to human beings. Some other metals possess both

of these characteristics depending on their oxidation state (the number of electrons which have been lost by the atom); for example, chromium is a necessary micronutrient and is essentially harmless in its +3 oxidation state (loss of 3 electrons), but is carcinogenic in its +6 oxidation state (loss of 6 electrons). When this is considered in conjunction with the fact that metals are persistent environmental contaminants, it is no real surprise to find that the monitoring and analysis of metal concentrations has been a major hub of analytical chemistry research. The need for this type of research is highlighted by the 2016 water crisis in Flint, MI, USA, or the 2015 Coal Ash Pond, NC, USA, or the 2010 Tisa River spill which occurred in Bucharest, Romania – in each of these instances a catastrophic effect of metal contamination in the aqueous environment has been observed.

Copper is thought to be the first metal worked and shaped by human kind for use as tools, vessels, jewelry and weaponry. Copper exhibits anti-microbial properties that have been used for centuries.^{5–8} In 2008, five alloys of copper were registered with the U.S. Environmental Protection Agency (US EPA) that reportedly killed 99.9% of bacteria within two hours of direct contact.⁹ Copper is used in the aquarium and aquaculture industries as an algacide and a treatment for the protozoan parasite *Cryptocaryon irritans* the cause of marine “ich” or white spot disease with a therapy range of 150 parts per billion to 200 parts per billion (ca. 2 μ M to 3 μ M).^{10–13} Copper is used as a fungicide and algacide for freshwater landscaping,¹⁴ and also arises from various natural and anthropogenic sources including pipe erosion, industrial waste

discharges, and the erosion of natural ore deposits.^{15,16} Generation of copper from these sources contaminates local waters which then permeate throughout the drinking water supply. Copper enters the marine environment by the same mechanisms as fresh waters, along with some unique contamination pathways, such as diffusion from marine paints and aquaculture waste.^{16,17,10,18,19,13} Marine coastal seawater levels of copper exhibit large concentration variances which depend on the contamination sources and the exchange rate with larger ocean waters.¹⁷ A 2006 report states that copper is commonly found at concentrations of 250 parts per trillion in ocean waters.¹⁶ Levels in the North Sea and off of the California coastline have been shown to contain copper in concentrations from 70 to 370 parts per trillion.²⁰ This research focuses on the quantification of copper in dietary supplements, fresh water and marine water matrices.

Most people obtain the proper amount of dietary copper through absorption from drinking water and consumption of foods rich in copper such as shellfish and legumes, but for certain individuals there is a need for additional dietary supplements containing copper to stave off medical conditions such as anemia and neutropenia.^{21,22} According to the US EPA and the World Health Organization (WHO), the maximum allowable copper content of drinking water is 1.3 parts per million (ca. 20 μM) and 2.0 parts per million (ca. 30 μM), respectively.^{15,23} A 2004 report from the US Department of Health and Human Services states that consumption of contaminated drinking water is the primary route to excess copper exposure in humans.²⁴ While acute copper exposure in healthy individuals is mitigated by the intestinal protein metallothionein, severe exposure results in nausea, vomiting and abdominal pain.²⁴ Long term exposure

has been shown to cause necrosis and fibrosis of the tissues in the liver and kidney, along with anemia.²⁴ Individuals with metabolism disorders of copper can exhibit symptoms of severe copper toxicity even at concentrations considered benign for healthy individuals.^{21,22,24–26} Deleterious effects are observed for excess copper exposure in marine life as well.

Copper concentrations as low as 200 to 250 parts per billion (ca. 3 to 4 μM) are capable of suppressing the immune response of vertebrate fish^{10,11} and kill invertebrates including sea anemones, crabs, and octopi¹². Copper levels of 300 parts per billion (ca. 5 μM) and higher have been shown to eradicate the bacterial colonies necessary for the oxidation of ammonia and nitrite, which can cause toxic water conditions for the fish.¹² It has also been reported that copper levels can suddenly and drastically change based on a number of mechanisms, such as changes in pH or complexation, and release from sediment or false coral structures.¹² Therefore, it is of utmost importance to determine copper concentrations at regular intervals to maintain the health of the fish, especially in aquaria where fish cannot freely migrate to less-contaminated waters.

Copper

Copper is the 29th element of the periodic table and a member of the “coinage metals” family along with silver and gold.²⁷ Copper may occupy any of four oxidation states, +1 to +4; however, most copper compounds are in either the +1 or +2 oxidation state.²⁷ In marine waters, copper is often found as $\text{CuCl}_{(\text{aq})}$ in its +1 oxidation state – in contrast to the usual +2 oxidation state observed for freshwater copper.^{28–30}

Pure copper occurs as a red-orange lustrous metal and has many uses due to its high degree of malleability, ductility and electrical conductivity. It is frequently used as the primary conductor in electrical wiring and electronic cabling. In the United States, copper is used as a sandwiching metal for dimes and quarters to reduce the required silver content for producing the currency, and one cent pieces (colloquially called pennies) are zinc slugs jacketed in a thin shell of pure copper. Copper is also frequently used for plumbing of various gases and water, though the use of copper for water pipes is being phased out in favor of polyvinylchloride analogs. Copper salts such as copper sulfate or copper citrate are packaged and sold as dietary supplements for human consumption, and are widely available for purchase in varying forms including tablets, capsules and as liquids. Additionally, copper salts are frequently found as components to plant food formulas like Miracle-Gro™.

Review of Established Methods

This review is divided into several sections. First discussed is the separation of transition metals and the subsequent evolution of those methods. This is followed by a discussion of transition metal detection methodologies which span the gamut, and include various spectroscopic, mass-spectrometric, and electrochemical techniques.

General Reviews

The reviews given in this section are of a general nature, which can familiarize the reader with additional background information and other research efforts that are taking place which are relevant but not central to the focus of this research. Richardson,

et al. have published a series of bi-annual reviews on the analysis of drinking water contaminants where metals analysis is frequently discussed.^{31–39} A general review of environmental analysis was written by Koester, *et al.* in 2003.⁴⁰ Wang and co-workers have reviewed the development of micro total analysis platforms for the purpose of metals analysis.⁴¹

US EPA Standard Methods

The US EPA has four standard methods for the analysis of trace metals in aqueous media, each relying on a different instrumental technology that has its own inherent advantages and disadvantages. Methods 200.5 and 200.7 both use inductively coupled plasma – atomic emission spectroscopy (ICP-AES),^{42,43} method 200.8 uses inductively coupled plasma – mass spectrometry (ICP-MS),⁴⁴ and method 200.9 uses graphite furnace atomic absorption spectroscopy (GF-AAS).⁴⁵ The method detection limit (MDL) or limit of detection (LOD) for copper of each method can be found in Table 1. While the US EPA standard methods all exhibit very low limits of detection, the instrumentation required to perform these analyses is costly and these methods have limitations with dissolved salts. In ICP methods, these salts can clog the ICP nebulizer apparatus and degrade the sampling cones of the mass spectrometer.^{46–48} Additionally, samples must be subjected to, “stringent preparation procedures” before analyses can be performed.^{42–45} Finally, none of the EPA standard methods have been validated for either seawater analysis or the analysis of dietary samples, and are intended only for the analysis of copper in freshwater and municipal wastewater samples.

Method	Technique	Cu MDL/LOD in parts per trillion
200.5	Axial ICP-AES	300
200.7	Radial ICP-AES	3000
200.8	ICP-MS	Scanning mode: 500
		Selected ion monitoring mode: 20
200.9	GF-AAS	700

Table 1: Summary of copper detection limits for EPA standard methods. Where a LOD was not given, the given MDL was used. Greatest sensitivity is achieved using EPA method 200.8, based on ICP-MS. In scanning mode the sensitivity is relatively lower than that reported for selected ion monitoring mode due to the wide AMU scanning window of the method. These methods are designed for the analysis of wastewater samples and soil samples, and are not designed for the analysis of marine waters.

Analytical Separation Techniques

Most metal species exist in the aqueous environment as cations – positively charged particles which are electron deficient. Therefore, it is unsurprising to find that liquid chromatography and, more specifically, ion chromatography, has been used extensively in the separation and analysis of mixtures of these analytes. Early work, such as that of Bond and Wallace in 1981 report on the separation of dithiocarbamate-metal complexes for the analysis of copper after liquid chromatography separation on a C18 column^{49,50}. This technique was further improved by Hojabri and co-workers in 1987 by adding 10 mM oxalic acid and 7.5 mM citric acid to the mobile phase and using a Dionex CS2 (an acidic, cation exchange) ion chromatography column.⁵¹ Bauer and co-workers reported on the use of post-column reaction for visible detection of metal ions after the ion chromatography separation of transition metals in 1989.⁵² In 1997 Ohta and co-workers reported the successful separation of alkali, alkaline earth,

and transition metals using a mobile phase of oxalic acid on an unmodified silica column.⁵³ The use of ion chromatography for separation and analysis of inorganic ions in water, food, and environmental samples from 1997 to 1999 was critically reviewed by Lopez-Ruiz,⁵⁴ and further reading can be found in a text by LaCourse.⁵⁵ 1998 is also when the method coined, “chelation ion chromatography” arose and was applied to the separation of lead, cadmium, copper and other transition metals - such as in the work reported by Lu, *et al.*⁵⁶

In 2000, Osipova, *et al.* separated copper from other transition metals on a C18 column using a method similar to Bond and Wallace⁵⁷. That same year, a report by Santoyo, *et al.* details the separation and analysis of transition metals using ion chromatography on a mixed functionalized quaternary ammonium and sulfonic acid resin using oxalic acid and lithium hydroxide as the mobile phase with post-column reaction detection,⁵⁸ and is similar to several technical notes released by Dionex, Inc.^{59–}

⁶¹ These separations rely on the constant dynamic equilibrium between the column stationary phase, and the Lewis base properties of organic acids such as oxalic or citric acid to effectively accomplish the analytical separation. A 2013 report by McGillicuddy and co-workers outlines a separation method for transition metals and alkaline earth metals using only HNO₃ and KNO₃ as the mobile phase using a hydroxyethyliminodiacetic acid functionalized monolithic silica column.⁶² While this column exhibited good performance, the manufacturing of these types of stationary phases are still inconsistent, and so they are ill-suited to critical, routine analytical work.

Capillary electrophoresis has also been used to separate mixtures of metal ions. Chen, *et al.* reported on the various effects of chemical modification to capillary columns for the separation of transition metal ions using capillary electrophoresis in 1995.⁶³ Another report from 1995 shows that transition metals were effectively separated using a mobile phase of imidazole, lactic acid, and 18-crown-6 ether on a fused silica capillary column; the authors also reviewed many prominent publications on the development of transition metal separations by capillary electrophoresis.⁶⁴ In 1996 Wen and Cassidy reported that by using Bi-potential Pulsed Electrochemical Detection (BPED) after capillary electrophoresis they were able to separate several transition metal ions using a polyimide-fused capillary column and a mobile phase of α -hydroxyisobutyric acid and creatine.⁶⁵

The detection of metal species in aqueous samples has been a long-standing pursuit in analytical chemistry. Unsurprisingly, a significant amount of research has been devoted to the improvement of the various reported methods over the decades.

Detection Methods

The detection of aqueous metal ions dates back decades, and as a result the literature is replete with reports on the analysis and determination of metal concentrations. These papers fall into three broad groups: optical methods, electrochemical methods, and mass spectrometric methods.

Optical Detection Methods

Optical detection methods are often seen in the literature, and function on either the measurement of photons absorbed by a metal-chromophore complex, measurement of photons emitted by a metal-fluorophore complex, or measurement of photons absorbed or emitted by excited atomic nuclei (such as in US EPA methods 200.5, 200.7, and 200.9). Brezea-Malcom and co-workers have summarized recent attempts at using micro-organisms combined with optical methodologies to determine metal concentrations⁶⁶, and Pal and others have discussed the use of fluorescence sensors for this application as well.⁶⁷

Most optical methods often reach limits of detection as low as 200 parts per trillion, such as that reported in 1998 by Lu, *et al.*⁵⁶ The major disadvantage to these methods is that, with exception to atomic absorption and atomic emission spectroscopy, detection is only possible through the addition of a chromophore or fluorophore chelating dye. While this can be done prior to the separation, the separation would then need to be tailored to the metal-chelate complex generated to isolate and quantify each metal in solution. If added after the separation, additional equipment such as pumps are needed to deliver the chelating dye and additional tubing is also required to allow time for the reaction to occur. In the case of atomic absorption and atomic emission spectroscopy, both methods require the use of nebulization equipment that is highly sensitive to sample viscosity, salt concentration, and particulate matter, and thus require very stringent preparation and filtration protocols. Additionally, in the case of inductively coupled plasma instruments, the instrument sampling cones are degraded

from samples of high acid and salt concentration – often used to digest organic material and ensure the metal ions are free in solution for these techniques.

Mass Spectrometric Methods

Inductively coupled plasma – mass spectrometry is the most common technique employed for the direct analysis of metals by mass spectrometry, and thus dominates the literature for this analysis. The coupling of liquid chromatography to inductively coupled plasma-mass spectrometry has been covered in a thorough review by Montes-Bayon and co-workers.⁶⁸ These techniques are very sensitive, often achieving detection limits ca. 20 parts per trillion, as stated for the US EPA method 200.8. As stated above, the instrumentation is highly sensitive to the salt and acid used to eliminate matrix effects, and copper detection is often interfered with by other ions with a mass to charge ratio of 63 atomic mass units, such as PO_2^+ , ArNa^+ , and TiO .^{44,46–48} Other work includes the use of Electrospray ionization for the indirect detection of metals through metal-ligand complexes, however, these methods rely on complex tandem mass spectrometry methods that require highly skilled operators, and these instruments come at a very high monetary cost.^{69,70}

Electrochemical Detection Methods

Electrochemical detection circumvents most of the issues observed in optical and mass spectrometric detection methods. These devices can be placed in-line with a separation technique, rely on an intrinsic property of the analyte, and do not experience the severe interferences from salt or acid described above. Therefore, electrochemical

detection is the focus of this research. Commonly reported electrochemical methods can be sub-divided into four groups: potentiometric, conductivity, voltammetric, and amperometric.

Potentiometric methods are those for which analyte concentration is measured as a function of electrical open-circuit potential, i.e. no electrical current flow. Conductivity measurements correlate electrical permittivity of a solution with analyte concentration. Voltammetric analyses measure the current passed through a working electrode surface, based on changes in potential at this working electrode. In these methods, the current is proportional to the concentration of the analyte, while the potential gives a means of identifying the reaction taking place. Amperometric methods measure the change in current at a working electrode over time. Direct current amperometry is the measurement of current while potential is held constant. The measurement of current while potential is changed using a multi-step potential vs. time waveform is termed Pulsed Electrochemical Detection (PED). Like in voltammetry, the measured current is proportional to the analyte concentration.

An in-depth discussion of electrochemical methods for metal speciation is given by Mota and co-workers, with a focus on the kinetics of detection.⁷¹ Suginta, *et al.* discuss the modification of electrode surfaces with the bio-polymers chitin and chitosan in their review,⁷² while Gumpu and co-workers discuss the modification of electrodes with nanoparticles for drinking water analyses.⁷³ Finally, a review of pulsed electrochemical detection after liquid chromatography and capillary electrophoresis has been published by Fedorowski and LaCourse⁷⁴.

Potentiometric Methods

In potentiometric analysis, no current is permitted to flow in through electrochemical cell.^{75,76} The potential difference across a membrane of glass or other material is measured and compared to a reference electrode potential, and this potential is correlated to the analyte concentration. The most common potentiometric method for analysis of aqueous copper is the copper Ion Selective Electrode (ISE);⁷⁵ however, it is well documented that chloride ions interfere with this analysis.⁷⁷ Furthermore, the copper ISE also experiences interferences from Hg^{2+} , Cd^{2+} and Ag^{1+} .⁷⁵ This reduction in selectivity makes the use of a copper ISE unattractive for samples that may possibly contain multiple metal species.

The application of the copper ISE for seawater analysis was thought to be highly problematic even as recently as 2003 due to the very significant interference of chloride.¹⁷ Current research into improving the copper ISE has focused on reducing this interference through modification of the electrode surface. A 2012 report by Qin and co-workers highlights the successes of this research and show a subnanomolar detection limit of copper in 0.5 M NaCl using an electrode with a PVC-modified membrane.⁷⁸ While the authors overcame the interference of chloride, they did not test their sensor's response to the other known interfering cations for copper ISEs such as Hg^{2+} , Cd^{2+} and Ag^{1+} . Therefore, the copper ISE remains an unproven and unattractive option for this research.

Conductivity Measurements

Conductivity measurements of a solution have traditionally been performed by passing the solution of interest between two platinum metal electrodes held at a constant voltage.⁷⁹ The electrical permittivity is determined by measuring the current that passes through the two electrodes. Since permittivity is a bulk property of a solution, these measurements are inherently non-specific. In 2014, Bajwa and co-workers reported development of a conductivity sensor based on a molecularly imprinted polymer that gave signal responses four times larger for Cu^{2+} , versus other ions such as Ni^{2+} , Co^{2+} , Zn^{2+} , or Na^{+} ; they report an achieved limit of detection of 1 part per million.⁸⁰ However, they did not address the fact that conductivity is not inherently specific to the analyte species. While they have reduced interferences, they have not eliminated them.

Voltammetric Methods

Voltammetric methods are those which measure current in response to a changing potential vs. time waveform at a working electrode surface.^{75,81-84} Four of the major voltammetric techniques are linear sweep voltammetry (LSV), cyclic voltammetry (CV), differential pulse voltammetry (DPV), and square-wave voltammetry (SWV). The measured current in these techniques is proportional to the analyte concentration, and the potential at which the current is observed can be used to identify the analyte species.

Briefly, LSV uses a linear potential sweep with time and has been used to give qualitative information about the reductive or oxidative species in solution. The potential vs. time waveform for a typical LSV experiment can be found in Figure 1-A. The potential vs. time waveform for a typical CV can be found in Figure 1-B. CV is a common technique for obtaining qualitative information about redox couples in solution,^{55,81–83} though quantitative applications are sometimes reported.

For both of these waveforms, the observed current is directly proportional to the square-root of the rate at which the potential is changed with time (scan rate). This is an important factor in selecting parameters for these types of experiments. However, these techniques are rarely applied to the analysis of metals because of electrode hysteresis and high non-Faradaic (capacitance) currents at high scan rates.

DPV and SWV are both examples of pulsed voltammetry and, as such, have some similarities despite the very different time scales for their potential-time waveforms. In both cases, current is sampled for a short amount of time both before the potential pulse, and again immediately before the end of the pulse time. Non-Faradaic current is eliminated by delaying current sampling, and by calculating the difference of the current at the two points an enhanced current is obtained.^{81,82,85–87}

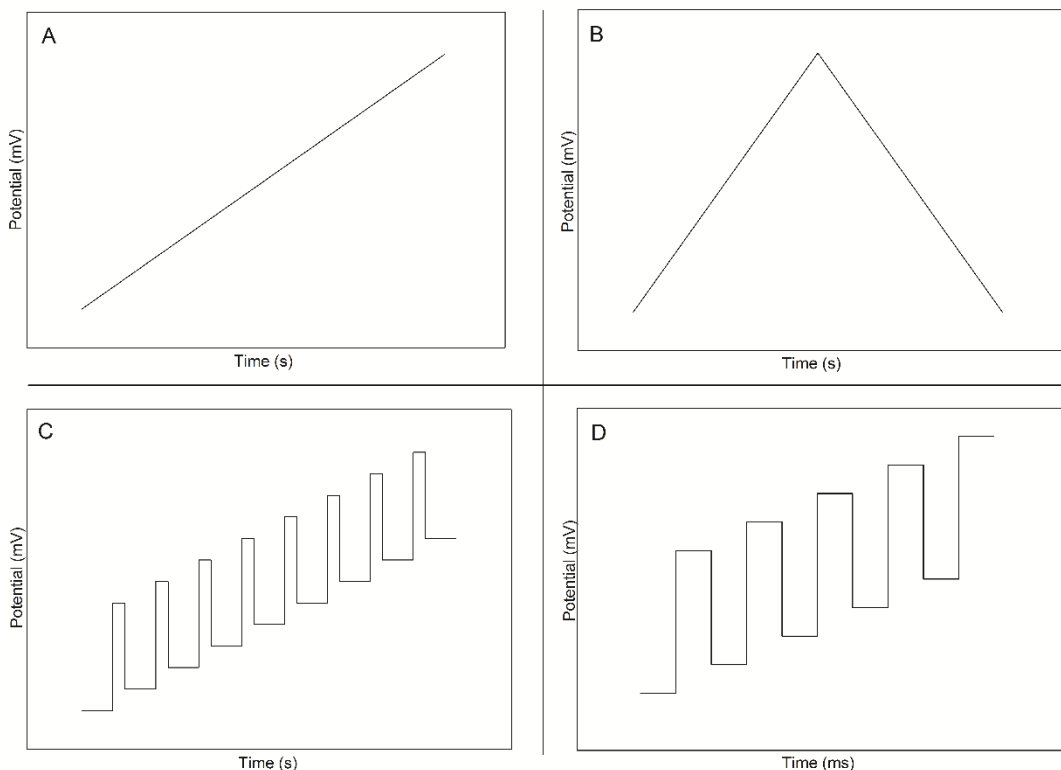


Figure 1: Generic potential vs. time waveforms for the four common types of voltammetry – LSV (A), CV (B), DPV (C) and SWV (D). Note that the time axis for SWV (D) is in ms, not s. Most commonly used for quantitative analyses of metal ions are DPV (C) and SWV (D). This is especially true for SWV, because the high frequencies used allow for very rapid waveform cycling. CV (B) is sometimes used for quantitative applications, but is more commonly used as a qualitative technique.

SWV has been used in seawater analysis following an adsorptive pre-concentration step, and electrolytic reduction of copper at the working electrode. Liu, *et al.* reported in 1999 a limit of detection of 25 parts per trillion (ppt) using a cysteine self-assembled monolayer modified gold electrode; this method was tested on the standard seawater reference solution CASS-3.⁸⁸ DPV has been utilized by Zhihua, *et al.* where a Cu^{2+} molecularly imprinted polymer working electrode produce a limit of detection of 100 ppt, and the method was successfully applied to analysis of river water;

this success is offset by the complexities of generating the molecularly imprinted polymer prior and then covering the entire sensor surface with PVC.⁸⁹

In most cases, the standard voltammetry methods are not used for detection and quantification of metals. Instead, metals are electrolytically pre-concentrated on the surface of the working electrode by holding the working electrode at a constant potential prior to application of LSV, DPV or SWV potential-time waveforms. Upon waveform application, the metal is stripped off of the electrode surface, producing an enhanced current response.^{82,84} These techniques are termed anodic stripping experiments when the working electrode functions as an electron sink during stripping step (analyte is oxidized), and cathodic stripping experiments when the working electrode acts as an electron source during the stripping (analyte is reduced). The most common methods for determining copper concentrations in water are Linear Sweep Anodic Stripping Voltammetry (LSASV or LSSV), Square-Wave Anodic Stripping Voltammetry (SWASV), and Differential Pulse Anodic Stripping Voltammetry (DPASV).

First reported in 1972 by T. M. Florence, copper was able to be detected in seawater samples using an *in situ* mercury plated electrode.⁹⁰ Due to environmental and toxicological concerns, the majority of research is now focused on mercury-free methods.^{91,92}

In 2014, Dai and co-workers reported copper detection using an amine functionalized mesoporous silica modified (NH₂-MCM-41) glassy carbon electrode and LSASV, with a 0.9 ppt detection limit and a linear response region of 5 to 1000 ppt

after a 4 minute preconcentration at -600 mV.⁹³ They were able to apply their method to the analysis of copper in tap water and lake water, and found the results were comparable to those obtained using inductively coupled plasma – optical emission spectroscopy.

Due to the enhancement of current observed in the two pulsed voltammetry techniques, DPASV and SWASV, these methods are more abundant in the literature. Reports using unmodified electrodes, along with limits of detection (LOD) and electrode materials can be found in Table 2.

Technique	Electrode Material	Cu ²⁺ LOD*	Reference
SWASV	Au Rotating Disk Electrode	10 ppt	94
DPASV	Vibrating 25 μ m Au wire electrode	400 ppt	95
	Vibrating 10 μ m Au wire electrode	200 ppt	
SWASV	Vibrating 25 μ m Au wire electrode	400 ppt	
	Vibrating 10 μ m Au wire electrode	300 ppt	
SWASV	Boron-doped Diamond	100 ppb	92
SWASV	Photolithographically assembled Micro-array of Pt electrodes	6 ppb	96

Table 2: Summary of DPASV and SWASV methods at unmodified electrodes. Noble metal and carbon based electrode materials are used for these analyses, and give very low LODs for copper. *Where LOD was not given, the lower LOQ was used.

To further improve signal to noise levels, researchers have taken to chemically modifying the surface of their working electrodes to allow either an adsorptive pre-concentration step, or increase selectivity for the metal species of interest. Table 3 is a summary of the methods employing either DPASV or SWASV at modified electrode surfaces.

Substrate	Electrode Modification	Pre-concentration Method	LOD *	Reference
Au	Meso-2,3-dimercaptosuccinic acid Self-assembled Monolayers (SAM)	15 minute open circuit adsorption	1 ppb	97
Carbon Paste	Ionic-liquid modified mesoporous silica	6 minute reduction at - 1.2 V	3 ppb	98
Glassy Carbon	Poly(2-amino-4-thiazole acetic acid)/Multi-wall Carbon Nanotubes	5 minute open circuit adsorption	30 ppt	99
Glassy Carbon	L-Cystene SAM on Au nanoparticles on single-wall carbon nanotubes	3 minute open circuit adsorption	1 ppt	100
Pencil Graphite	Polystyrene sulfonate carbon nanopowders	5 minute open circuit adsorption	100 ppt	101
Carbon Paste	Iodoquinol modified multiwall carbon nanotubes	15 minute reduction at - 0.5 V	300 ppt	102
Glassy Carbon	N-doped graphene	5 minute reduction at - 1.1 V	60 ppt	103
Carbon Paste	Aminopropyl grafted silica	10 minute open circuit adsorption	60 ppt	104
Carbon Paste	<i>In situ</i> deposited Sb film	5 minute reduction at - 1.2 V	2 ppb	105
Carbon Paste	Pyrochatechol Violet	1 minute open circuit adsorption	1 ppb	
Glassy Carbon	SnO ₂ /Graphene	2 minute reduction at - 1.0 V	10 ppt	106
Carbon Paste	N,N' – bis(3-(2-thenyldienimino)propyl) piperazine coated silica nanoparticles	1 minute reduction at - 1.1 V	0.1 ppt	107
Glassy Carbon	1,4-diaminobenzene hydrogel	10 minute reduction at - 0.8 V	0.5 ppt	108
H _x TiS ₂ Nanosheets	p-aminothiophenol on Au nanoparticles	2 minute reduction at - 1.0 V	6 ppt	109
Carbon Paste	Xanthate functionalized carbon nanotubes + graphene	3 minute reduction at - 0.45 V	600 ppt	110

Table 3: Summary of DPASV and SWASV using modified working electrodes. Electrode surface modification combined with various types of pre-concentration methods, both with and without applying a reduction potential, have been extensively studied in attempts to lower method LOD and improve overall signal to noise levels in analyses. These types of studies are more commonly done with carbon-based electrode materials than inorganic materials (such as gold). *Where a LOD was not given in the publication, the lower LOQ was used.

Amperometric Methods

Amperometric detection methods rely on the measurement of current at a fixed potential, or pulsed potential sequence. A 2013 report by Mari and co-workers describes a Bi-potential Pulsed Electrochemical Detection (BPED) analysis of copper

in fresh water using a microfabricated Au electrode in a stopped-flow thin layer cell.¹¹¹ Their sensor relies on complete electrolysis of all metals in the 2 – 4 μL volume using a 1 minute 0 mV deposition step followed by a pulse to 850 mV for 640 seconds to obtain the current. Using their device, the authors observed a linear signal range for copper from 50 ppb to 7.5 ppm; no limit of detection was given. This device, while giving excellent performance, relies on complex microfabrication techniques that would make it prohibitive to be employed by un-trained and un-specialized analysts in the field.

Bi-potential Pulsed Electrochemical Detection (BPED)

BPED is a form of pulsed electrochemical detection, where the potential is stepped between a reducing potential and an oxidizing potential. A generic waveform with labeled pertinent parameters is shown in Figure 2.

During the reduction step, metals with a reduction potential greater than E_{red} will be reduced onto the surface of the working electrode. The potential is then increased or “stepped” to E_{ox} where reduced metals are anodically stripped from the surface: the current (I) vs. time (t) is then plotted. This plot is termed a chrono-amperogram.

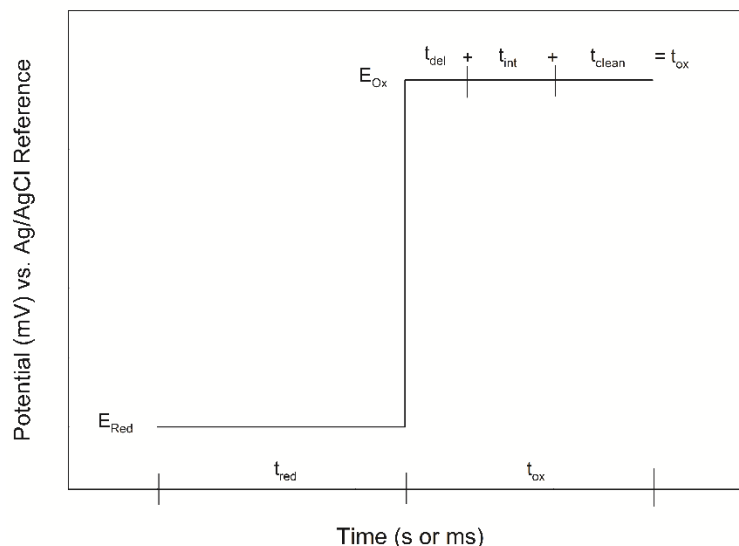


Figure 2: Generic BPED waveform. The first step is reduction which takes place at E_{red} for a length of time (t_{red}), causing reductive pre-concentration of metal at the electrode surface. This is followed by oxidative stripping at E_{ox} for t_{ox} , where t_{ox} is the sum of $t_{\text{del}} + t_{\text{int}} + t_{\text{clean}}$. This oxidation signal is enhanced by the previously executed reduction step. The purpose of t_{del} is to further increase the signal to noise level of the technique by allowing non-Faradaic (capacitive charging) current to fall off prior to quantification by integration under the resulting I vs. t plot – a chronoamperogram. t_{clean} serves to ensure that the electrode surface is completely free of reduced metal before starting the next cycle of the BPED waveform.

The oxidation time is broken into three segments for processing of data: t_{del} which allows for the dissipation of non-Faradaic capacitance, t_{int} over which the I vs. t curve is integrated to produce the charge passed through the working electrode, and t_{clean} which is used to ensure that the electrode surface has no metal on the surface prior to repeating the waveform. This stripping procedure produces an enhanced current due to the pre-concentration and also due to the potential step, which is analogous to an infinitely fast potential scan in voltammetric techniques.

Research Objectives

The purpose of this research was to explore the use of BPED for analysis of copper in the freshwater environment and in dietary samples. Since the BPED waveform does not have inherent selectivity, the technique was applied after IC separation of copper with lead and cadmium – other metals which may be of interest to environmental and toxicological researchers. This is the focus of chapter 3, and shows that BPED can be successfully applied to determine copper in samples of dietary supplements and samples with very little chloride present at gold electrodes, but analysis at glassy carbon electrodes is less successful.

From results observed during the development of the chromatography BPED method, it was hypothesized that a BPED method could be developed for the analysis of copper at the benchtop without prior separation using the reduction potential as the selective element in the analysis. This necessitated developing a method for determining optimal analysis parameters, and computer techniques were used extensively in assistance with data analysis to expedite the optimization process. This optimization process at gold and the resulting BPED method are presented in chapter 4. Also presented in chapter 4 is the results from using a glassy carbon working electrode to create a benchtop BPED method for marine water samples.

Chapter 2: Experimental

Instrumentation

IC-BPED Instrumentation

Ion chromatography – BPED (IC-BPED) experiments were carried out using a Dionex, Inc. (Sunnyvale, CA) GP-50 model pump with an Alltech in-line degassing unit (Grace Industries, Columbia, MD). Separations were performed on a Dionex CS12A (4 mm x 250 mm) with a Dionex CG5A (4 mm x 50 mm) guard column and a mobile phase of 15.9 mM HNO₃/425.0 mM KNO₃ flowing at a rate of 1.00 mL·min⁻¹. Samples were introduced using a Rheodyne model 9010 manual injection valve (Rohnert Park, CA) fitted with a 50 µL injection loop (25 cm length of 0.020 inch ID PEEK tubing). Analyte detection was performed using a Dionex, Inc. ED-40 electrochemical detection module interfaced to a Dionex electrochemical thin-layer flow cell fitted with a 1 mm diameter gold working electrode and Ag/AgCl/pH combination reference electrode was used for detection and quantification. All liquid connections were made with PEEK tubing and fittings to prevent degradation by the nitric acid and all experiments were conducted at ambient temperature (between 22 °C and 25 °C). Once installed on the IC-BPED system, mobile phases were kept under N_{2(g)} at 8 psi. Both the GP-50 and ED-40 were controlled using DX-Lan and Dionex PeakNet software (version 5.21) installed on a Microsoft Windows™ 98 SE virtual pc. The virtual pc was hosted on a physical computer running Microsoft Windows™ XP

and created using the Microsoft Virtual PC 2007 software suite. A block diagram of the liquid chromatography system is shown in Figure 3.

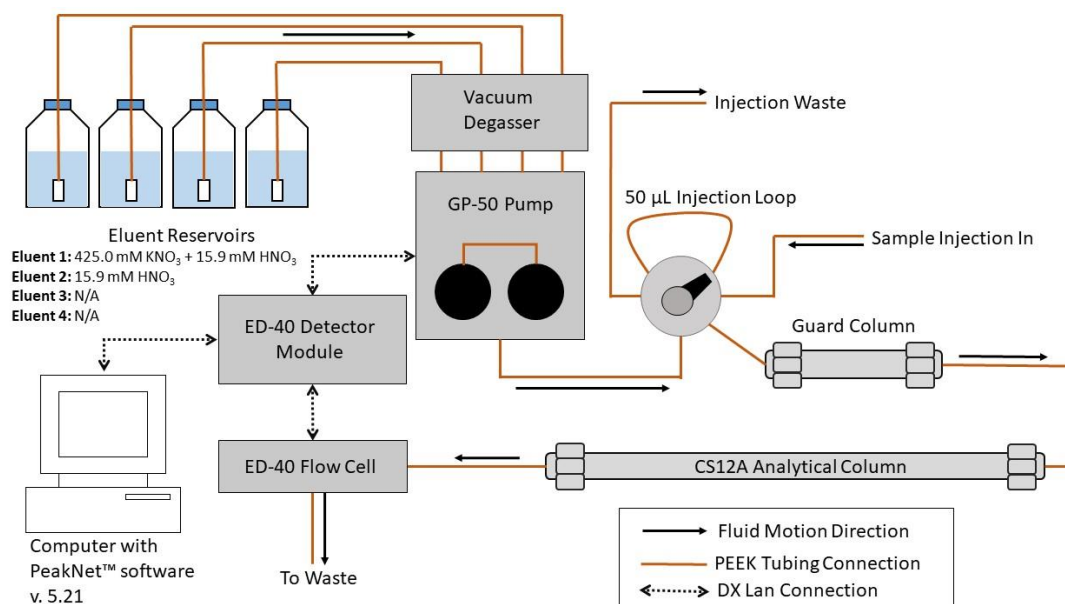


Figure 3: IC system block diagram. Only two of the 4 possible eluent channels of the GP-40 pump were utilized throughout this research. The primary mobile phase was 425 mM KNO₃, 15.9 mM HNO₃ and a second eluent bottle with 15.9 mM HNO₃ was installed on the system to serve as a diluent as needed. Injection loop volume = 50 µL (25 cm of 0.020 inch I.D. PEEK tubing). Unless otherwise stated, the system was run using only the primary mobile phase (100%). Both the ED-40 and GP-50 were controlled using a PC running Dionex PeakNet™ version 5.21 over DX-LAN connections.

Electrochemical Instrumentation

Voltammetric and amperometric studies were performed using a Pine Instruments (Durham, NC) WaveNow™ portable USB potentiostat interfaced to a Windows XP PC and controlled using Pine Instruments AfterMath™ software. A glass conical cell was used for all electrochemical experiments in a standard 3 electrode configuration; an Ag/AgCl reference electrode (BAS, Inc., West Lafayette, IN) and Pt wire counter electrode (BAS, Inc.) was used. In samples containing low concentrations

of chloride anion, a gold working electrode was used ($d = 1$ mm, Pine Instruments) and in samples containing high chloride content a glassy carbon working electrode ($d = 5$ mm, Pine Instruments) was used. The cell configuration is depicted in Figure 4. Rotation of the working electrode was provided by using an analytical rotor (Pine Instruments, Model AFM-SRX) and controlled using a model MSRX speed controller (Pine Instruments). Rotation was set to 900 rpm in all experiments, unless otherwise specified.

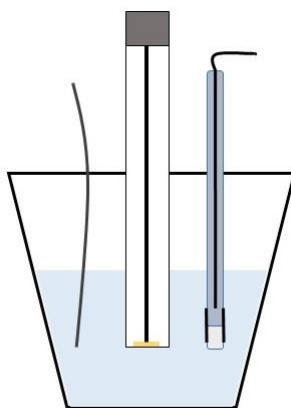


Figure 4: Simplified 3 electrode cell. Left is Pt wire aux, Center is rotating disk working electrode, right is Ag/AgCl Reference Electrode. A gold working electrode was used for all fresh water analysis, and glassy carbon was used for all marine water analysis.

Spectrophotometric Instrumentation

A model V-560 Spectrophotometer (Jasco, Inc., Easton, MD) was used in this research to independently confirm the concentration of the SENSEable supplements copper solution using disposable 3 mL cuvettes.

Reagents and Solutions

Chemicals and Reagents

Ultrapure NERL grade water was obtained from Fisher Scientific (Pittsburgh, PA). Artificial seawater was generously donated by The National Aquarium (Baltimore, MD). Certified trace metals free artificial seawater was purchased from VWR International (Radnor, PA). Metal standards were purchased as commercially available spectroscopy standards with certified concentrations of 1000 ppm for copper (Fisher Scientific, Pittsburgh, PA), 1000 ppm for lead (Ricca Chemical, Arlington, TX), and 10,000 ppm for cadmium (Fisher Scientific). Potassium nitrate (KNO_3 , >99% pure, VWR International, Radnor, PA) and nitric acid (HNO_3 , >69.3% pure, Fisher Scientific) used in making supporting electrolyte solutions and chromatographic mobile phase solutions were used without further purification. Copper(II)sulfate pentahydrate ($\text{CuSO}_4 \cdot 5 \text{H}_2\text{O}$, >98% pure, Fisher scientific) used to generate copper calibration solutions was purchased from Fisher scientific, and used without further purification. Dietary supplements included SENSEable supplements copper sulfate liquid (Smell Taste Technology, Liberty Lake, WA), copper citrate capsules (Pure Encapsulations, Sudbury, MA) and copper glycinate chelated tablets (Solgar, Inc., Leonia, NJ); all of which were purchased from Amazon.com.

Preparation of Chromatographic Mobile Phases and Solutions

All standards, solutions and dilutions were prepared using volumetric glassware unless otherwise specified. Chromatographic mobile phase was generated in 2.00 L

batches using volumetric glassware and sonicated under vacuum for approximately three (3) hours prior to installation on the IC system. This mobile phase was generated by dissolving 85.943 g of $\text{KNO}_{3(s)}$ in the minimum amount of NERL water, then adding 2.00 mL of concentrated $\text{HNO}_{3(l)}$ using a glass pipet before diluting the entire volume to 2.00 L. Additional batches of mobile phase were used to generate standard samples, and these were produced in 1.00 L batches by an analogous protocol. An analogous protocol was also used when generating $\text{KNO}_3/\text{HNO}_{3(aq)}$ supporting electrolyte solutions for voltammetric and amperometric studies.

Calibration of the JASCO V-560 spectrophotometer response was done by generating a 10.01 parts per thousand (ppth) solution of copper sulfate pentahydrate ($\text{CuSO}_4 \cdot 5 \text{H}_2\text{O}$, >98% pure, Fisher scientific) by dissolving 393.4 mg of the pentahydrate salt in 10.00 mL of NERL water using volumetric glassware. A 1:10 dilution of this stock, and 1:100 dilution of this stock was also prepared using volumetric glassware.

Calibration of the IC-BPED response was done by diluting purchased metal standards to 100.0 ppm to make a working calibration stock. Stocks were made for each analyte metal individually, and for the mixture of copper, cadmium and lead in the mobile phase. For both this mixed metal stock and the individual metal solutions, a series of calibrant solutions in the mobile phase ranging from 1.00×10^{-3} ppm to 1.00 ppm were made by dilution using volumetric glassware.

Solutions of the dietary supplement samples were prepared by first placing a single tablet or capsule, in the case of the Solgar brand supplement and Pure

Encapsulations brand supplement respectively, in a glass vial and pipetting 20.00 mL of mobile phase into the vial. The tablet or capsule was allowed to dissolve over a period of 24 hours with periodic vigorous shaking. After this period, insoluble matter was permitted to settle to the bottom of the vial, and a 1:200 dilution of the resulting supernatant solution was made using the mobile phase as diluent with volumetric glassware prior to analysis by IC-BPED. For the SENSEable supplements brand copper sulfate supplement, a 1:100 dilution of the supplement was prepared using the mobile phase as the diluent, and this solution was then diluted again 1:100 with mobile phase as diluent for a total dilution factor of 1:10,000.

Additional calibration solutions of copper for voltammetric and amperometric studies were generated as needed by diluting an aliquot of the 1000 ppm commercial standard to the volume required to achieve the desired stock concentration using the electrochemical supporting electrolyte solution using volumetric glassware.

Procedures and Sample Handling

Polishing of Electrodes

For each electrode material a separate micro fiber polishing pad (Buehler, Inc., Lake Bluff, IL) was impregnated on one side with coarse alumina powder (0.3 μm , Buehler, Inc.) for coarse polishing, and Gamma alumina micropolish no 3 (0.05 μm , Buehler, Inc.) on the other side. Initial addition of polishing compound was done by creating an aqueous slurry of alumina on the polishing pad surface and spreading this

with a gloved finger. The electrode material and compound grade was marked on the pad with indelible ink.

During electrode polishing, a small quantity of water was added to the surface of the horizontal pad to act as lubricant. Electrodes were then polished by holding the electrode perpendicular to the polishing pad and moving the electrode in a figure eight pattern for 100 strokes. The electrode was then rinsed with copious amounts of ultrapure water and sonicated in methanol for 10 minutes to remove all traces of alumina from the surface before moving to the next grit level.

Daily polishing of glassy carbon working electrodes was done using the fine alumina powder, but gold electrodes were polished only after a week of disuse, or when electrode performance was observed to degrade due to electrode fouling. Coarse alumina polishing was only performed when electrodes presented with a high level of fouling or if, upon magnification, there were scratches observed on the surface. In cases of extreme damage to the electrode surface, the electrodes were first polished on 600 grit emery paper followed by the coarse alumina, and then the fine alumina until a mirror shine was achieved and no mechanical defects were observed at 10 x magnification under a jeweler's loupe.

Cyclic Voltammetry and Anodic Stripping Voltammetry

In CV or ASV experiments, 20.00 mL of the supporting electrolyte solution was transferred to the conical glass cell *via* volumetric pipet. Supporting electrolyte solutions were prepared as described in preparation of chromatographic mobile phases

and other solutions. The reference and counter electrodes were inserted into the solution, followed by placement of the RDE working electrode. The RDE was placed in the cell so that it made contact with the solution surface and rotation was turned on. The voltammetric response was recorded for this solution three times prior to addition of any copper.

For cyclic voltammetry experiments, the working electrode potential was held for 5 seconds at the initial potential vs. Ag/AgCl then swept to the vertex potential at a rate of 50 mV/s unless otherwise specified. Upon reaching the vertex potential, the potential sweep was reversed and potential was swept back to the initial potential at the same rate. After collection of the background voltammogram, the solution was spiked with copper standard and the voltammogram was re-collected in triplicate. This was done with multiple spikes of copper standard to observe the response as the concentration of copper in the cell increased.

In ASV experiments, the working electrode potential was held at the observed reduction potential for copper in the supporting electrolyte from the CV experiments for a time period from 0 to 60 seconds. Potential was then swept at a rate of 50 or 100 mV/s to 200 mV more positive than the observed oxidation potential for copper from the CV experiments. After collection of background voltammograms, the solution was spiked with copper standard and the process was repeated.

Initial BPED Waveform Development

Initial development of BPED waveforms was performed in the 3 electrode cell described above using the RDE appropriate for the matrix. Initial values for E_{red} and E_{ox} were determined through observation of copper reduction and oxidation potentials in the supporting electrolyte solution or sample matrix from collected CV data. Initial values for t_{red} were determined by performing ASV experiments with increasing deposition times on solutions of 1 ppm copper. The initial value of t_{ox} was arbitrarily set to the same as t_{red} to ensure a clean electrode surface between pulse cycles in stand-alone applications. For HPLC-BPED, the initial value of t_{ox} was set to 1 second, with $t_{\text{del}} = 0$ ms, $t_{\text{int}} = 10$ ms, and $t_{\text{clean}} = 990$ ms.

Optimization of BPED for IC Analysis

In the IC-BPED system, optimization was done on-line through modification of BPED waveform parameters. Optimization was carried out by observing the effect of a changing parameter on the signal of a 1 ppm copper solution injection. The parameter value which gave the greatest signal was chosen as the optimum value.

Optimization of BPED Waveforms for Benchtop Applications

Optimization of all BPED waveforms for benchtop methods were performed in the 3 electrode cell described previously. During optimization, all parameters except the one being optimized were held constant. Once a parameter was optimized, the optimized value was used in any further experiments. System noise was estimated by performing a chronoamperometric experiment using identical parameters to the test

experiment, but a t_{red} of 10 milliseconds. This reduction time is short enough to allow the formation of the electrochemical double layer with only negligible reduction of metal species.

Parameters were optimized in the following order: E_{red} , t_{red} , E_{ox} , t_{del} , t_{int} , t_{clean} . In all cases, time parameters were chosen that would result in greatest cycling frequency possible while maintaining method sensitivity. Signal to noise ratios were calculated by subtracting the non-Faradaic currents from the observed signal, and then dividing the average signal by the standard deviation of the corresponding blank ($t_{\text{red}} = 10$ ms) experiments.

Data Analysis and Handling

Data analysis was extensively performed using the Mathworks MATLAB™ computing software (Release 2012a – 2015b). To expedite the analysis and optimization of waveform parameters, several function and script files were written for the MATLAB™ computing environment. The code for these functions, and a discussion of programming in the MATLAB™ environment is given in Appendix 1.

Sample Storage Recommendations

Artificial seawater samples were stored at room temperature, away from light in acid washed glass bottles to minimize the risk of metal leeching by the sample container. It was observed during this research that samples and standards made in artificial seawater and exposed to light for an extended period developed algal or mold growth. When this was observed, solutions with the algal/mold growth were discarded

and fresh solutions were made. This growth did not occur in samples stored away from light or bottles wrapped with aluminum foil. Freshwater samples were stored in acid washed glass bottles at room temperature. Samples taken from display and treatment tanks of aquariums were stored in plastic vials away from light at ambient temperature.

Acid washing of containers was performed by rinsing with a 1 M solution of Nitric Acid followed by copious rinses with ultrapure water. Containers thus prepared were rinsed with the solution being contained, if needed immediately. If they were not, then the container was air dried with the cap placed loosely on top to prevent contamination. Dietary supplements were stored at ambient temperature following manufacturer recommendations. Tap water was collected, after allowing water to flow for 15 minutes to allow for equilibration of metal ion concentrations, directly in an acid washed glass bottle.

Safety Considerations

Experimental waste should be disposed of responsibly, per the hazardous waste disposal guidelines of the institution at which the work is being performed. Utmost care should be taken when handling nitric acid as it is highly corrosive. The use of nitrile or latex gloves, safety glasses or goggles, and a laboratory coat with sleeves that cover the entire arm are highly recommended. Additionally, dispensing concentrated acid should be performed in a fume hood with the sash at the proper operating height to prevent inhalation of acid vapors. When handling copper or other metal salts and solutions, individuals should wear gloves, safety glasses and a laboratory coat.

Additionally, good chemical hygiene should be practiced always; this including washing of hands before, and after experiments, before bathroom breaks, or prior to consumption of food or drink.

Chapter 3: IC-BPED analysis of Copper, Lead and Cadmium

Introduction

Electrochemical analysis can be done at mercury based electrodes such as the hanging mercury drop or mercury thin film electrode. Due to environmental concerns though, use of these electrodes has fallen off in recent years. Noble metal electrodes are classically used in electroanalytical techniques of other species due to their chemical inertness and wide window of usable potentials, often ca. 2 V. In copper analysis, common noble metal electrodes include gold, silver, and platinum metals. In this research, gold was chosen as a suitable working electrode because of its well defined behavior in performing copper stripping analyses.^{94,112}

The primary issue with BPED analysis of metals is the lack of specificity in the detection waveform. In theory, any metal with a standard reduction potential (E°) more positive than the applied E_{red} will be reduced onto the electrode surface. Likewise, any metal that is reduced on the surface will be stripped during the oxidation half-cycle of the waveform, so long as the oxidation potential (E_{ox}) is sufficiently positive with respect to the standard reduction potential of the species. This implies that metal species should be separated prior to detection, and this separation can be achieved readily using cation exchange chromatography. This chapter focuses on the development of BPED after cation exchange separation of metals to achieve quantitative detection of copper, lead, and cadmium.

Waveform development began with initial qualitative study of the reduction-oxidation (redox) system using voltammetric methods (CV and ASV). From these voltammetric experiments an initial BPED waveform was created, tested, and optimized. This method was successfully applied to the analysis of copper in dietary supplements.

Voltammetric Studies of Copper Redox system

Development of any BPED waveform must necessarily begin with analysis of the redox system in question. This is because the reduction and oxidation potential will shift from E° due to changes in the supporting electrolyte's ionic strength, which in turn causes a change in the analyte activity coefficient. Potential changes can often be described using the Nernst equation (Eq. 1). In order to predict the potential shift from E° , the activity coefficient of the analyte must be known. In solutions with high ionic strengths, these become impossible to predict by theoretical means, and therefore CV and ASV experiments are often employed to elucidate the redox potentials for a system. Therefore, CV and ASV experiments were performed to acquire an understanding of the redox potentials for the $\text{Cu}^{2+}/\text{Cu}^0$ system in the mobile phase.

$$E = E^\circ + \frac{RT}{nF} \ln \left(\frac{\alpha_{\text{Ox}}}{\alpha_{\text{Red}}} \right)$$

Equation 1: The Nernst Equation. In this equation R is the ideal gas constant, T is the temperature in °C, n is the number of electrons in the reductive process and F is Faraday's constant.

Cyclic Voltammetry Results

Figure 5 shows the results of CV experiments for a solution of 15.9 mM HNO_3 (solid line), and 15.9 mM HNO_3 + 425.0 mM KNO_3 (dotted line) using the Au RDE with a diameter of 1 mm. KNO_3 was introduced as a solid to the cell containing 15.9 mM HNO_3 and allowed to fully dissolve by the convective action of the RDE prior to analysis. The inset voltammogram shows a much wider potential sweep window that uses -700 mV as the first switching potential and +600 mV as the second switching potential.

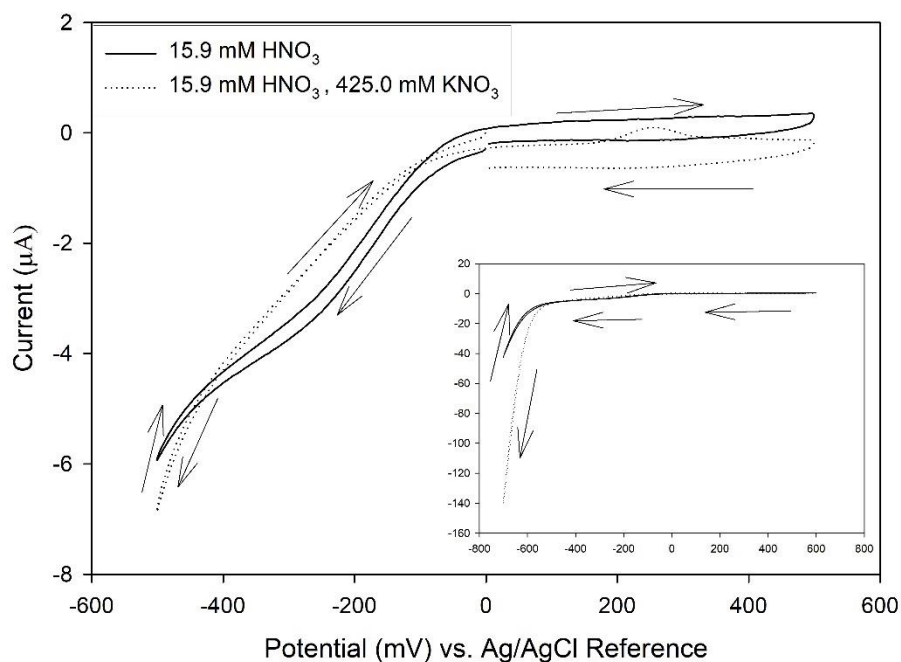


Figure 5: Background CV Results. Working electrode = 1 mm Au RDE, Pt wire Auxiliary electrode, Ag/AgCl Reference electrode. Arrows indicate potential sweep directionality. The CV waveform used the following parameters: $E_i = E_f = 0$ mV, $E_{\text{sw,low}} = -500$ mV, $E_{\text{sw,high}} = 500$ mV, $\Delta E/\Delta t = 50$ mV/s. Inset is the CV results using a low switching potential of -700 mV, and high switching potential of 600 mV, all other parameters were the same. During CV experiments the RDE was set to a speed of 900 rpm.

The large cathodic wave visible in Figure 5 starting at ca. -100 mV is attributed to the reduction of dissolved $O_{2(g)}$ in the supporting electrolyte solution. To determine the veracity of this hypothesis, a second CV was performed with collection occurring before (solid line) and after bubbling $N_{2(g)}$ (dotted line) through the supporting electrolyte solution for twenty (20) minutes. The results of this experiment are shown in Figure 6.

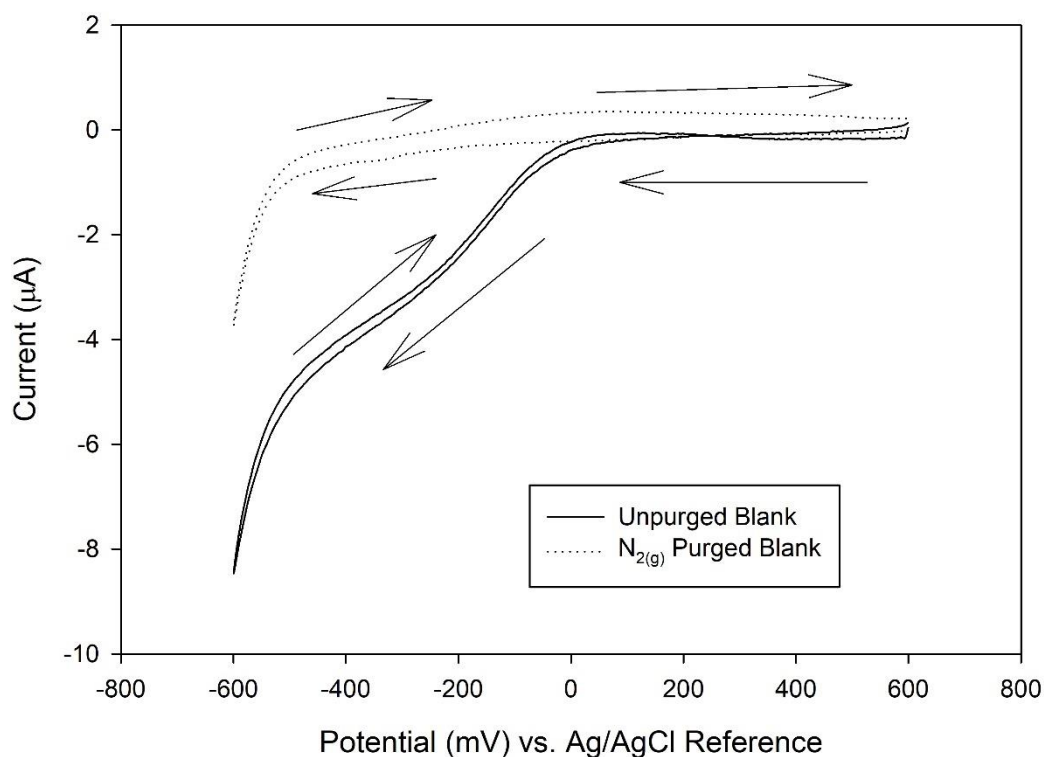


Figure 6: Effects of dissolved oxygen on CV response. Supporting electrolyte composition = 50.0 mM KNO_3 , 15.0 mM HNO_3 . Working electrode = 1 mm Au RDE set to 900 rpm, Pt wire auxiliary electrode, Ag/AgCl Reference electrode. Arrows show the directionality of the potential sweep. The large cathodic wave from ca. 0 to -600 mV disappears upon bubbling nitrogen gas through the supporting electrolyte solution.

Despite the apparent interference from dissolved $O_{2(g)}$ at negative potentials, this contribution was not expected to interfere with the analysis of copper in the IC-BPED system. This is because the deposition would occur at these negative potentials, prior to being stepped up to a more positive potential where dissolved $O_{2(g)}$ shows no interference. Additionally, as stated in Chapter 2, the mobile phase used in the IC-BPED method is thoroughly degassed and stored under $N_{2(g)}$ while installed on the IC system, so the amount of dissolved $O_{2(g)}$ was reduced significantly.

Figure 7 shows the CV results for the supporting electrolyte (solid line), as well as background subtracted CV traces for samples spiked to 5 ppm Cu (dotted) and 10 ppm Cu (dashed) using the 1 mm Au RDE with the following CV waveform: $E_i = E_f = +500$ mV vs Ag/AgCl Reference, $E_{switch} = -800$ mV vs Ag/AgCl reference, and $\Delta E/\Delta t = 50$ mV/s. This CV was collected without sparging the supporting electrolyte with $N_{2(g)}$ to show that detection of copper was possible without sparging of the analytical sample.

Copper reduction is observed in conjunction with the previously observed reduction from dissolved $O_{2(g)}$ in the region of -400 mV to -800 mV. This is followed by an anodic signal from ca. 0 to 250 mV. This oxidation signal is highlighted in the inset of figure 7, that shows the region from -400 mV to $+500$ mV of the CV sweep only. The dramatic increase of this signal with copper concentration indicates that it is due to the stripping of reduced copper, independent of dissolved $O_{2(g)}$. To assist in the analysis of the data, several functions were written in MATLAB™ that expedited the

analysis including several functions designed to combine and average CV data files.

These functions can be found in Appendix A.

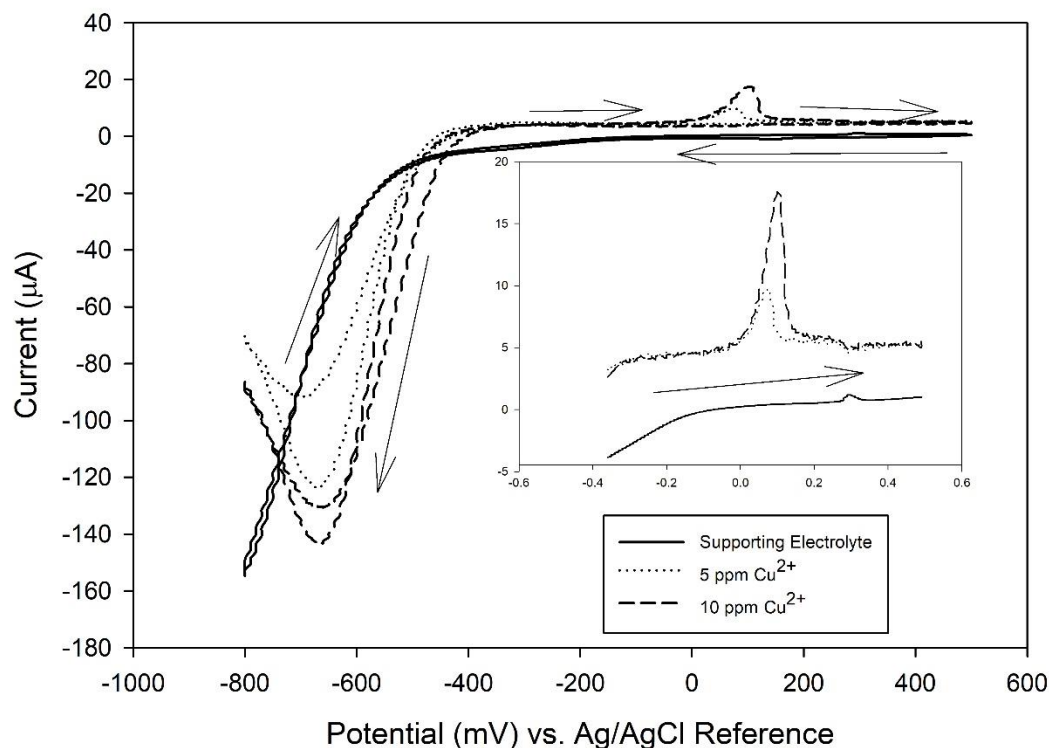


Figure 7: Copper CV results. Copper runs are background subtracted to more clearly show the reduction of the copper metal. Arrows give the potential sweep direction. Sweep was started at 500 mV at 50 mV/s and swept to -800 mV before returning to 500 mV. Inset is oxidative sweep from -400 mV to 500 mV. Supporting electrolyte composition = 425.0 mM KNO_3 , 15.9 mM HNO_3 .

Anodic Stripping Voltammetry

ASV was used to confirm the starting potential for deposition, and to observe the relationship between reduction time and stripping signal at the gold electrode surface. The results of the ASV studies for the deposition potential are shown in Figure 8 a, (left) and for deposition time in Figure 8 b, (right). For both of these experiments, the supporting electrolyte was not purged with $\text{N}_{2(g)}$. During the study of the deposition

potential, reduction time was held at a constant 30 s, and during the study of the deposition time the potential was held at -400 mV vs. Ag/AgCl. To assist in the analysis of the data, several functions were written in MATLAB™ that expedited the analysis these functions can be found in Appendix A.

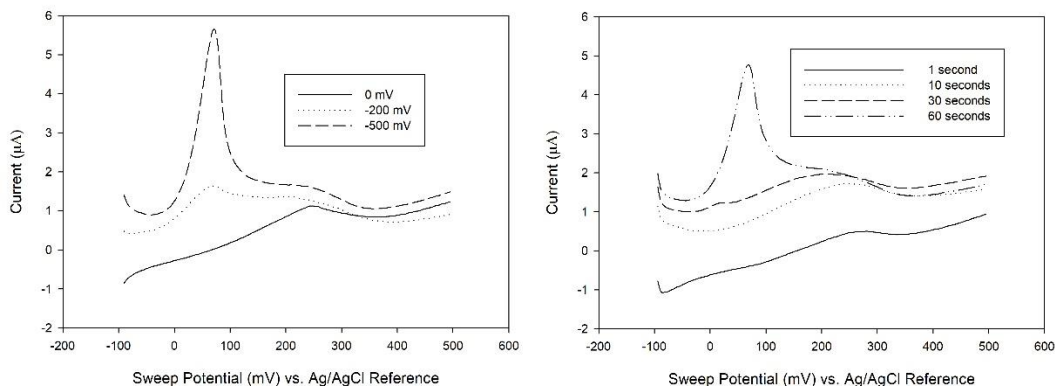


Figure 8: ASV Response at Au. Both experiments were performed using the 1 mm Au RDE in 425.0 mM KNO₃, 15.9 mM HNO₃. A (left) shows the results of the reduction time study for a 1.999 ppm Cu solution using a constant reduction time of 60 s. Note that oxidation is only observed when reduction times more negative than -200 mV are applied. B (right) shows the results of the reduction time study for a 1.999 ppm Cu solution, using a reduction potential of -400 mV. Supporting electrolyte composition was the same as for Figure 7.

As was observed in the CV data, reduction starts to occur at potentials more negative than -200 mV, which is more negative than the standard reduction potential of ca. 0 mV vs. the Ag/AgCl reference. When using -400 mV a reduction time of 30 seconds or more is required to generate significant signal, even for a concentration as high as 1.999 ppm. Taken together, this indicated that a significantly negative reduction potential would be required to achieve rapid reduction of copper on the electrode surface for IC-BPED analysis. The next section focuses on the development

of the IC-BPED method, the optimization of the IC-BPED waveform, method validation, and its application to the analysis of dietary supplements.

BPED after Ion Chromatography

Initial BPED Parameters

Based on the CV data the following parameters were chosen for initial IC-BPED analysis of Cu solutions: $E_{\text{red}} = -600 \text{ mV}$, $E_{\text{ox}} = 200 \text{ mV}$, $t_{\text{red}} = 500 \text{ ms}$, $t_{\text{del}} = 10 \text{ ms}$, $t_{\text{int}} = 20 \text{ ms}$, $t_{\text{clean}} = 200 \text{ ms}$. This waveform was immediately applied to the IC-BPED analysis of 1.00 ppm Cu solution as a proof of concept.

Initial IC-BPED Results

Figure 9 shows the results of IC-BPED analysis of an injection of 1.00 ppm Cu when using the initial BPED parameters. The signal to noise level is very low for this analysis, and the large negative signal at ca. 2 minutes is the void volume peak resulting from a change in ionic strength of the mobile phase. This was confirmed in subsequent injections, which were matrix matched to the mobile phase composition of 425.0 mM $\text{KNO}_3 + 15.9 \text{ mM HNO}_3$. Copper elutes from the CS12A column from ca. 5 to 7 minutes after injection with a tailing peak (asymmetry at 10% height > 1).

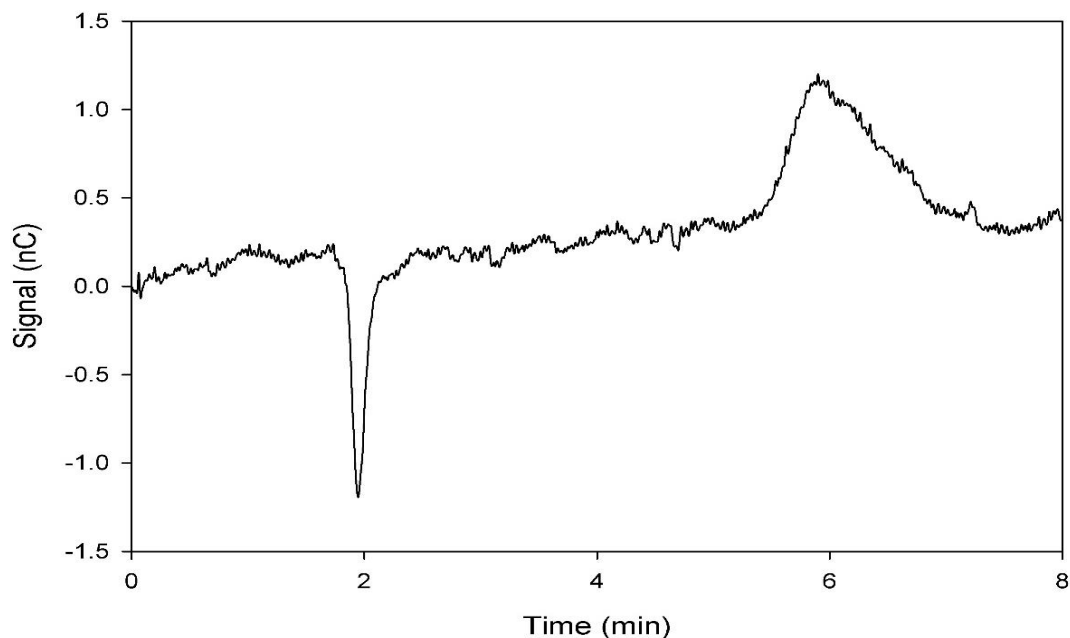


Figure 9: Initial IC-BPED waveform results for 1.00 ppm copper solution. The peak at 2 minutes is the system void volume, and copper elutes from ca 5 to 7 minutes with isocratic elution using a mobile phase of 425.0 mM KNO₃, 15.9 mM HNO₃ a gold working electrode (d = 1 mm) was used in the ED-40 flow cell. BPED parameters are: E_{red} = - 600 mV, E_{Ox} = 200 mV, t_{red} = 500 ms, t_{del} = 10 ms, t_{int} = 20 ms, t_{clean} = 200 ms. The method signal to noise was improved by a series of on-line optimization experiments.

Effects of [KNO₃] on Chromatographic Parameters

Analysis of the retention time and peak asymmetry at 10% height was carried out for various concentrations of KNO₃ to determine the optimum eluent. As shown in table 4, there is a miniscule effect on the retention time for copper as the concentration of KNO₃ is changed, and there is no apparent effect on peak asymmetry indicating that the separation remains consistent across a wide range of salt concentrations.

[KNO ₃] (mM)	Retention Time (min)	Peak Asymmetry at 10%
500	6.17	2.33
475	6.27	2.00
450	6.37	2.07
425	6.47	2.07
400	6.63	1.93
375	6.73	2.15

Table 4: Effect of KNO₃ concentration on copper retention time and peak asymmetry (10% peak height). As KNO₃ concentration is lowered, a miniscule shift towards longer retention times is observed, while no real effect on peak shape is observed. Each run was 30 minutes, with copper injection taking place at 0.90 min.

Optimization Results

On-line optimization of the BPED waveform was carried out by monitoring the change in the chromatographic signal to changes in single parameters. Parameters were optimized in the order of: E_{red} , E_{ox} , t_{del} , t_{int} , and then t_{red} . Signal to noise levels were determined by measuring the average height of the copper peak for a series of three 1.00 ppm injections, and comparing that to the average RMS noise measured by the PeakNet software. From this process, signal to noise plots for each parameter were generated and used to choose the optimal parameter values – an example of one of these plots is shown in Figure 10. Table 5 summarizes the optimization windows and optimum values chosen for each parameter tested throughout this process.

Parameter	Optimization Window	Optimum Value chosen
E_{red} (mV)	-100 mV to -700 mV	-700 mV
t_{red} (ms)	10 ms to 1000 ms	1000 ms
E_{ox} (mV)	0 mV to 500 mV	500 mV
t_{del} (ms)	10 ms to 30 ms	20 ms
t_{int} (ms)	10 ms to 200 ms	50 ms
t_{clean} (ms)	N/A	200 ms

Table 5: Summary of optimization process for IC-BPED waveform. Parameters were optimized in the order of E_{red} , E_{ox} , t_{del} , t_{int} , and then t_{red} . Changes in t_{clean} had no observed effect on the chromatographic signal, and therefore this parameter was not optimized.

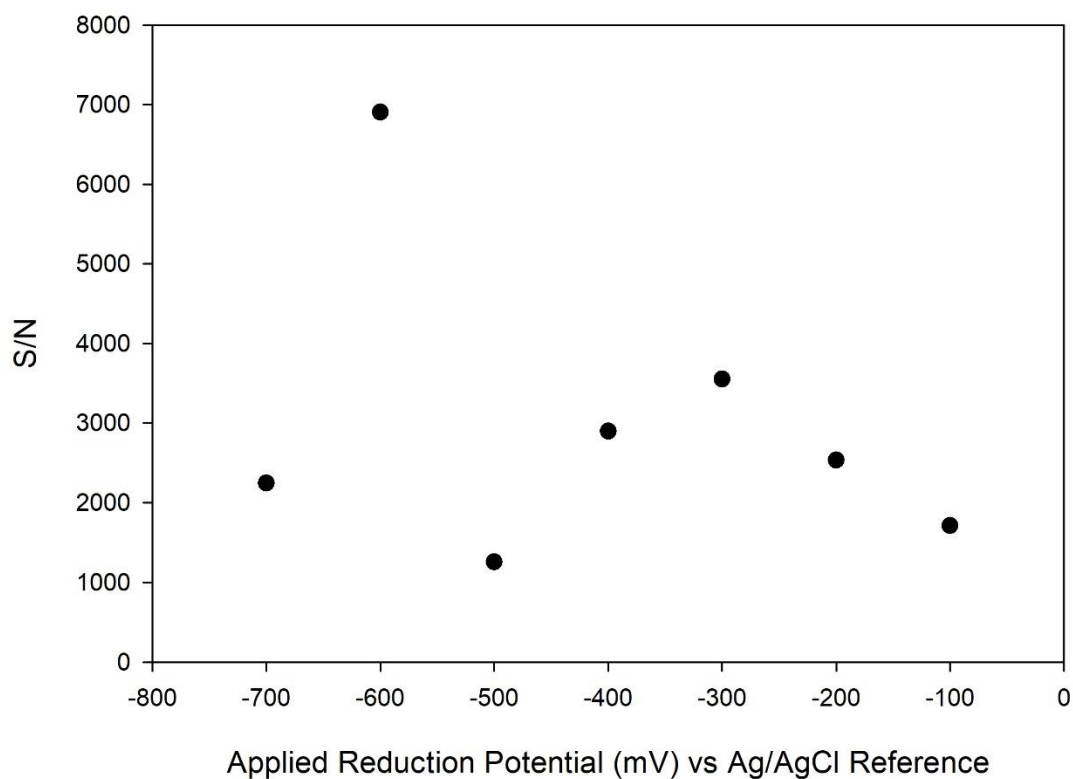


Figure 10: Example IC-BPED S/N plot. -700 mV was chosen for the E_{red} of the waveform to ensure the reduction of as many metals in the sample mixture as possible, specifically to ensure the reduction of Pb^{2+} . However, if a more selective waveform is desired, the use of -600 mV would be optimum.

Optimization of the E_{red} parameter was in strong agreement with observed CV data, though a local maximum of ca. -300 mV was also observed, as shown in Figure 10. The large increase at -600 mV is attributed to a large increase in reduction kinetics offered by the increased potential. Likewise, the E_{ox} optimization also agreed with the CV observations, with the optimum chosen parameter given in Table 5. With the time parameters, the ED-40 module limits waveforms to a total of two (2) seconds, and the

minimum increment for a parameter is ten (10) milliseconds. Ten milliseconds at the end of the waveform were reserved for a recycle time to revert from E_{ox} to E_{red} , so only 1.99 seconds of time were available for optimization. Due to the rapid kinetics of the stripping, as delay time was increased beyond twenty (20) milliseconds, a significant drop in S/N was observed. This rapid stripping also accounts for the relatively short optimal integration time of 50 milliseconds. Finally, the optimization of the reduction time gave results as expected from the ASV experiments performed – longer reduction times give increased signal to noise. Only two of the values tested in the window given in Table 5, 500 milliseconds and 1000 milliseconds, gave any appreciable signal above the noise level.

Table 6 gives the optimized IC-BPED waveform parameters found in the on-line study. The full ED-40 cycling frequency of 0.5 Hz was utilized, and t_{red} was allowed to occupy the bulk of that time, to ensure maximum sensitivity. The reduction potential chosen is off optimum for copper detection, but was chosen to be sufficiently negative to detect other metals.

Parameter	E_{red}	E_{ox}	t_{red}	t_{del}	t_{int}	t_{clean}
Value	-0.70 V	0.50 V	1.74 s	0.01 s	0.05 s	0.20 s

Table 6: Optimized IC-BPED values. Units are those used in the ED-40 firmware. The maximum frequency of 0.5 Hz was used for this waveform, resulting in a total waveform time period of 2 s. The chosen reduction time was selected to allow the detection of metals in addition to copper, despite being sub-optimal for the analysis of copper itself.

Method Validation and Analytical Figures of Merit

Analytical Figures of Merit

After optimization of the IC-BPED waveform was complete, a series of calibration standards were run on the system to determine several analytical figures of merit including: limits of detection (LOD), limits of quantitation (LOQ), analytical sensitivity, and limits of linearity (LOL). The resulting chromatogram is shown in Figure 11. The calibration is a series of injections timed every thirty (30) minutes to ensure that no overlap could take place with a system peak that appeared ca. 28 minutes after each injection.

The analytical figures of merit for the IC-BPED method are given in Table 7. Quantitation of analytical signal was performed by determining peak height using Dionex PeakNet™ software. Cadmium and copper show the most sensitive responses, with LOQs of 50 ppb. Lead however, does not show the same sensitivity with an LOQ of 300 ppb. Cadmium has the lowest limit of linearity (LOL) of 600 ppb, while copper has a LOL of 1.00×10^3 ppb and lead has an observed LOL of 5.00×10^3 ppb. All analytical sensitivities given are relative to the method and the instrumentation used, the use of a different instrument would cause these to vary; however, this is a useful metric for comparing the relative sensitivity of the method across analytes on a single instrument. Copper exhibits the greatest analytical sensitivity of the three metals (445 ± 7 nC/ppb), while lead has the lowest (100 ± 4 nC/ppb) and cadmium is intermediate. All three metals produce signals that have RSDs < 5% and exhibit linear

least squares regression coefficients (R^2) > 0.99 for Pb and Cd, with Cu having an R^2 > 0.999.

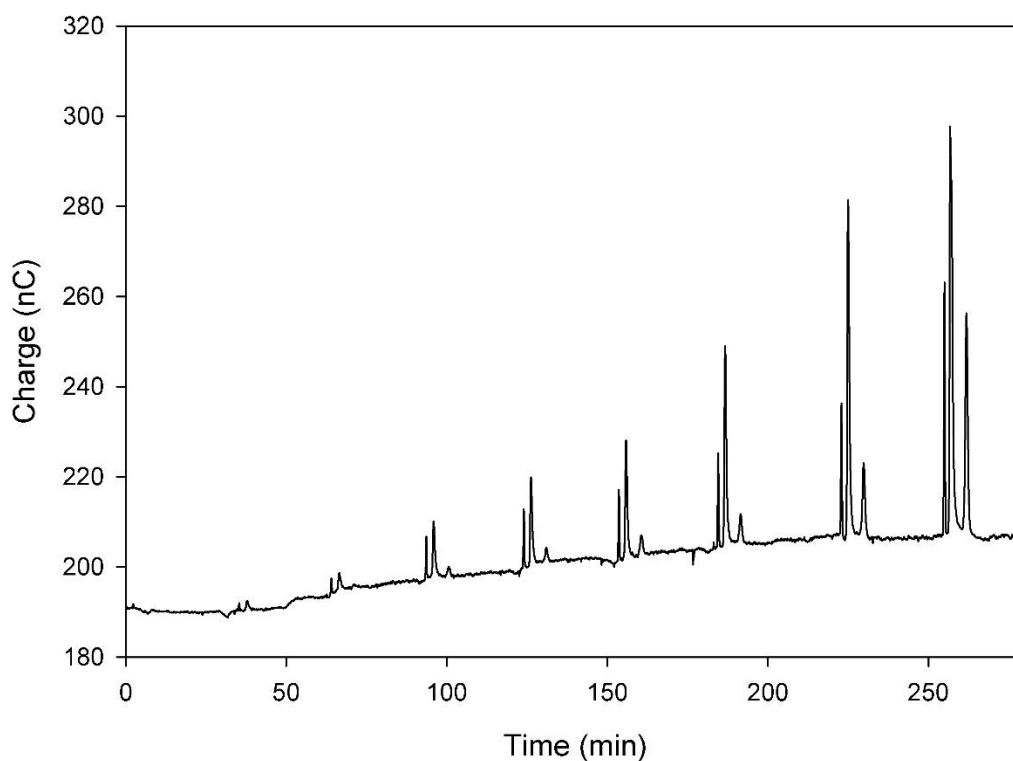


Figure 11: Calibration series run on the IC-BPED system. Injections increased in concentration from 0 ppb Cd, Cu, Pb to 5.00×10^3 ppb Cd, Cu, Pb. Injections were performed 30 minutes apart. Quantitation was performed using peak heights in Dionex PeakNet software.

Metal/Figure	LOD (ppb)	LOQ (ppb)	LOL (ppb)	Analytical Sensitivity	%RSD
Cu	20	50	1.00×10^3	445 ± 7 nC/ppb	3%
Cd	20	50	600	260 ± 10 nC/ppb	4%
Pb	200	300	5.00×10^3	100 ± 4 nC/ppb	4%

Table 7: Analytical figures of merit for the IC-BPED method determined for Cu, Cd, and Pb. All units are in parts per billion, with exception to LOL which is given in ppm. LOD estimated using $3 \cdot \sigma_{bl}$ and LOQ estimated using $10 \cdot \sigma_{bl}$.

Figure 12 is a chromatogram showing the result of a series of replicate injections of 1.00 ppm metal mixture. This chromatogram was analyzed to determine the repeatability of the method, and the %RSD was calculated to be 3% for copper, and 4% for lead and cadmium.

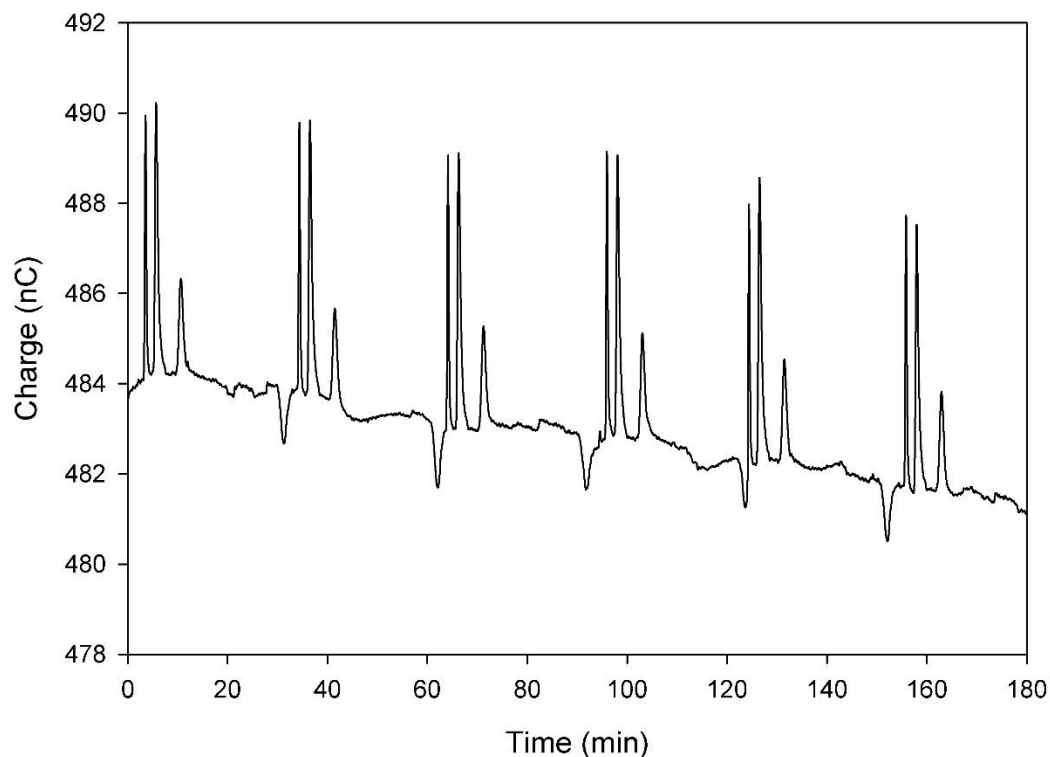


Figure 12: Chromatographic reproducibility of IC-BPED method. Six replicate injections of 1.00 ppm mixed metals standard was used. All three metals gave signals with RSDs < 5% based upon peak heights calculated in Dionex PeakNet. Copper exhibited the greatest reproducibility with an RSD of 3%. The baseline change during the run is due to instrumental drift.

Method Validation – Blind Samples

Validation of the HPLC-BPED method was carried out by performing a blind study with a sample that was prepared by a colleague containing any proportion of Cd,

Cu, and Pb, from none to the method's limit of linearity for each metal. This sample was then analyzed using a bracketed analysis method on the IC-BPED system with three replicate injections and three injections spiked with 100.0 ppb standard mixture. The resulting chromatogram is shown in Figure 13.

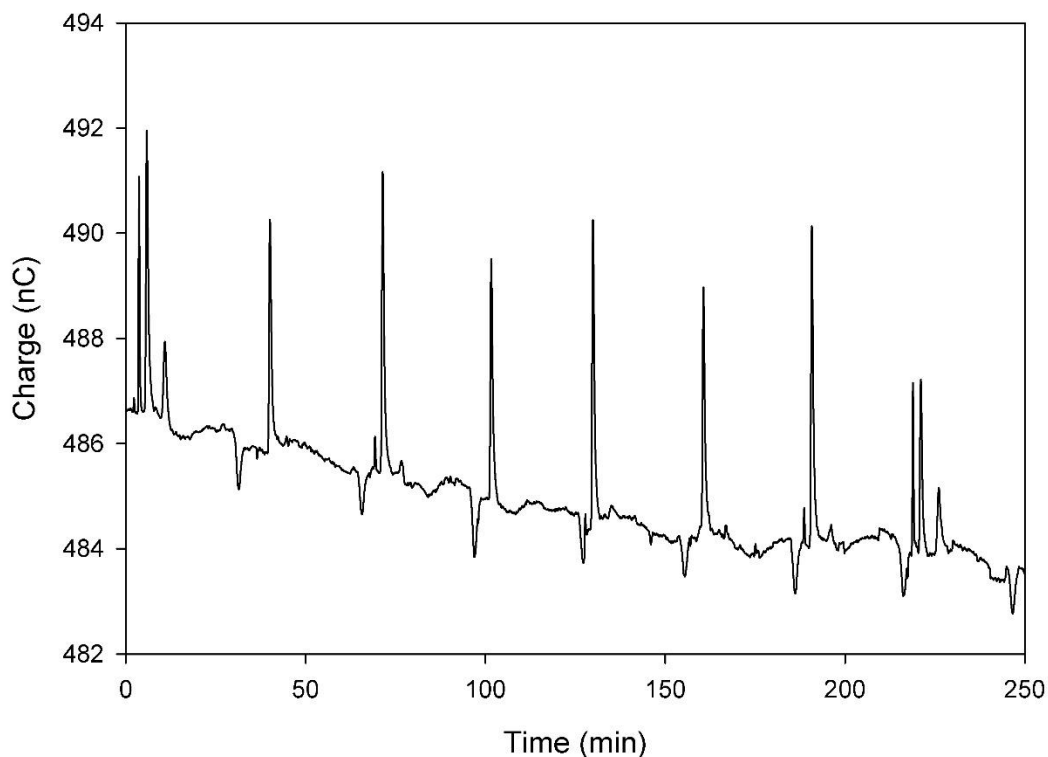


Figure 13: Chromatographic run for the blind sample. Injections are in order of high standard, a repeated pair of blind and spiked blind (3 times) and low standard and are performed 30 minutes apart for a total run time of 250 minutes (8 injections).

The blind sample was determined to contain 810 ± 20 ppb Cu and no Cd or Pb, at 95% confidence, which agreed with the calculations used to prepare the sample that stated 800 ppb of Cu only was used in the sample's preparation. The %RSD for this method is approximately 2.5%, and within the average tolerance of 3% for

electrochemical methods. To add strength to the validation, the method was also used to analyze the copper content of various dietary supplements. Three supplements were chosen: a liquid supplement, a solid tablet, and an encapsulated powder. The analysis of these supplements is discussed in the next section of this document.

Analysis of Copper Dietary Supplements

Liquid Copper Supplement Analysis

A liquid copper dietary supplement manufactured by SENSEable™ supplements was analyzed by Ultraviolet-Visible (UV-Vis) spectroscopy and by IC-BPED. Additionally, the manufacturer was willing to give the preparation procedure for their supplement, which states that 25.00 g of Copper Sulfate pentahydrate solid (USP grade) is dissolved in 25.00 L of pure water and then transferred to the dropper bottle. This generates a solution of 6.36 ppth Cu. In order to analyze this supplement by IC-BPED a 100 μ L aliquot was diluted 1:100, and a subsequent 1:100 dilution was made for a total dilution factor of 1:10,000. When analyzed by IC-BPED using 6 replicate injections in a bracketed analysis, a concentration of 6.3 ± 0.1 ppth at 95% confidence was determined for the original sample, which agrees with both the labelling of the supplement bottle and the preparation procedure provided by the manufacturer. When analyzed by UV-Vis spectroscopy, the concentration of copper in this supplement was found to be 6.38 ± 0.02 ppth at 95% confidence, which also agrees with the bottle labeling and the manufacturer's preparation calculations. Additionally, a Student's t-test gives a t_{exp} value of 1.32 when comparing the results of

the IC-BPED method to the UV-Vis method, which is less than the t_{crit} value at 95% confidence for an analysis with $N = 7$ degrees of freedom ($t_{\text{crit}} = 2.365$) indicating that the two means are not statistically different. No Cd^{2+} or Pb^{2+} was detected in this sample.

Additional Dietary Supplements and Tap Water Analysis

In addition to the liquid dietary supplement, a tablet supplement manufactured by Solgar, Inc. was also analyzed by the IC-BPED method. This was done by placing the tablet in a vial, pipetting 20.00 mL of supporting electrolyte solution into the storage vial, and allowing the tablet to digest overnight. The resulting solution was then diluted 1:200 prior to analysis on the IC-BPED system using a bracketed analysis. Based on the bottle label, each tablet contains 2 mg of copper chelated with glycine. The IC-BPED results determined that 1.8 ± 0.2 mg of copper was present in the tablet, which agrees with the bottle label, and no cadmium or lead was detected in this sample.

An encapsulated powder supplement was also analyzed using the HPLC-BPED method, with limited success. The supplement was prepared by placing one capsule in a glass storage vial and pipetting 20.00 mL of supporting electrolyte solution into the vial, and allowing the capsule to digest overnight. The resulting solution was diluted 1:200 prior to analysis on the IC-BPED system in a bracketed analysis. The resulting chromatogram is shown in Figure 15. Based on the label, 2 mg of copper citrate powder are contained within each gelatin capsule. This would mean that the resulting solution, prepared as described above, should have a concentration of approximately 500 ppb,

though the sample could vary from ca. 375 ppb to 625 ppb based on the bottle label. In figure 14, it is observable that the signal generated by the supplement exceeds the high concentration bracket, indicating that a concentration of over 600 ppb copper is present, and no cadmium or lead was detected in this sample.

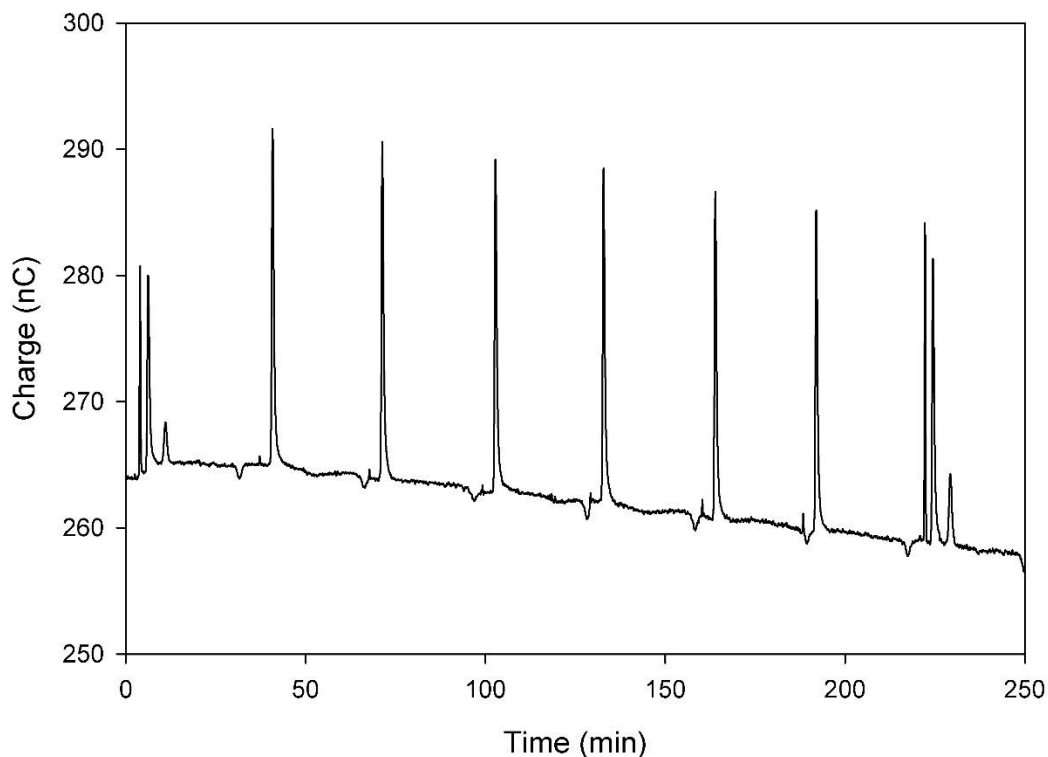


Figure 14: Chromatogram for copper citrate supplement. Signals for copper exceed the high concentration bracket of 600 ppb. Eight injections were performed 30 minutes apart to prevent overlap with the system peak that occurs at ca. 28 minutes after injection.

Finally, tap water drawn from the laboratory sink was analyzed on the IC-BPED instrument using a bracketed analysis of 6 replicate injections. Fortunately, no lead or cadmium was detected in the tap water sample, but the copper level was found to be between 700 ppb and 580 ppb.

BPED after Ion Chromatography at Glassy Carbon

While a method at a gold working electrode is useful for determination of copper in most samples such as ground waters, tap water, beverages, etc. This method falls short in samples where chloride concentration is very high. This is due to an inherent incompatibility between the gold working electrode and solutions of high chloride content. Unlike with gold, the signal generated by the BPED waveform on glassy carbon was a negative peak that had a weakly linear correlation with concentration. The response for an injection of 5 ppm Cu, Cd and Pb is shown in Figure 15.

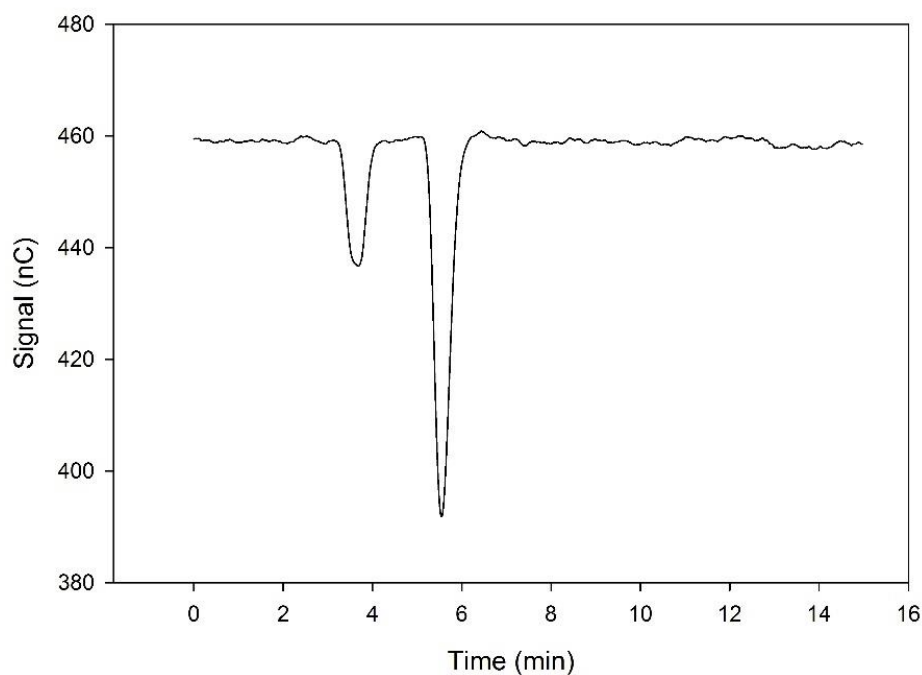


Figure 15: Chomatogram of 5 ppm Cd, Cu, and Pb using glassy carbon working electrode. Pb is not detected, Cd and Cu generate a negative response. Mobile phase is 425.0 mM KNO₃, 15.9 mM HNO₃. This signal exhibits a linear correlation from 2.00 to 10.0 ppm for Cu ($R^2 = 0.95$).

This indirect response is linear, with a working range of 2.00 to 10.0 ppm of Cu ($R^2 = 0.95$). While this may be of use to industrial analysis or analysis of waters suspected to be highly contaminated, it is not useful for the analysis of copper in the therapeutic region of aquaria or at the levels used for routine drinking water analysis.

Conclusions

This research has presented a method that can separate and quantify Cu, Cd, and Pb in freshwater samples. With limits of detection in the parts per billion range, and limits of linearity in the low parts per million range, the method is applicable to most environmental samples with only simple sample preparation protocols required. The effects of chelating acids in the mobile phase pose an intriguing avenue for future study, as they have an observable effect on both the chromatographic peak shape and on the BPED sensitivity for copper. The largest limitation with the method is the incompatibility of gold electrodes with samples containing large concentrations of chloride.

While a silver cartridge could be used to overcome this limitation, a glassy carbon electrode was investigated as an alternative material because it is chloride compatible. When the same waveform was applied to the glassy carbon electrode a negative detection peak was observed that had a linear response factor from 2.0 to 10.0 ppm ($R^2 = 0.95$).

Chapter 4: BPED at the Benchtop

Introduction

During optimization of the IC-BPED waveform, it was observed that the signals for Cd and Pb would diminish more rapidly than the signal for Cu as the reduction potential was moved more positive. This was attributed to the relative differences between the standard reduction potentials of the three metals, and termed the tunability of the BPED waveform. This concept is illustrated in Figure 16. These observations led to the hypothesis that, if the reduction potential was sufficiently controlled, a benchtop BPED analysis method for copper could be developed which did not require the use of ion chromatography prior to BPED, and it could then be possible to develop a point-of-use analysis test for copper and other metals. This chapter shows the results of research aimed at achieving that end.

First discussed is the analysis of copper at gold electrodes using BPED, the optimization of the BPED waveform and the analytical figures of merit determined for that method. This chapter concludes with a discussion of attempts to generate an analogous method at a glassy carbon working electrode, which was ultimately less successful.

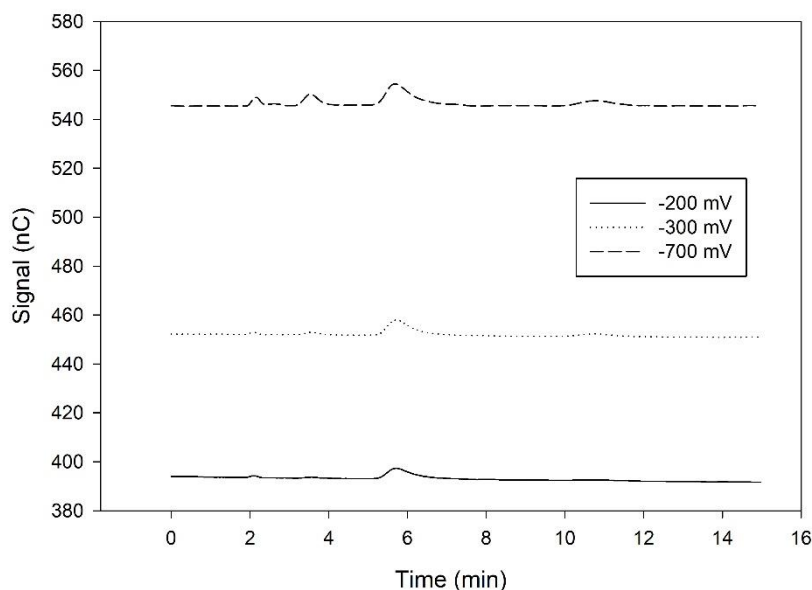


Figure 16: Tunability of the BPED waveform. Signal is not corrected for background current (no offset was performed). As reduction potential is made less negative, the Cd and Pb peaks rapidly fall off, leaving only the Cu peak as the predominant analyte in the chromatogram. Reduction potentials are given in the legend and correspond as follows: solid line -200 mV, dotted line -300 mV, dashed line -700 mV vs. Ag/AgCl Reference.

As can be seen in Figure 16, the cadmium peak at ca. 4 min and the lead peak at ca. 10 minutes fall off in intensity very rapidly as E_{red} is made less negative. This is due to the relative differences in the standard reduction potentials of lead ($E^\circ = -126$ mV vs. Standard Hydrogen Electrode) and cadmium ($E^\circ = -403$ mV vs Standard Hydrogen Electrode) compared to copper ($E^\circ = 342$ mV vs Standard Hydrogen Electrode).

Benchtop BPED at Gold

Initial BPED Parameters

The initial BPED waveform parameters are given in Table 8. The waveform was applied to a solution of supporting electrolyte, then the solution was spiked with 1.00 ppm Cu. An increase in signal indicated that the waveform was detecting Cu, the results are shown in Figure 17.

Parameter	E_{red} (mV)	E_{ox} (mV)	t_{red} (s)	t_{del} (ms)	t_{int} (ms)	t_{clean} (ms)
Value	-500	250	30	0	1000	0

Table 8: Initial parameters for benchtop analysis of copper on Au electrodes. The parameters were chosen based on CV and ASV results observed in previous experiments.

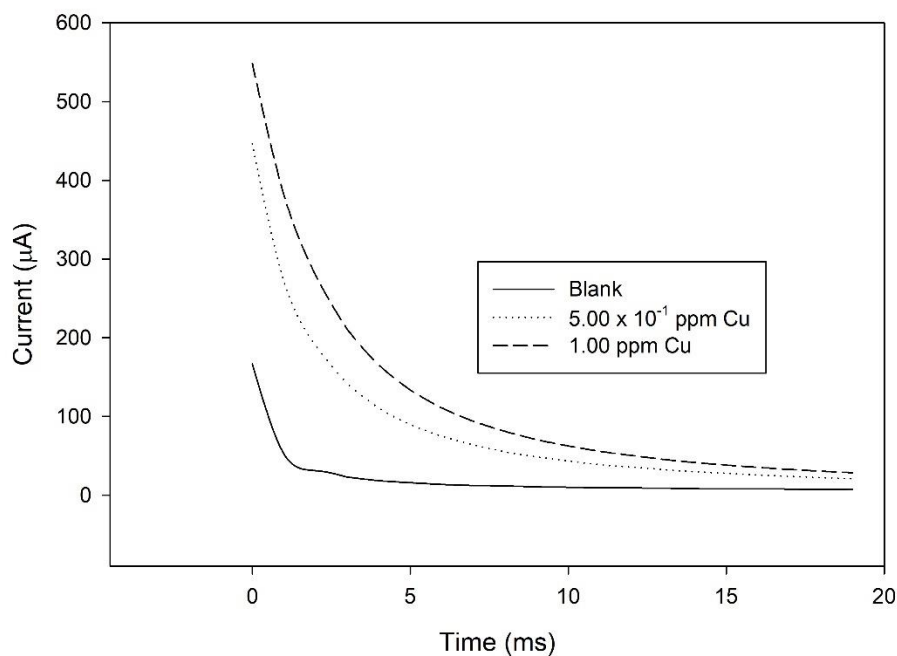


Figure 17: Application of BPED waveform at Au working electrode to analysis of copper in 250.0 mM KNO₃, 15.0 mM HNO₃. Waveform was applied to copper spikes of 5.00 x 10⁻¹ ppm and 1.00 ppm – there is significant current from 0 to 15 ms observable due to the stripping of deposited copper.

This waveform gave satisfactory detection at very high concentrations of copper, but further optimization was desired to improve the waveform to ensure peak performance. This optimization process is more complex than the on-line optimization used during the IC-BPED method because no software existed that could readily import, parse and calculate the various figures required to determine optimal waveform parameters such as the area under the curve and S/N. Throughout this work, MATLAB™ was used to assist in data analysis, and an extensive amount of code was written for the analyses performed. These functions and scripts can be found in Appendix A of this text.

MATLAB Functions for Data Analysis

Initial code was written to simply import the data files (output as comma delimited text by the AfterMath™ software) and combine them into a single, workable text file. This function was used for all analyses, including those performed using CV and ASV to reduce the amount of time between experiment and results. This is especially true for optimization of the delay and integration time parameters, where it is necessary to remove a single row (millisecond) of data at a time and repeat the entire series of computations (convert to area, determine averages, determine noise, compute S/N). When one considers that an average data set contained at least 1000 rows of data, it was clear that an automated process would significantly reduce delays in the progress of this research. Automation of these calculations reduced the time required to perform analysis from several days, to several minutes. Additional code files were

written to execute commands rapidly on-demand as the data was compiled. These MATLAB™ script files are also available in Appendix A. While a significant amount of time was spent writing these files, they can be used by future researchers who wish to accomplish similar goals with their data analysis – and thus represent a significant investment in future time savings as well.

Waveform Optimization

The waveform was optimized by successive manipulation of a single waveform parameter and observing the effect on the signal to noise ratio. Signal to noise ratios were calculated by subtracting the average area of three background (supporting electrolyte only) measurements from three replicate measurements of 1.00 ppm Cu. Background corrected copper areas were averaged together to generate an average signal measurement, and divided by the standard deviation of the same areas to determine the signal to noise ratio (S/N). The standard deviation is used as a representation of the noise because the integrated area will vary based on the total system noise.

This calculated S/N was plotted against the parameter as shown in Figures 18 through 22, and the parameter which gave the greatest signal to noise value was chosen. When several values had similar signal to noise values, the parameter that would give the greatest waveform cycling frequency was chosen.

Figure 18 depicts the S/N vs. the applied reduction potentials in a range of -600 mV vs. Ag/AgCl to -150 mV vs. Ag/AgCl, which was chosen based on the observed behavior from the CV data in Figure 7 (chapter 3).

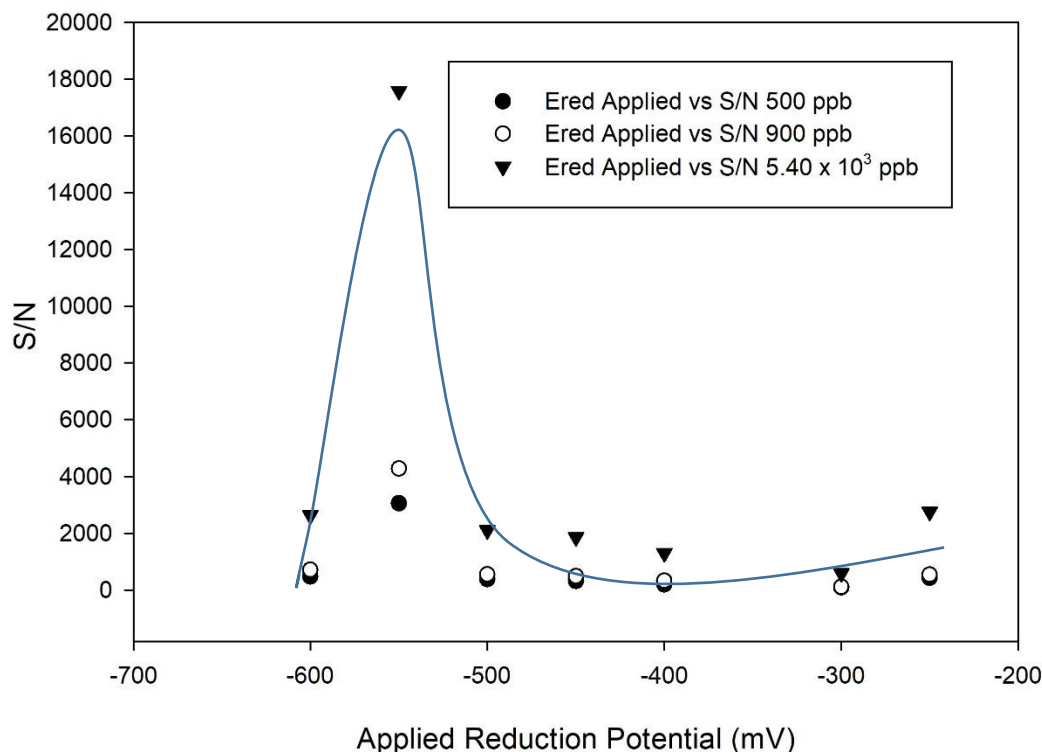


Figure 18: S/N plot for Au benchtop BPED reduction potential optimization. Window was -600 mV to -150 mV, and optimization was repeated over a series of three copper spikes (500 ppb, 900 ppb, and 5.40×10^3 ppb). A strong spike for all concentrations of copper analyzed is observed at -550 mV vs. Ag/AgCl reference. Trendline does not represent a mathematical fit, but is simply added for clarity of illustration of the observed S/N trend.

Figure 17 shows that the optimum E_{red} is -550 mV vs. Ag/AgCl reference across multiple concentrations of copper from 5.00×10^{-1} ppm to 5.40 ppm. This is in the observed reduction window from the CV data (figure 7, chapter 3), and was chosen and used in future optimizations. The general S/N trend is represented by the drawn line in

the figure, and is not meant to represent a mathematical fit – it only serves to assist in illustrating the observed S/N trend.

Unlike in the IC-BPED method, the benchtop method utilizes a commercial potentiostat that is not limited to a waveform of 2.00 s or less. This allows for a significantly longer reduction time to be applied. Optimization of the reduction time was carried out using a series of reduction times from 30 s to 120 s. This window was chosen based upon the results from the ASV experiments performed in chapter 3. Figure 19 shows that reduction times and copper signal are near directly proportional, though there is a fluctuation in the actual S/N data, the source of this fluctuation is unknown. Despite the fluctuation, a reduction time of 100 s was chosen for further work. Again, the trendline in the graph is not meant to depict any mathematical fit to the data – it is merely an aid to illustrating the observed fit.

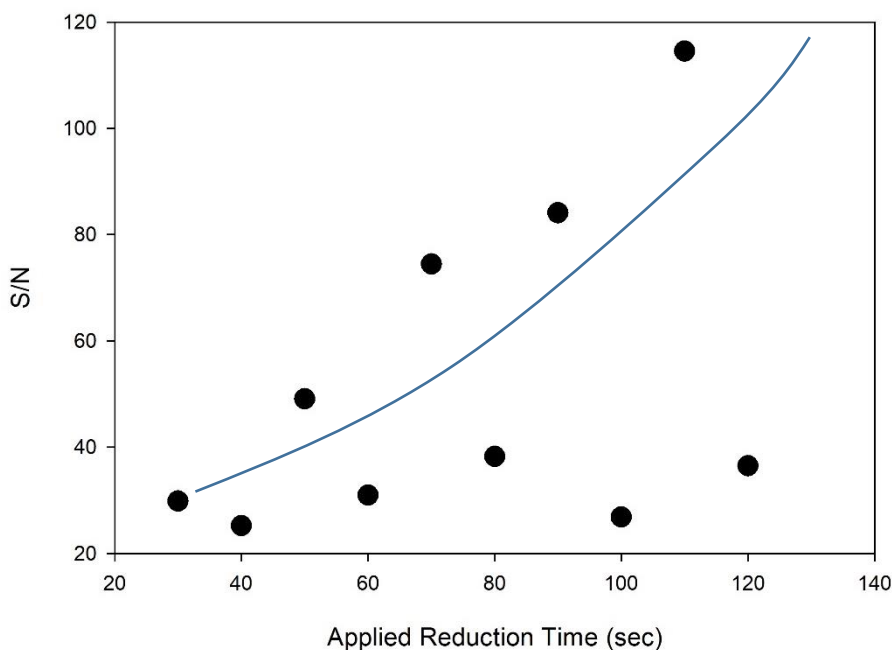


Figure 19: S/N plot of reduction time optimization for benchtop BPED at Au RDE. A reduction time of ca. 90 s is chosen as the optimum value for future experiments. The observed fluctuation does not correspond to any known phenomena.

In figure 20, the optimum value of 250 mV vs. Ag/AgCl reference for E_{ox} is seen based on the S/N values calculated. This also corresponds to the CV observations from figure 7 (chapter 3). Therefore, this value was chosen and used in the remaining optimization experiments. Once again the observed trend is illustrated with a drawn line, but this line does not represent a mathematical fit to the data – it is only placed for clarity of illustration.

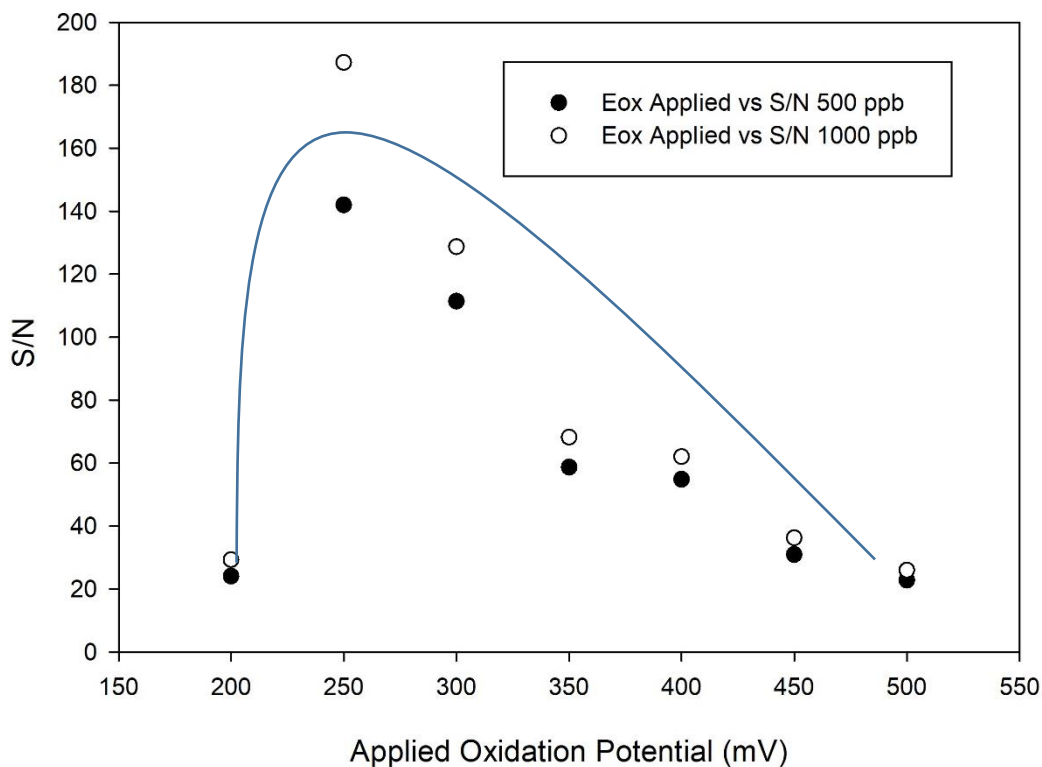


Figure 20: Optimization of oxidation potential for benchtop BPED on Au electrode. The observed optimum oxidation potential is 250 mV vs. Ag/AgCl reference, and is used in all further experiments. The drawn trendline is shown only to illustrate the observed trend, and does not represent a mathematical fit to the data.

Figure 21 shows the results of the delay time study performed, which made extensive use of various algorithms written in MATLAB™ (found in Appendix A) –In the data, there appears to be three maxima, 0 ms, 20 ms, and 40 ms. The 0 ms delay time is likely an artifact that is contaminated with non-reproducible, non-Faradaic double-layer charging current produced by the potential step. The maxima at 40 ms is attributed to an artifact of the potentiostat electronics interfacing with the 60 Hz line voltage (which has a period of ca. 16.67 ms). Therefore, 20 ms was chosen to ensure that the signal collected is free of this non-Faradaic current.

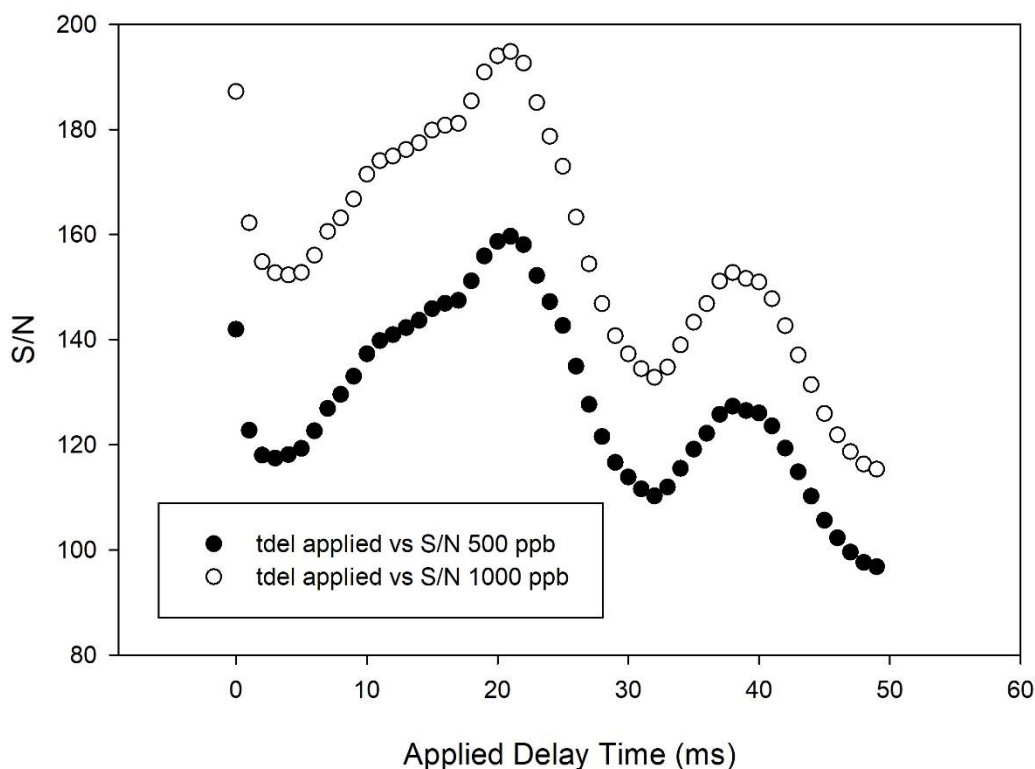


Figure 21: Delay time optimization results for benchtop BPED on Au. Extensive use was made of code written in MATLAB™ to make the many calculations required for this determination. 20 ms was chosen as the optimum delay time and used in further experiments.

Figure 22 shows that the optimum integration time was only a few milliseconds, but a review of the I vs. t curves themselves showed a significant portion of current above the baseline as far as 40 ms from 0, and so 20 ms was used as the integration time and the initial maxima attributed to an artifact of the current spike from the potential step. The cleaning time of 20 ms was chosen to ensure that the electrode was free of copper between waveform cycles.

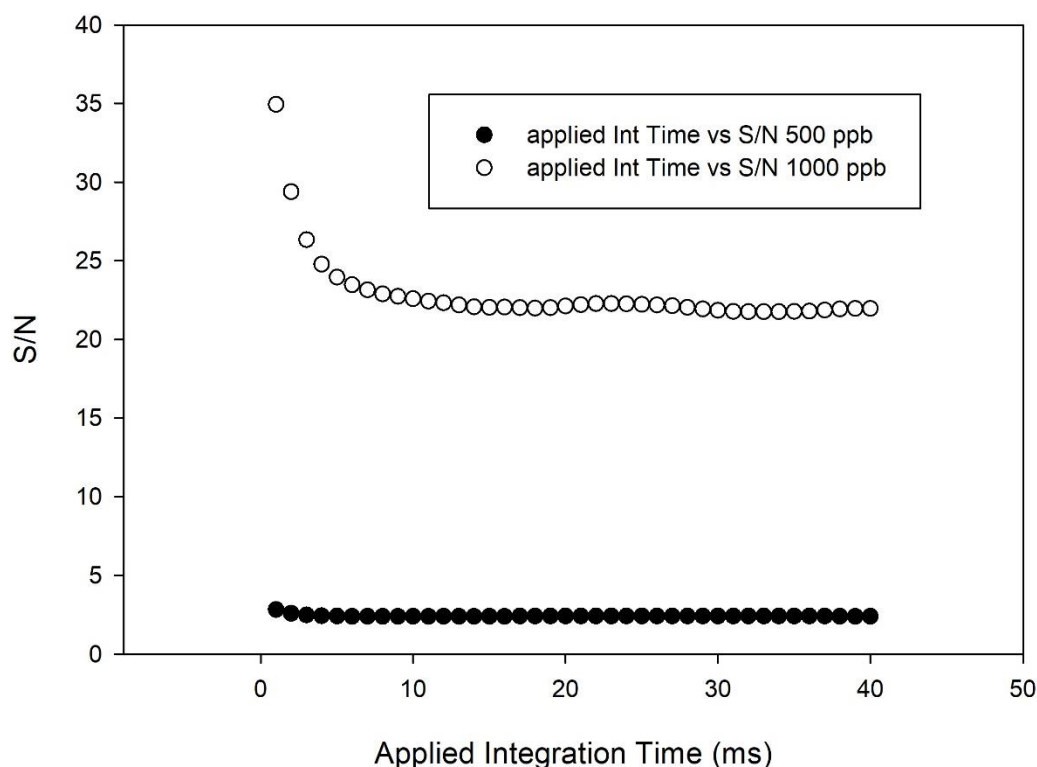


Figure 22: Integration time optimization for benchtop BPED on Au electrode. Despite the S/N spike at very short integration times, a review of I vs. t plots showed that a significant amount of current was present above background as far as 40 ms from the start of oxidation, and therefore a 20 ms integration window was chosen for future experiments.

A summary of the optimized parameters for benchtop BPED at the Au electrode is given in Table 9. These parameters were used to determine the analytical figures of merit for the method – which is the focus of the next section.

Parameter	E_{red} (mV)	E_{ox} (mV)	t_{red} (s)	t_{del} (ms)	t_{int} (ms)	t_{clean} (ms)
Value	-550	250	100	20	20	20

Table 9: Optimized analysis parameters for benchtop BPED copper detection at the Au RDE in 250 mM KNO₃, 15.0 mM HNO₃. These parameters were used in determining the analytical figures of merit for the method.

Analytical Figures of Merit

To determine the Analytical Figures of Merit for the benchtop analysis waveform, a calibration plot was constructed. The analytical figures of merit are summarized in Table 10, and the calibration plot is shown in Figure 23. This plot was made by spiking the supporting electrolyte solution (blank) with successive additions of copper standard solution and measuring the resulting BPED signal. The background corrected signal gave a linear response from 5 ppb to 75 ppb Cu²⁺, with a LOQ of 50 ppb ($R^2 = 0.997$). Analysis out to 500 ppb, shows that the signal falls off with additional copper in an exponential fashion. This full curve can be found in the inset of Figure 23.

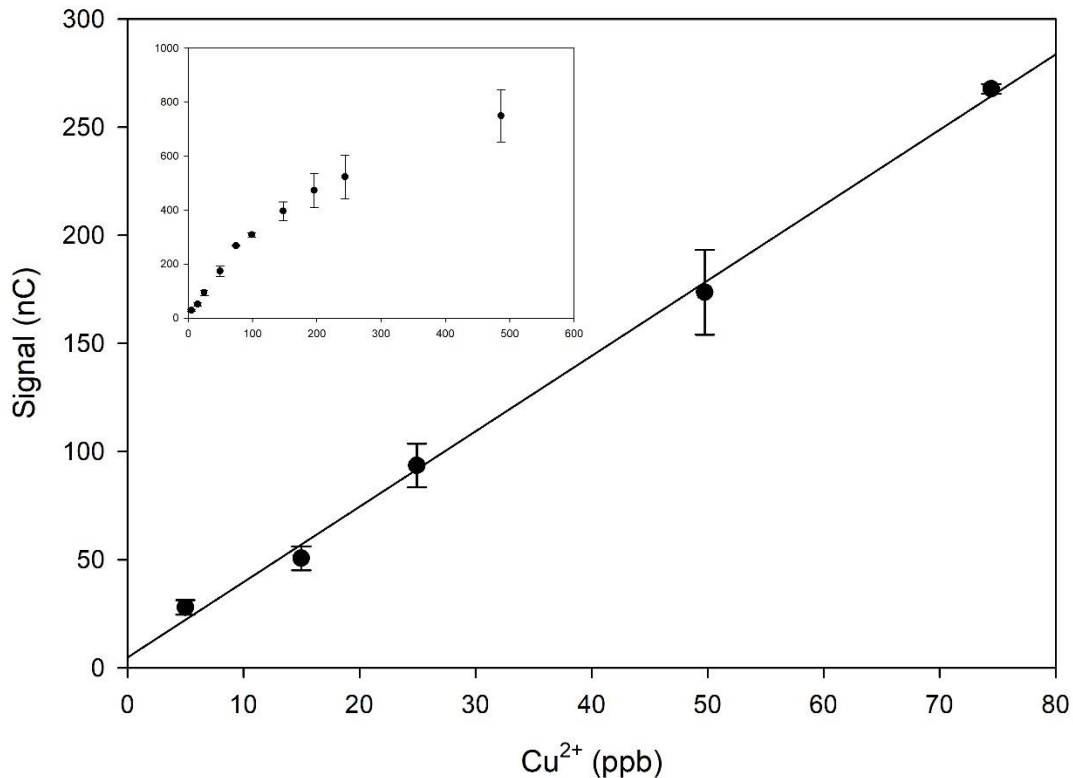


Figure 23: Calibration plot for analysis of copper by the described benchtop method at an Au RDE. Plot was made using the method of standard additions. The linear region for the method is from 5 to 75 ppb with $R^2 = 0.997$. Inset is the full calibration range to 500 ppb.

Figure	Value
LOD	5 ppb
LOQ	20 ppb
LOL	75 ppb
R^2	0.997
Analytical Sensitivity	3.49 nC / ppb Cu
Linear Dynamic Range	20.00 ppb – 75.0 ppb Cu

Table 10: Analytical figures of merit for benchtop analysis of copper at the Au RDE.

Summary of results

This research focused on the development and optimization of a benchtop BPED waveform for analysis of copper using Au electrodes. There is a single

limitation to this method that cannot be overcome however – the inherent incompatibility of Au electrodes with solutions of high chloride content. Therefore, additional work was performed which sought to investigate the possibility of using a glassy carbon electrode for benchtop BPED analysis. This work required writing an extensive amount of computer software to assist with the optimization calculations, and this software can be found in Appendix A.

Voltammetric Studies of Copper at Glassy Carbon

The behavior of copper at glassy carbon was investigated by CV in artificial seawater to determine what potentials were necessary to create a benchtop waveform. Unlike with gold, no nitric acid or potassium nitrate was added to the solution to generate a supporting electrolyte, as the high sodium chloride concentration of artificial seawater acts as the supporting electrolyte. The results of the CV study are shown in Figure 24.

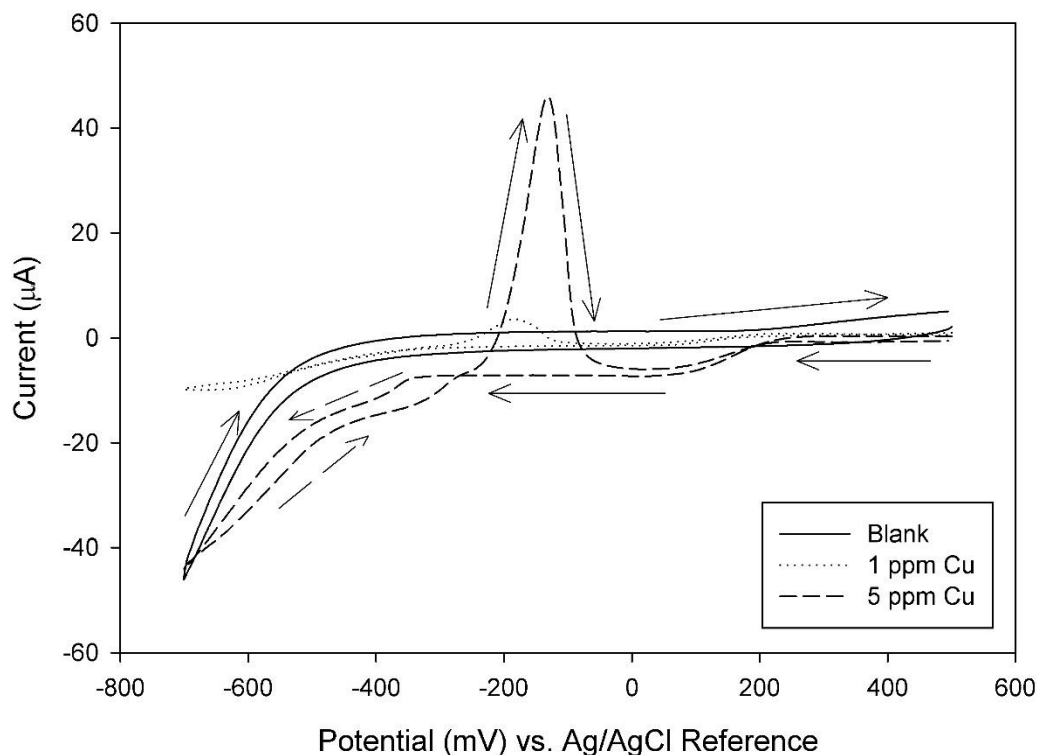


Figure 24: CV results for Cu in artificial seawater at the glassy carbon RDE ($d = 5$ mm). Arrows made with long dashes apply only to the 5 ppm Cu run, and all arrows show the directionality of the potential sweep. Reduction is observed from -400 to -700 mV in the 5 ppm Cu run, with oxidation occurring from -200 mV to 0 mV. Data was collected on a glassy carbon RDE spun at 900 rpm with $E_i = E_f = 500$ mV, $E_{\text{switching}} = -700$ mV, and $\Delta E/\Delta t = 50$ mV/s.

Copper deposition is observed again from ca. -200 to -700 mV as was the case with Au, but where glassy carbon differs from Au is in the stripping potential. This is from ca. -200 mV to 0 mV. Further inquiry on the redox behavior of copper in seawater at glassy carbon was done using ASV techniques.

Anodic Stripping Voltammetry

Again, just as with the Au electrode, ASV was used to assist in the initial study of reduction time dependence on the behavior of copper stripping. Figure 25 shows

the results of the ASV study for the glassy carbon electrode. In this experiment, the reduction potential was held at -600 mV vs. Ag/AgCl reference before being swept at a rate of 100 mV/s from -500 mV to 400 mV vs. Ag/AgCl. There is an initial stripping peak observable from 0 mV to 200 mV that decreases in intensity as reduction time is increased, while the stripping peak observed in the CV experiments from -200 mV to 0 mV increases as reduction time increases.

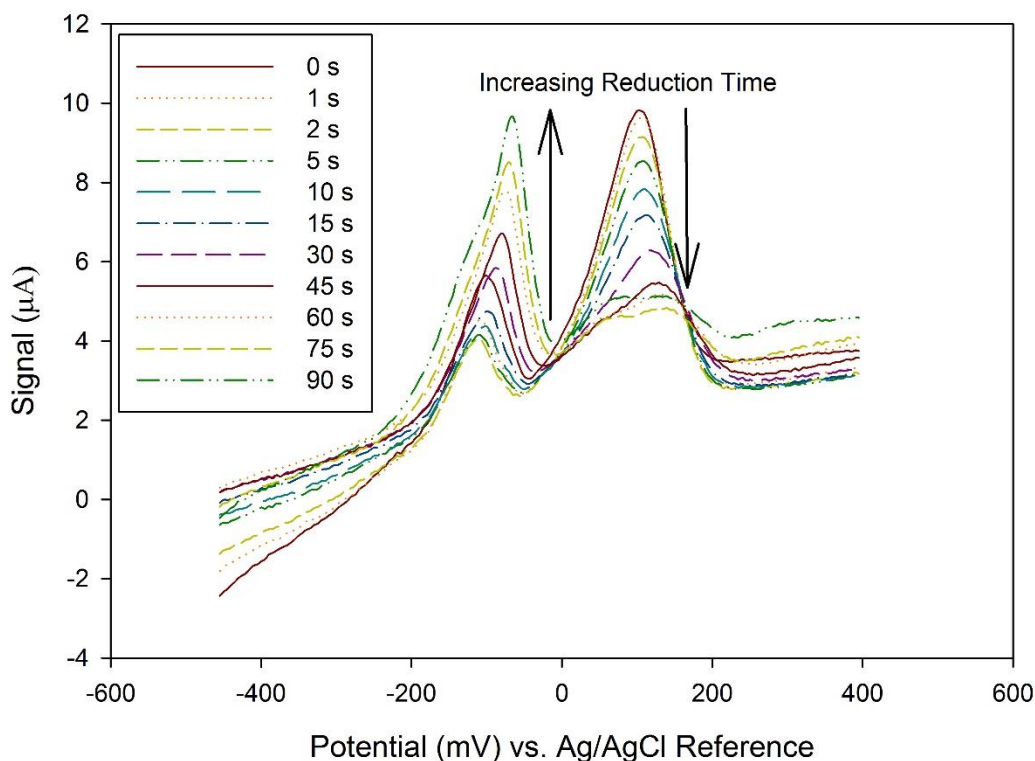


Figure 25: ASV study for copper at glassy carbon RDE ($d = 5$ mm) in marine water. The stripping peak from 0 to 200 mV decreases as reduction time is increased, and the peak from -200 mV to 0 mV increases proportionally with reduction time. Potential was swept from -500 mV to 400 mV at a rate of 100 mV/s. Copper was reduced onto the electrode prior to the potential sweep by holding the working electrode at a potential of -600 mV for the specified time.

Benchtop BPED at Glassy Carbon

As with the gold electrode, the initial waveform was built based upon the analysis of the voltammetric data provided. This was applied to to determine if BPED at glassy carbon was possible, especially given the observed results of IC-BPED using glassy carbon electrodes. The results of this experiment are given in Figure 26, and waveform parameters are given in Table 11. As the level of copper was increased in the electrochemical cell, the signal at $t = 0$ ms increased, and anodic currents were sustained from ca. 0 ms to ca. 800 ms.

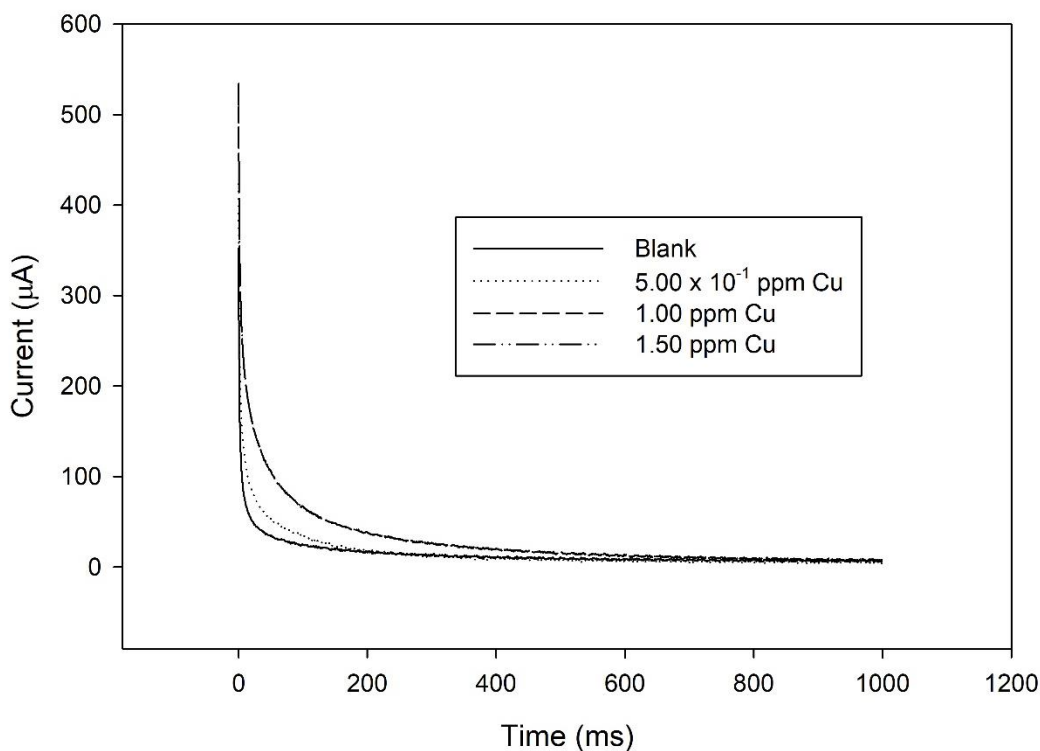


Figure 26: Batch BPED at glassy carbon with initial waveform parameters as described in table 10. As Copper concentration increased, anodic current was continually sustained from ca. 0 ms to ca. 800 ms. The largest signal increases are observed at $t = 0$ ms.

Parameter	E _{red}	E _{ox}	t _{red}	t _{del}	t _{int}	t _{clean}
Value	-550 mV	300 mV	30 s	0 ms	20 ms	180 ms

Table 11: Benchtop BPED waveform parameters for glassy carbon (d = 5 mm) RDE in marine water.

Calibration of Electrode Response and Application to a copper standard

The LOD, LOQ, and LOL for this method was determined using standard additions in a similar method to how it was determined for benchtop BPED at Au. The results from the standard additions plot of copper to artificial seawater are shown in Figure 27. The BPED method on glassy carbon gives a LOD for copper of 20 ppb, and a LOQ of 50 ppb, but has a higher LOL of 3.5 ppm. Additionally, the background signal levels for the glassy carbon benchtop method are significantly lower than that observed in the benchtop method using Au electrodes.

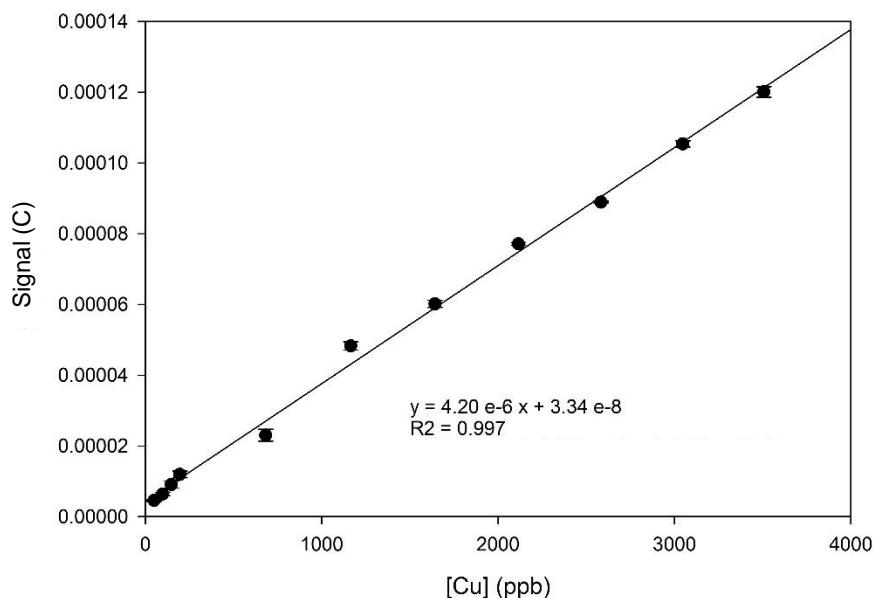


Figure 27: Standard additions plot for copper in marine water on glassy carbon electrodes using the benchtop BPED waveform parameters given in table 10. The method LOD was 20 ppb (estimated using $3 \cdot \sigma_{bl}$), the LOQ was 50 ppb (estimated using $10 \cdot \sigma_{bl}$), and the LOL was 3 ppm. The method gave a linear fit (using linear-least squares regression) with a regression coefficient (R^2) of 0.997.

The method was applied to the analysis of an 800 ppb sample of copper standard in artificial seawater. This was done by determining the response factor of the system for a 500 ppb sample, and 1000 ppb sample then averaging this response factor using a 15 ms delay time and 5 ms integration time (in effect calculating the slope of the line between these two points). Application this average response factor to the signal observed from analysis of the standard solution yielded a copper concentration of 760 ± 40 ppb at 95% confidence. This indicates that the method is applicable to samples of copper in aquarium systems, though further work should be performed to improve the method. Determination of copper in marine systems in the presence of cadmium and lead was not performed, because for the intended application (in professional aquaria) these metals are not expected above trace concentrations.

Conclusions

Benchtop BPED methodologies present an opportunity for future research in the development of an *in situ* point of analysis method. This research sought to determine if it was possible to develop a quantitative method for the analysis of copper in freshwater at gold and a method for analysis of copper in marine water at glassy carbon electrodes. The LOD for the benchtop method on Au was 5 ppb and the LOQ was 20 ppb; however, for the glassy carbon method the LOD was estimated at 20 ppb Cu^{2+} and the LOQ was 50 ppb. Finally, both methods gave a linear response, with a linear least-squares regression coefficient (R^2) > 0.99 for concentrations from their LOQ to 75 ppb on Au electrodes, and 3.50×10^3 ppb on glassy carbon respectively.

The most challenging aspect of this research, was attempting to overcome the voluminous amount of calculations needed to optimize the waveforms. To that end, multiple functions were written for the MATLAB™ computing environment which served to compress the time required for analysis from several days per parameter, to only several minutes. The full repository of code can be found in Appendix A. Further work should explore the selectivity of the marine BPED method against cadmium and lead, which may interfere if present above the expected trace concentration levels in marine aquariums.

Chapter 5: Summary, Impact, and Future Directions

Summary and Overall Impact

Copper has long history of use in dietary supplements, as an antifungal and antimicrobial agent as a biocidal paint additive, and in aquaria as a treatment for the protozoan *Cryptocaryans irritans*, a.k.a. “ich”. While the literature is replete with methods for the analysis of copper, most methods are focused on concentrations above the therapy level for marine aquaria of 180 to 220 ppb, or use highly specialized, and expensive equipment such as ICP-MS and ICP-OES.

This research has shown that BPED can be utilized after IC separation to perform the analysis of metals in aqueous samples. Additional work was done in developing benchtop BPED methodologies for copper on gold and glassy carbon electrodes. The development of the benchtop waveforms required significant amounts of calculations which, if done by hand, would have seriously delayed the analysis time. Since no commercial software was available, a series of functions and scripts were written in the MATLAB™ computing environment which were used extensively in expediting the analysis of the optimization data collected throughout this research.

Analysis by BPED after ion chromatography was successfully applied to determining copper levels in various samples including a blind sample, and multiple dietary supplements using a gold working electrode. Copper was quantifiable in concentrations from 50 ppb to 1.00×10^3 ppb, with the linear least squares regression coefficient (R^2) > 0.999 and a RSD of 3%. The method was successfully applied to the

analysis of copper in a liquid dietary supplement, as well as a glycinated chelate of copper in a tablet form of supplement with both samples matching their bottle labels. Additionally, a comparison copper concentration results between the liquid supplement bottle label, UV-Vis analysis of the liquid supplement, and IC-BPED analysis of the liquid supplement using a Student's t-test showed that the IC-BPED method gave the same population as the UV-Vis method; both methods matched the bottle label and the manufacturer's production protocols. A third supplement gave a concentration that was higher than expected during analysis, and was unable to be successfully compared to the bottle label. Finally, the IC-BPED method was used to determine the concentration of copper in laboratory tap water, and the water was found to contain between 580 to 700 ppb copper. Additionally, the method could quantify both lead and cadmium, metals which are of toxicological interest. Both lead and cadmium showed linear responses in the regions of 50 ppb to 600 ppb Cd and 200 ppb to 5.00×10^3 ppb Pb with linear least squares regression coefficients (R^2) > 0.99 and RSDs of 4% for both analytes.

The presented IC-BPED method has applications in routine analyses of freshwater samples for copper, cadmium and lead. Additional work can be done on the analysis of dietary supplements of copper, providing an alternative avenue for QC and QA measurements in these supplements.

The IC-BPED method using Au electrodes is limited to samples that do not contain high levels of chloride anion, a common contaminant, as the gold working electrode is inherently incompatible with this ion. This can be overcome by using a

silver cartridge to remove the chloride on-line with the sample injection, or through use of alternative electrode materials. The use of glassy carbon was investigated to overcome this limitation.

As an alternative to using a silver cartridge, glassy carbon was investigated as an alternative working electrode material. Negative signals were observed for copper that had a weak linear response (linear least squares regression coefficient, $R^2 = 0.95$) in the concentration region of 2.00 to 10.0 ppm. The IC-BPED method on glassy carbon is limited by the lack of sensitivity, but may be useful in certain industrial situations. Additionally, cadmium could be detected using glassy carbon as the working electrode material, however the response was not quantifiable at the concentrations investigated of 2.00 ppm to 10.0 ppm. Additionally, lead was not detected at all when using the glassy carbon working electrode on the IC-BPED method. Despite these setbacks, the method may be useful for determination of copper levels in situations where the contamination is expected to be extreme.

Based on observations made during the development of the IC-BPED method, further investigations were carried out on developing a benchtop BPED method for copper using Au electrodes. Optimization of this BPED waveform required the creation of a significant amount of computer software to expedite analysis of the data. This code is found in Appendix A. Analysis of analytical figures of merit for this method showed a LOQ of 20 ppb, and LOL of 75 ppb. Still, this method is limited by the incompatibility of Au electrodes with chloride solutions.

To overcome this limitation, glassy carbon was investigated as an alternate electrode material. This method used artificial seawater as the supporting electrolyte, with no chemical modifications or additions. After performing voltammetric analyses, initial waveform parameters were chosen and these initial parameters were applied to a series of standard additions of copper to artificial seawater. The resulting analytical figures of merit showed a LOD of 20 ppb and LOQ of 50 ppb. Analysis of a standard solution of copper in aquarium artificial seawater shows that this method is poised for application to the direct analysis of copper in marine waters at the point of analysis. This *in situ* sensor should be the focus of future work for this research.

Future Directions

The major future direction for this research is in the development of an *in situ* analysis platform that can be deployed at the point of analysis and can provide constant monitoring of copper concentration (or other metals of interest if the waveform is appropriately adapted). This immediate direction would require the validation and examination of selectivity for the reported methods in chapter 4 along with the development and fabrication of the *in situ* sensor itself.

This is easily accomplished by placing a peristaltic pump in-line with an aquarium tank and flowing the solution through a commercially available Dionex ED-40 flow cell, such as the one used in the IC-BPED portion of this work. Should the peristaltic pump exhibit flow rate issues, a similar pump such as a milligat could be utilized to provide steady flow of aquarium tank water. Further work would need to go

into the determination of prolonged running effects (such as working electrode fouling) and standard operating procedures would need to be developed so that non-specialized personnel could monitor the copper levels.

Development of this sensor would allow for constant monitoring of copper levels in marine tanks – a significant boon to the aquarium industry. Additionally, with a few modifications to the BPED waveform and subsequent optimization it would be possible to adapt this sensor to additional metals of interest in environmental and toxicological systems. Data from the IC-BPED analysis of metals shows that both Cd and Pb are readily detected at gold electrodes with BPED.

Potential applications for this work are limited only by the ability to discriminate between analyte species based upon their reduction potentials and upon the physical and chemical limitations of the working electrode materials chosen to work with. This research has shown that there is a great opportunity to develop these point of use sensors utilizing BPED analysis. Careful optimization of BPED waveforms is required to attain optimum performance; however, this task is made easier using computer software which automates the tasks of unifying data files, computing required figures, and outputting data into a usable format. Examples of such software can be found in Appendix A.

Appendix

Appendix A: MATLAB Code Repository

Introduction

This appendix contains a series of functions written for the MATLAB (Release v. 2012b – 2015b, Mathworks, Inc.) programming environment. Script files are sequences of commands which are run at the MATLAB command line, and variables produced are stored in the active MATLAB workspace – these types of files are used for rapid execution of common tasks that the user wishes to execute inside the normal MATLAB workspace. In contrast, functions are sequences of commands which only return variables in the function output declaration. Both function and script files have the extension *.m – a proprietary format for MATLAB; but store the information as plain text, and so can be edited in any standard text editing software (e.g. notepad). These *.m files do require the use of the MATLAB environment for their execution, and are not stand-alone executable files, though MATLAB does have the capability to generate stand-alone executables from completed functions. Function and script files are differentiable by examination of the first line of code. Functions start with the “function” declaration followed by the function output variables and the input arguments, script files lack this declaration. For additional information on the MATLAB environment, readers are referred to the most recent edition of Getting

Started with MATLAB: A Quick Introduction for Scientists and Engineers by Rudra Pratap.

Syntax of MATLAB Code and Code practices

This section seeks to present code in accordance with proper programming techniques (nesting, commenting, etc.). In the MATLAB™ environment, statements which begin with a percent symbol (%) are comment lines, and are not executed during program runtime. Additionally, statements are written using the practice of “camel case” where words begin with lowercase letters and intervening words start with capitals (e.g. reductionTime). Code statements should be terminated with the semi-colon (;) to suppress the output of the statement at the command line, statements left unterminated will have their output displayed in the MATLAB™ workspace. This can cause significant issues if there is a particularly large calculation taking place. Where functions call upon other functions – the functions called upon in the code must be placed in the active directory of MATLAB™. This can be easily accomplished by keeping all functions for a project in a single directory and ensuring that you run the *addpath* command at the start of your session, where the argument of the command being your project directory (i.e. `addpath('codeDir\');` where `codeDir` is the directory containing your files). Finally, Boolean operators such as AND, OR, and NOT, are given specific syntax within MATLAB™; AND is represented by a double ampersand (&&) when working with scalar values, and a single ampersand (&) when working with vectors, NOT is represented by preceding the variable name with a tilde (~), OR is performed using the bar/pipe character (|) placed between the two variables.

Every attempt is made to avoid certain the use of certain variable names in scripts (and in functions, but to a lesser extent) which are reserved for the MATLAB™ system, such as *i*, *pi*, etc. As a check, it is always considered good practice to run a variable name through the MATLAB™ *exist* function to determine if the name already exists inside the environment. Finally, variables have an inherent scope of being local (i.e. only viewable within the script or function being run) unless they are declared as global variables using the statement “global [variableName];”. Declaring a variable as global allows the variable to be accessed by multiple functions.

Since the code projects used were directed to analyze the data generated by the Pine Instruments WaveNow™ potentiostat, which outputs raw data as a two-column comma delimited (*.csv) text file with a single header row denoting the data contents and units (e.g. Time (s)), thus functions were written with instrument-specific code in mind. Additional functions were written to analyze data from the BAS, Inc. Epsilon-EC or 100-B/W potentiostats which output data as a two column ASCII format text file (*.txt extension), with a variable number of header lines, which also incorporate some instrument-specific functionality. Where instrument specific code has been written, comments will denote changes that can be made to make the functions more generalized. Finally, some functions were written to compliment data analysis tasks for specific experiments performed throughout this research – these functions are given as examples for how common problems can be resolved using MATLAB™ or a similar programming environment.

Format of this Code Repository

In this document, each function or script will be clearly marked and accompanied by a brief description of its intended purpose. When two functions are similar, this will be noted in the function description. This repository is not intended to be an exhaustive formulation of code, but a starting point for new and unique functions which may be adapted to serve similar purposes.

Function: pineDataImport.m

Purpose: Import data files collected with WaveNow potentiostat.

Code:

```
function [ dataOut ] = pineDataImport( startPath )
%PINEDATAIMPORT Imports I vs t data from Pine WaveNow instrument converted
%to a csv file.
%-----
% If no start path is specified by startPath, function will call uigetdir to find
% the root folder for data files, followed by a multi-select uigetfile to import
% the data from the *.csv files into matlab.
% Function will then zip them into a cohesive matrix (dataOut) of format
% x,y1,y2,y3,y4,...,yn for further processing by other functions/guis.
%
%-----Written 6/10/2014 by Ben Cunning - willcun1@umbc.edu -----

%Get start path using uigetdir only if one not specified during input (startPath)
if nargin < 1
    defaultPath = uigetdir;
else
    defaultPath = startPath;
end

%using path, perform uigetfile on multiselect – if multiple file types desired
%they can be added to the initial section of the uigetfile function
%for more info users should refer to: doc uigetfile
[fileNames,filePath] = uigetfile('*.csv','Please select data files',...
    defaultPath,'MultiSelect','on');

%Import the first data file, then initialize an empty matrix to store the
%'zipped' files for the data set, and count elements in fileNames to get
%end of loop variable.
```

```

importedData = dlmread(strcat(filePath,fileNames{1}),'',1,0);
[numRows,numCols] = size(importedData);
dataOut(numRows,numCols) = 0;
loopEnd = numel(fileNames);

%set the first two columns of dataOut equal to first data set, then use a
%for loop to finish the import and writing of y values to the matrix.
dataOut = importedData;
for loopCount = 2:loopEnd
    importedData = dlmread(strcat(filePath,fileNames{loopCount}),'',1,0);
    dataOut(:,loopCount+1) = importedData(:,2);
end

end

```

Function: pineAreaCalc.m

Purpose: Estimate area under the I vs. t curve using the trapezoid method, gives sum of the entire area under the curve based on a given delay time.

Code:

```

function [ calcArea ] = pineAreaCalc( dataIn,delTime )
%PINEAREACALC calculates trapezoidal area for data from Pine Instrument
%-----
% Function will call MATLAB function trapz to calculate areas of data
% passed to function. Output is matrix of areas (calcArea) with inputs of
% the raw data matrix (dataIn) and desired delay time (delTime).
%-----Written 6/24/2014 by Ben Cunning - willcun1@umbc.edu -----

%if # input Args < 2, prompt for delTime, else use delTime to cut
%appropriate # of rows from matrix for processing, int32 function used to
%ensure that the string is properly converted to integer for use by function
if nargin < 2
    delTimeStr = inputdlg('Please enter the delay time in milliseconds.','...
    'Delay Time Input');
    delTime = int32(str2double(delTimeStr));
end

%Calculate time interval per data point
timeInt = dataIn(2,1)-dataIn(1,1);
%calculate rows to cut
cutRows = delTime/timeInt;
if cutRows > 0
    %cut delay time out

```

```

        dataInDel = dataIn(cutRows:end,:);
    else
        dataInDel = dataIn;
    end
    %Calculate Areas
    calcArea = trapz(dataInDel(:,1),dataInDel(:,2));

end

```

Function: pineDataExport.m

Purpose: Function served to write data as an output to a *.csv format file

Code:

```

function [ ] = pineDataExport( dataToExport,outPath,outName )
%PINEDATAEXPORT Function writes data to output to a file
%-----
% Function will write input data (dataToExport) to *.csv file.
% function can accept a start path (outPath), and/or file name (outName)
% passed to it by external function.
%-----Written 6/24/2014 by Ben Cunning - willcun1@umbc.edu -----

%if both path and name pre-specified, jump to export
%check for existing path and filename, if neither do following:
if not(exist('outPath','var')) && not(exist('outName','var'))
    %get file name for output
    [outName,outPath] = uiputfile('*.csv','Save Data File As...');
end

%if only name missing do following
if exist('outPath','var') && not(exist('outName','var'))
    %get file name for output
    [outName,~] = uiputfile('*.csv','Save Data File As...','outPath,...
        'DefaultName');
end

%if only path missing do following
if not(exist('outPath','var')) && exist('outName','var')
    %get file name for output
    outPath = uigetdir();
end

%write data to file for output
dlmwrite(strcat(outPath,outName),dataToExport,',' );

```

end

Function: pineMultiDataImport.m

Purpose: Expedite import of multiple data sets into the workspace

Code:

```
function [ dataArray ] = pineMultiDataImport( numSets,startPath )
%PINEMULTIDATAIMPORT calls pineDataImport function for multiple data sets
%-----
% Function calls pineDataImport for a set number of times based on the
% variable numSets given by user (i.e. 5 times if numSets = 5, etc.) using the
% given starting directory (startPath) as a "home".
% Data is output in a cell array (dataArray) for call by later functions/guis.
%-----Written 6/10/2014 by Ben Cunning - willcun1@umbc.edu -----
doesDirExist = exist(startPath,'dir');
%get primary start path to pass to pineDataImport unless already specified
if or(nargin < 2, doesDirExist == 0)
    startPath = uigetdir;
end

%Initialize data array
dataArray = cell(numSets,1);

%set for loop for importing data sets
for importCount = 1:numSets
    dataArray{importCount}=pineDataImport(startPath);
    %display loop iteration
    if importCount < numSets
        str = strcat('Loop_',num2str(importCount),...
            ' complete, proceeding to next. ');
        disp(str)
    end
end
end

end
```

Function: pineMultiDataSetImport.m

Purpose: Imports multiple data sets into a data cell array of matrices.

Code:

```
function [ dataArray ] = pineMultiDataSetImport( numFilesPerSet,numSets )
%PINEMULTIDATASETIMPORT Imports multiple data sets into a data array
%-----
% Function calls pineDataImport for a set number of times based on the
% variable numSets given by user (i.e. 5 times if numSets = 5, etc.) with the
```

```

% provided number of files per data set (numFilesPerSet, triplicate = 3)
% Data is output in a cell array (dataArray) for call by later functions/guis.
%-----Written 6/10/2014 by Ben Cuning - willcun1@umbc.edu -----

defaultPath = uigetdir;

%Initialize data array
dataArray = cell(numSets,1);
dataSet = cell(numFilesPerSet,1);

%using path, perform uigetfile on multiselect
[fileNames,filePath] = uigetfile('*.csv','Please select data files',...
    defaultPath,'MultiSelect','on');

%Set dataSet to be the filenames that are included in numFilesPerSet
for setLoop = 1:numSets
    a = (((setLoop-1)*numFilesPerSet)+1);
    b = (setLoop*numFilesPerSet);

    for nameLoop = a:b
        c = nameLoop-(numFilesPerSet*(setLoop-1));
        dataSet{c,1} = fileNames{nameLoop};
    end

    %Import the first data file, then initialize an empty matrix to store the
    %'zipped' files for the data set, and count elements in fileNames to get
    %end of loop variable
    importedData = dlmread(strcat(filePath,dataSet{1}),'',1,0);
    [numRows,numCols] = size(importedData);
    dataOut = zeros(numRows,numCols+(numFilesPerSet-1));
    loopEnd = numel(dataSet);

    %set the first two columns of dataOut equal to first data set, then use a
    %for loop to finish the import and writing of y values to the matrix.
    dataOut(:,1:2) = importedData;
    for loopCount = 2:loopEnd
        importedData = dlmread(strcat(filePath,dataSet{loopCount}),'',1,0);
        dataOut(:,loopCount+1) = importedData(:,2);
    end
    % Write dataOut to array position
    dataArray{setLoop,1} = dataOut;
end

```

end

Function: pineTextZipper.m

Purpose: Function reads in multiple text files, while eliminating repeated time data columns from data matrix to produce a matrix for use in other functions.

Code:

```
function [ dataOut ] = pineTextZipper( )
%PINETEXTZIPPER Combines multiple text files into a single matrix
% Function reads in individual text files, removes repeated x column
% (time data) and adds the non-redundant y-data to a new matrix (dataOut).
```

```
%get the primary data directory using GUI
defaultPath = uigetdir;
```

```
%using path, perform uigetfile on multiselect
[fileNames,filePath] = uigetfile('*.csv','Please select data files',...
    defaultPath,'MultiSelect','on');
```

```
%Import the first data file, then initialize an empty matrix to store the
%'zipped' files for the data set, and count elements in fileNames to get
%end of loop variable.
```

```
importedData = dlmread(strcat(filePath,fileNames{1}),'',1,0);
[numRows,numCols] = size(importedData);
dataOut(numRows,numCols) = 0;
loopEnd = numel(fileNames);
```

```
%set the first two columns of dataOut equal to first data set, then use a
%for loop to finish the import and writing of y values to the matrix.
dataOut = importedData;
for loopCount = 2:loopEnd
    importedData = dlmread(strcat(filePath,fileNames{loopCount}),'',1,0);
    dataOut(:,loopCount+1) = importedData(:,2);
end
```

end

Function: pineMultiAreaCalc.m

Purpose: Function calculates the trapezoidal area under the curve for multiple data sets.

Code:

```
function [ calcAreas ] = pineMultiAreaCalc( dataIn,deltaTime )
%PINEMULTIAREACALC calculates multiple areas for data from Pine Instrument
%-----
```



```

% Function will call MATLAB function trapz to calculate areas of data
% passed to function. Function will parse data differently to
% sub-function pineAreaCalc depending on if the data input (dataIn) is a cell
% array or matrix. Output will be a matrix (calcAreas) with 1 row per set.
% Function will pass desired delay time (delTime) to sub function.
%-----Written 6/24/2014 by Ben Cuning - willcun1@umbc.edu -----

%if # input Args < 2, prompt for delTime, else use delTime to cut
%appropriate # of rows from matrix for processing
if not(exist('delTime','var'))
    delTimeStr = inputdlg('Please enter the delay time in milliseconds.','...
    'Delay Time Input');
    delTime = str2double(delTimeStr);
end

cellFlag = iscell(dataIn);

%If dataIn is a cell array the following will execute:
if cellFlag == 1
    %Find number of cell elements in dataIn - this is the number of data
    %sets to be processed
    dataSets = numel(dataIn);

    %using for loop, extract a matrix for further processing
    for setCount = 1:dataSets
        calcMat = dataIn{setCount};
        xData = calcMat(:,1);
        yData = calcMat(:,2:end);
        [~,numCurves] = size(yData);
        %initialize output matrix on 1st iteration
        if setCount == 1
            calcAreas = zeros(dataSets,numCurves);
        end
        %Get numCurves in matrix, initialize area storage matrix
        [~,numCurves] = size(yData);
        areasMat = zeros(numCurves,1);
        %using another for loop assemble and send data to sub-function to
        %get area matrix. This can slow processing – optimization needed.
        for areaLoop = 1:numCurves
            %Assemble data to send
            sendDat = [xData,yData(:,areaLoop)];
            %get areas
            areasMat(areaLoop,1) = pineAreaCalc(sendDat,delTime);
        end
    end
end

```

```

        end
        % Write areasMat to output array
        calcAreas(setCount,:) = areasMat';
    end
else
    %initialize calcAreas to matrix of dimension [numCurves,2]
    [~,numCols] = size(dataIn);
    numCurves = numCols-1;
    calcAreas = zeros(numCurves,1);
    %strip x values from dataInput
    xData = dataIn(:,1);
    %pull y values from dataInput
    yData = dataIn(:,2:end);
    %get areas by calling sub-function in for loop
    for loopCount = 1:numCurves
        %calcAreas(loopCount,2) = loopCount; - useless for now
        calcAreas(loopCount,1) = pineAreaCalc([xData,yData(:,loopCount)]...
            ,delTime);
    end
end
end

end

```

Function: CellCalculator.m

Purpose: Calculates average current, area and standard deviation of area for data from a cell array.

Code:

```

function [ CurCell,TpzCell ] = CellCalculator( dCell,nRs,nS )
%CELLCALCULATOR Takes input cell and outputs avg I, A and Std(A).
% Made to assist in optimization. dCell is input cell array of matrices
% in format x|y1-1...yn-1|y1-2...yn-2|...; nRs is number of runs per data
% set (e.g. for triplicate runs enter 3); nS is the number of data sets
% collected (e.g. for 10 potentials enter 10).

%preallocate output cells
nC = numel(dCell);
CurCell = cell(nC,1);
TpzCell = cell(nC,1);
for cO = 1:nC
    %get data from cell array
    mData = dCell{cO};
    %eliminate x's
    mData(:,1) = [];

```

```

[r,~] = size(mData);
    %preallocate variables
    AvgI = zeros(r,nS);
    tOut = zeros(nS,2);
    data = zeros(r,nRs);
    %calculate nS times the mean currents, traps and std traps for each nRs
    for ceS = 1:nS
        %build working dataset
        for dBuild = 1:nRs
            col = int32((((ceS-1)+1)+((dBuild-1)*nS)));
            data(:,dBuild) = mData(:,col);
        end
        %get average currents
        dMean(:,1) = mean(data,2);
        %average traps
        dMnInt = mean(trapz(data(5:25,:)));
        %std traps
        dStdInt = std(trapz(data(5:25,:)));
        %Output to matrices
        AvgI(:,ceS) = dMean; clear dMean;
        tOut(ceS,1) = dMnInt; clear dMnInt;
        tOut(ceS,2) = dStdInt; clear dStdInt;
    end
    clear data;
    CurCell{cO} = AvgI; clear AvgI;
    TpzCell{cO} = tOut; clear tOut;
end

end

```

Function: CellCalculatorTox.m

Purpose: Similar to CellCalculator.m, this function will perform a similar function while assisting in waveform parameter optimization of t_{del} and t_{int} .

Code:

```

function [ CurCell,TpzCell ] = CellCalculatorTox( dCell,nS,tdel,tint )
%CELLCALCULATOR Takes input cell and outputs avg I, A and Std(A).
% Made to assist in optimization. dCell is input cell array of matrices
% in format x|y1-1...yn-1|y1-2...yn-2|...; nRs is number of runs per data
% set (e.g. for triplicate runs enter 3); nS is the number of data sets
% collected (e.g. for 10 potentials enter 10). tdel and tint are the
% delay and integration times in ms respectively. If optimizing these
% values, do not enter them.

```

```

%preallocate output cells
nC = numel(dCell);
CurCell = cell(nC,1);
TpzCell = cell(nC,1);
if ~exist('tdel','var')
    tdel = 'a';
end
if ~exist('tint','var')
    tint = 'a';
end
%if tdel = 0 ms, set to start for row 1 in all later calculations.
if tdel == 0
    tdel = 1;
end
for cO = 1:nC
    %get data from cell array
    mData = dCell{cO};
    %eliminate x's
    mData(:,1) = [];
    [r,~] = size(mData);
    %preallocate variables
    AvgI = zeros(r,nS);
    tOut = zeros(nS,2);
    if tint ~= 'a' && tdel > 0
        %calculate nS times the mean currents, traps and std traps for each run
        for ceS = 1:nS
            %get average currents
            dMean(:,1) = mean(mData,2);
            %average traps
            dMnInt = mean(trapz(mData(tdel:tdel+tint,:)));
            %std traps
            dStdInt = std(trapz(mData(tdel:tdel+tint,:)));
            %Output to matrices
            AvgI(:,ceS) = dMean; clear dMean;
            tOut(ceS,1) = dMnInt; clear dMnInt;
            tOut(ceS,2) = dStdInt; clear dStdInt;
        end
    end
end
%if tdel not specified, then optimize tdel assuming no optimum tint
if tdel == 'a' && tint == 'a'
    for count = 1:r-1
        trap = trapz(mData(count:end,:));
        tOut(count,1) = mean(trap);
    end
end

```

```

        tOut(count,2) = std(trap);
    end
end
%if tdel given, and tint not, then optimize tint
if tint == 'a' && tdel > 0
    for count = 1:(r-tdel)
        trap = trapz(mData(tdel:tdel+count));
        tOut(count,1) = mean(trap);
        tOut(count,2) = std(trap);
    end
end
CurCell{cO} = AvgI; clear AvgI;
TpzCell{cO} = tOut; clear tOut;
end

end

```

Function: pineTextZipAndDump.m

Purpose: This function will import a series of data files, “zip” them together similar to pineTextZipper.m, and then write them to a specified output file.

Code:

```

function [ fileNames ] = pineTextZipAndDump( )
%PINETEXTZIPANDDUMP Import/Export file function
% Function imports files, zips them, and then exports a single *.csv file.
% Function optionally returns the list of file names (fileNames) in a cell array.

%Get startpath using uigetdir only if one not specified in input
if nargin < 1
    defaultPath = uigetdir;
else
    defaultPath = startPath;
end

%using path, perform uigetfile on multiselect
[fileNames,filePath] = uigetfile('*.csv','Please select data files',...
    defaultPath,'MultiSelect','on');

%Import the first data file, then initialize an empty matrix to store the
%'zipped' files for the data set, and count elements in fileNames to get
%end of loop variable.

importedData = dlmread(strcat(filePath,fileNames{1}),',',1,0);
[numRows,numCols] = size(importedData);

```

```

dataOut(numRows,numCols) = 0;
loopEnd = numel(fileName);

%set the first two columns of dataOut equal to first data set, then use a
%for loop to finish the import and writing of y values to the matrix.
dataOut = importedData;
for loopCount = 2:loopEnd
    importedData = dlmread(strcat(filePath,fileName{loopCount}),'',1,0);
    dataOut(:,loopCount+1) = importedData(:,2);
end

%get outPath and outName
[outPath,outName] = uiputfile('*.csv');
%write file out
dlmwrite(strcat(outName,outPath),dataOut);
end

```

Function: current2areaConverter.m

Purpose: Convert current data to area data row-by-row

Code:

```

function [ areaMatrix ] = current2areaConverter( currentMatrix )
%CURRENT2AREACONVERTER Converts current matrix to area matrix by row
% Written 2/25/2015 by Ben Cunning - willcun1@umbc.edu: Function is
% meant to be a sub-function that will calculate the trapezoidal area
% beneath an I vs. t curve row-by-row vs. giving a summated area for the
% entire length of time. Output will be matrix of trapezoidal areas (areaMatrix), to
% finish integration, user need only sum the appropriate rows.

```

```

[numRows, numCols] = size(currentMatrix);
areaMatrix = zeros(numRows-1,numCols);

for trapCol = 1:numCols
    for trapRow = 1:numRows-1
        areaMatrix(trapRow,trapCol) = trapz(...
            currentMatrix(trapRow:trapRow+1,trapCol));
    end
end

end

```

Function: pineDataAnal.m

Purpose: A summary function used to call several sub-functions to fully analyze a set of data.

Code:

```

function [ ] = pineDataAnal( numFilesPerSet,numSets )
% PINEDATAANAL Is the global function for analyzing Pine instrument data
%-----
% This function will call several sub functions to import, perform
% analysis on, and export various pine instruments data. Requires
% numSets (number of data sets) to be put in so that it can decide between
% singular and multiimport algorithms, etc. with numFilesPerSet equal to
% the number of replicate measurements in each data set (e.g. triplicate = 3).
%-----Written 6/26/2014 by Ben Cuning - willcun1@umbc.edu -----

%check for import of multiple data sets
if numSets == 1
    %Find and Fetch raw Data
    importedData = pineDataImport;
    %Send data to area analyzer
    areasOut = pineAreaCalc(importedData);
    %Send areas to file for writing
    pineDataExport(areasOut);
else
    %Get Raw data SETS
    importedData = pineMultiDataSetImport(numFilesPerSet,numSets);
    %Send data to area calculator
    areasOut = pineMultiAreaCalc(importedData);
    %Send areas to file for writing
    pineDataExport(areasOut);
end

end

```

Function: pineWaveOptimizer.m

Purpose: The purpose of this function is to assist in the optimization of BPED waveform parameters.

Code:

```

function [ ] = pineWaveOptimizer(
glassyCarbonFlag,rows2skip,numDataSets,numCurvesPerSet,numBlankSets)
% PINEWAVEOPTIMIZER function assists in optimization of BPED Waveforms
%-----
% This function will import data collected in an optimization method and
% assist in calculation of S/N data for selection of optimal parameters.
% The output of this function will be a *.csv format file which can be
% put into a graphical software for analysis of S/N curves. The
% function will subtract background current from the copper bearing

```

```
% signal then calculate the area based on the trapezoid method. For
% glassy carbon an additional delay time to compensate for the slow slew
% rate of the Pine WaveNow as it moves to the oxidation potential
% (approx. 2 ms) which will be accounted for while importing raw data
% files. glassyCarbonFlag should be set to 1 for glassy carbon
% electrode data or 0 for gold electrode data. rows2skip is the delay time,
% numDataSets is the number of concentration sets to analyze with
% total number of measurements equal to numCurvesPerSet, and
% the total number of blank measurements equal to numBlankSets
%-----Written 8 Sep. 2014 by Ben Cuning - willcun1@umbc.edu-----
```

```
%Import files - can be a zipped Matrix since all collection is usually a
%long tox during optimization
importedData = pineTextZipper;
%delete the first 2 ms from copper arrays
if glassyCarbonFlag == 1
    dataMat = importedData(rows2skip:end,:);
    %if not glassy carbon, just write to the storage matrix
else
    dataMat = importedData;
end
%Get the size of this matrix
[rows,~] = size(dataMat);
%get x data for way later when ouput requires times
timeDat = dataMat(:,1);
%Set blanks array and copper array - y data only!
blankDat = dataMat(:,2:(numBlankSets*numCurvesPerSet)+1);
copperDat = dataMat(:,(numBlankSets*numCurvesPerSet)+2:end);
%figure out how many sets of copper data there are store as int32 should
%give the number of copper concentrations studied
numCopSets = int32((numDataSets-numBlankSets)/numBlankSets);
%Get average current for the blanks and the corresponding st'd dev'ns, then
%get areas, average areas and std dev areas (maybe get propogated errors
%from current as well for comparison?)
%pre-allocate loop variable
blkAvgI = zeros(rows,numBlankSets);
blkStdI = zeros(rows,numBlankSets);
for AvgILp = 1:numBlankSets
    %for convenience let's calculate the start and end indices for the
    %blank curves to average - use int32 since they must be integers
    firstCurve = int32(1+((AvgILp-1)*numCurvesPerSet));
    lastCurve = int32(numCurvesPerSet+((AvgILp-1)*numCurvesPerSet));
    %now let's get some means and write them to the output matrix
```



```

    blkAvgI(:,AvgILp) = mean(blankDat(:,firstCurve:lastCurve),2);
    %now st'd deviations
    blkStdI(:,AvgILp) = std(blankDat(:,firstCurve:lastCurve),1,2);
end
%lather, rinse, repeat for Cu
CuAvgI = zeros(rows,numCopSets*numBlankSets);
CuStdI = zeros(rows,numCopSets*numBlankSets);
blankCorrected = zeros(rows,numBlankSets*numCurvesPerSet*numCopSets);
for CuConcLp = 1:numCopSets
    %calculate indices for data set
    firstCopper = int32(1+((CuConcLp-1)*numBlankSets*numCurvesPerSet));
    lastCopper = int32((numBlankSets*numCurvesPerSet)...
        +((CuConcLp-1)*(numBlankSets*numCurvesPerSet)));
    %set data to be corrected by concentration set
    calcData = copperDat(:,firstCopper:lastCopper);
    %do blank correction - this is harder than it seems because there
    %are 3 coppers for every 1 blank average
    %set up so only 1 "set" is being pulled
    for blkSubLp = 1:numBlankSets
        %Do Correction
        for subLp = 1:numCurvesPerSet
            %Gotta pull the right values for individual curves
            copIndex = int32((1+((blkSubLp-1)*numCurvesPerSet))+(subLp-1));
            blankCorrected(:,(copIndex+((CuConcLp-1)*numCurvesPerSet*...
                numBlankSets))) = calcData(:,copIndex)...
                - blkAvgI(:,blkSubLp);
        end
        %get averages - put in appropriate place
        CuAvgI(:,(blkSubLp+((CuConcLp-1)*numBlankSets))) = mean(...
            blankCorrected(:,(1+(numCurvesPerSet*(blkSubLp-1))):...
                (numCurvesPerSet+(numCurvesPerSet*(blkSubLp-1)))),2);
        %get std Deviations - put in place
        CuStdI(:,(blkSubLp+((CuConcLp-1)*numBlankSets))) = ...
            std(blankCorrected(:,(1+(numCurvesPerSet*(blkSubLp-1))):...
                (numCurvesPerSet+(numCurvesPerSet*(blkSubLp-1)))),0,2);
    end
end
%use stds to propagate error for area calculations to compare to
%area std devs. The propagated area is the square root of the
%sum of the squares of the sum of ((sqrt(std(y1)^2+std(y2)^2)/2)^2. Which
%is a hot mess to describe.
%Get the variances (i.e. std^2)

```

```

blkVarI = blkStdI.^2;
CuVarI = CuStdI.^2;
%now I have to add elements together progressively from the variances
%within a column - time for a for loop; pre-allocate a storage variable for
%the square roots of the sums of variances
blkSqrtSumVarI = zeros(rows-1,numBlankSets);
cuSqrtSumVarI = zeros(rows-1,numBlankSets*numCopSets);
for varSumLp = 1:rows-1
    blkSqrtSumVarI(varSumLp,:) = ...
        (sqrt(blkVarI(varSumLp,:)+blkVarI(varSumLp+1,:))./2).^2;
    cuSqrtSumVarI(varSumLp,:) = ...
        (sqrt(CuVarI(varSumLp,:)+CuVarI(varSumLp+1,:))./2).^2;
end
%get sqrt of sums of that hot mess above to get final propagations
blkPropEr = sqrt(sum(blkSqrtSumVarI));
CuPropEr = sqrt(sum(cuSqrtSumVarI));

%Get Areas
areasVec = pineMultiAreaCalc(dataMat,0);
%now need to calculate averages and standard deviations for blanks and
%coppers - yet another for loop, start by preallocating variables
avgArea = zeros(numDataSets,1);
stdArea = zeros(numDataSets,1);
for areaLp = 1:numDataSets
    %Calc indices
    firstArea = int32(1+(numCurvesPerSet*(areaLp-1)));
    lastArea = int32(numCurvesPerSet+(numCurvesPerSet*(areaLp-1)));
    %Calc Avgs
    avgArea(areaLp,1) = mean(areasVec(firstArea:lastArea,1));
    %Calc Stds
    stdArea(areaLp,1) = std(areasVec(firstArea:lastArea,1));
end
%assemble area output matrix
blkArDat(:,1) = avgArea(1:numBlankSets,1)';
blkArDat(:,2) = stdArea(1:numBlankSets,1)';
blkArDat(:,3) = blkPropEr';
%pre-allocate copper storage matrix to store copper areas and copper stds.
copArDat = zeros(numBlankSets,(numCopSets*3));
for copDatAsmby = 1:numCopSets
    copArDat(:,(1+(3*(copDatAsmby-1)))) = ...
        avgArea(1+(numBlankSets*copDatAsmby):...
            numBlankSets*(1+copDatAsmby),1);
    copArDat(:,(2+(3*(copDatAsmby-1)))) = stdArea(1+...

```

```

        (numBlankSets*copDatAsmby):numBlankSets*(1+copDatAsmby),1);
    copArDat(:,(3+(3*(copDatAsmby-1)))) = CuPropEr...
        (1,(1+(numBlankSets*(copDatAsmby-1)))):...
        numBlankSets*(1+(copDatAsmby-1)));
end
areaOut = [blkArDat,copArDat];
%-----
%-----Data Output Section-----

%Initialize some arrays to use as column and row titles for a table - just
%because it would be nice to put out a couple of *.csv files that are ready
%to graph
areaParameterValues = cell(numBlankSets,1);
areaParamValsMat = zeros(numBlankSets,1);
%get the parameter Values
for paramInput = 1:numBlankSets
    loopNum = num2str(paramInput);
    prompt = strcat('Please input parameter_',loopNum);
    areaParameterValues{paramInput,1} = ...
        inputdlg(prompt,'Parameter Input...');
    areaParamValsMat(paramInput,1) = str2double(...
        cell2mat(areaParameterValues{paramInput,1}));
end
%now make the area column titles
areaColumnTitles{(numCopSets*3)+4} = [];
areaColumnTitles{1} = cell2mat(inputdlg('Parameter being optimized?',...
    'Parameter???'));
%Always put 'Blanks' as 2nd column title and BlankStd as 3rd and Blank prop
%error as 4th
areaColumnTitles{2} = 'Blanks';
areaColumnTitles{3} = 'Blank Standard Deviation';
areaColumnTitles{4} = 'Blank Propagated Error';
%now to do copper names
for colNameLp = 1:numCopSets
    areaColumnTitles{5+((colNameLp-1)*3)} = copCon{colNameLp};
    areaColumnTitles{6+((colNameLp-1)*3)} = ...
        strcat(copCon{colNameLp},' Standard Deviations');
    areaColumnTitles{7+((colNameLp-1)*3)} = ...
        strcat(copCon{colNameLp},' Propagated Error');
end

%Now to assemble the output matrix for the area data, should mimic the

```

```

%format of the area row vector for column names:
%i.e.
%Parameter|BlankAreas|BlankStdDev|BlankPropError|CuAreas|CuStd|CuPropE|etc.
areaOutDat = [areaParamValsMat,areaOut];

%For currents, the file table should look like this:
%Time|BlankCurrent@Param1|BlankStd@Param1|Cu#x@Param1|...
%Cu#xStd@param1|... so
%we should only need column titles - which should be assemblable without
%any futher user input could probably use this to assemble the output
%matrix as well...
%column 1 title should always be time so pre-allocate name cell vector and
%set cell 1 = Time; also pre-allocate output matrix and set column 1 = time
%data.
Ioutput = zeros(rows,((numBlankSets*2)+(numCopSets*2)));
Ioutput(:,1) = timeDat(:,1);
%calculate for loop increment index to make coding the loop simple
InmLpInd = 2+(numCopSets*2);
for InmLp = 1:numBlankSets
    %Calculate indicies of each name to input for blank and blank std
    blkIndex = 2+((InmLp-1)*InmLpInd);
    blkStdInd = 3+((InmLp-1)*InmLpInd);
    %put in blank data into data matrix
    Ioutput(:,blkIndex) = blkAvgI(:,InmLp);
    Ioutput(:,blkStdInd) = blkStdI(:,InmLp);
    %use sub loop to fill in copper names and data
    for copSubLp = 1:numCopSets
        %calculate copper indicies
        copInd = 4+(2*(copSubLp-1))+((InmLp-1)*InmLpInd);
        copStdInd = 5+(2*(copSubLp-1))+((InmLp-1)*InmLpInd);
        %put copper data into storage matrix
        Ioutput(:,copInd) = CuAvgI(:,InmLp);
        Ioutput(:,copStdInd) = CuStdI(:,InmLp);
    end
end
%clear all those loop indicies
clear blkIndex blkStdInd copInd copStdInd copSubLp InmLp;

%get output names and paths for current and area data
pathOut = uigetdir('C:\Users\Terciel\Documents\Research',...
    'Select Output Path');
[InameOut,Ipath]=uiputfile('*.csv','Save Current Data As...',...
    strcat(pathOut,'\CurrentOutput.csv'));

```

```
[AnameOut,Apath] = uiputfile('*.csv','Save Area Data As...','...
    strcat(pathOut,'\AreaOutput.csv'));
dlmwrite(strcat(Ipath,InameOut),Ioutput);
dlmwrite(strcat(Apath,AnameOut),areaOutDat);

end
```

Function: areaErrorProp.m

Purpose: This function was written to propagate errors in calculating the area under the curve using the trapezoid method for I vs. t curves.

Code:

```
function [ areaErrors ] = areaErrorProp( currentErrors )
%AREAERRORPROP Propagates errors for trapezoidal areas
%   Written 2/25/2015 by Ben Cuning - willcun1@umbc.edu: Function is
%   meant to assist in error propagation by row of individual areas. In
%   this way the error can be accurately propagated throughout the data
%   handling. currentErrors should be a matrix of blank corrected errors
%   in average current values. areaErrors will be a matrix of the
%   propagated errors when trapezoidal integral estimation is performed.

%Get size of current matrix, preallocate area matrix
[numRows, numCols] = size(currentErrors);
areaErrors = zeros(numRows-1,numCols);
%calculate variances
currentVariances = currentErrors.^2;

%in loop, perform calculations by row and column (row loop nested in col loop)
for propCol = 1:numCols
    for propRow = 1:numRows-1
        areaErrors(propRow,propCol) = sqrt(...
            currentVariances(propRow,propCol)+...
            currentVariances(propRow+1,propCol));
    end
end

end
```

Function: blankICxn.m

Purpose: Function was written to correct average current data for background (blank) currents and propagate errors from this process

Code:

```
function [ corIs,corErs ] = blankICxn( blankCurrentAvg,blankError,
dataAvgCurrents, dataErrors )
```

```
%BLANKICXN Corrects Avgcurrents for average background & propagates errors
% Written 2/25/2015 by Ben Cuning - willcun1@umbc.edu: Function will
% correct average signal current data (dataAvgCurrents) for the average blank
% signal (blankCurrentAvg) with standard deviations of dataErrors and blankError
% respectively.
% Function will then propagate the error for these corrections, and output the
% corrected current (corIs) and propagated error (corErs).
```

```
%get size of average current matrix, preallocate outputs
```

```
[sigRows,sigCols] = size(dataAvgCurrents);
```

```
corIs = zeros(sigRows,sigCols);
```

```
corErs = zeros(sigRows,sigCols);
```

```
%calculate variances
```

```
dataVariance = dataErrors.^2;
```

```
blankVariance = blankError.^2;
```

```
%calculate errors in for loops
```

```
for corRow = 1:sigRows
```

```
    for corCol = 1:sigCols
```

```
        corIs(corRow,corCol) = ...
```

```
            dataAvgCurrents(corRow,corCol)-blankCurrentAvg(corRow,1);
```

```
        corErs(corRow,corCol) = sqrt(dataVariance(corRow,corCol)...
```

```
            +blankVariance(corRow,1));
```

```
    end
```

```
end
```

```
end
```

Function: CurrentAverager.m

Purpose: Calculate average and standard deviation of current matrices passed to the function – primarily used in generating average and standard deviation of currents from cyclic voltammetry runs. Written for use in a larger function.

Code:

```
function [ avgCurrent,currentError ] = CurrentAverager( currentMatrix )
```

```
%CURRENTAVERAGER Calculates the average of currents and std. deviations
```

```
% Written 2/25/2015 by Ben Cuning - willcun1@umbc.edu: Function will
```

```
% calculate the average current value for a matrix (currentMatrix) passed to it.
```

```
% Function assumes runs are column seperated (i.e. all rows in a column
```

```
% are from the same run). Returns will be a vector of row averages (avgCurrent)
```

```
% and row standard deviations (currentError) for the current matrix passed.
```

```
avgCurrent = mean(currentMatrix,2);
```

```
currentError = std(currentMatrix,0,2);
```

```
end
```

Function: cyclicVoltammetryAverager.m

Purpose: Generate the average cyclic voltammogram data for a set of CV data.

Code:

```
function [ ] = cyclicVoltammetryAverager( numSets )
%CYCLICVOLTAMMETRYAVERAGER calculates average Voltammogram for
CV data
%-----
% numSets = the number of replicate CV measurements to average
%-----Written 8/6/2014 by Ben Cuning - willcun1@umbc.edu -----

%pre-allocate input array
dataCell = cell(numSets,2);
%Get Raw data SETS
%first argument in uigetdir can be changed to the default data directory for the user
inPath = uigetdir('C:\My Documents\Data',...
    'Where to load?');
%import data
for inLoop = 1:numSets
    dataCell{inLoop,2} = pineDataImport(inPath);
    setLabel = num2str(inLoop);
    dataCell{inLoop,1} = inputdlg(strcat('Set ',setLabel,' Name?'));
end
%pre-allocate output array
averagedData = cell(numSets,1);
%Get average of currents for voltammograms
for avgLoop = 1:numSets
    dataToAvg = dataCell{avgLoop,2};
    %strip x values for output later
    xData = dataToAvg(:,1);
    yData = dataToAvg(:,2:end);
    avgData = mean(yData.');"
    %reassemble x values with average y values
    averagedData{avgLoop,1} = [xData,avgData];
end
```

Function: pineCalibrationDataProcessing.m

Purpose: Function was written to assist in generating calibration plots for direct BPED analysis of samples. Output is a data structure – which is different than a

cell array. Readers are referred to MATLAB documentation on structures for more information.

Code:

```
function [ OutputStructure ] =...
    pineCalibrationDataProcessing( numSigs,numBlanks,sigsPerSet )
%PINECALIBRATIONDATAPROCESSING calculates calibration data
% Written 2/25/2015 by Ben Cuning - willcun1@umbc.edu: Function will
% assist in construction of calibration plots for amperometric sensor
% data. Function will import raw data, parse for blank and signal data,
% using the numBlanks (number of replicate blank measurements) and
% the number of concentrations (numSigs) with sigsPerSet replicates.
% Function then averages currents and calculates std deviations,
% corrects for background and propagates error, calculates areas by row
% and further propagates error, finally outputting a structure (Output Structure)
% of calculated and raw data.

%call function to import data files
importedData = pineDataImport();

%parse imported data to isolate currents and time
time = importedData(:,1);
[numRows,~] = size(time);
blankCurrents = importedData(:,2:numBlanks+1);
sigCurrents = importedData(:,numBlanks+1:end);
sigIAverage(numRows,numSigs) = 0;
sigIError(numRows,numSigs) = 0;

%time to get averages
[blankIAvg,blankIStdDevs] = CurrentAverager(blankCurrents);
for average = 1:numSigs
    [sigIAverage(:,average),sigIError(:,average)] = CurrentAverager(...
        sigCurrents(:,(1+(sigsPerSet*(average-1)))):...
        (sigsPerSet+(sigsPerSet*(average-1))));
end

%Correct for background, propagate errors
[blankCorrectedCurrents,blankCorrectedCurrentStdDevs] = ...
    blankICxn(blankIAvg,blankIStdDevs,sigIAverage,sigIError);

%Convert Current data to areas row-by-row
correctedAreas = current2areaConverter(blankCorrectedCurrents);
blankAreas = current2areaConverter(blankIAvg);
```



```
%propagate errors for area conversion
correctedAreaErrors = areaErrorProp(blankCorrectedCurrentStdDevs);
blankAreaErrors = areaErrorProp(blankIStdDevs);
```

```
%build output structure
OutputStructure.time = time;
OutputStructure.sigAvgI = sigIAverage;
OutputStructure.sigIErr = sigIError;
OutputStructure.blankIAvg = blankIAvg;
OutputStructure.blankIStdDevs = blankIStdDevs;
OutputStructure.correctedSignals = blankCorrectedCurrents;
OutputStructure.correctedErrors = blankCorrectedCurrentStdDevs;
OutputStructure.blankAreas = blankAreas;
OutputStructure.blankAreaErrors = blankAreaErrors;
OutputStructure.correctedAreas = correctedAreas;
OutputStructure.sigAreaErrors = correctedAreaErrors;
```

```
End
```

Function: pineStdAdd.m

Purpose: Similar to the previous function, this function was written to assist in generation of standard additions plots by analyzing raw data.

Code:

```
function [ ] = pineStdAdd( numSets, numCurvesPerSet,delTimeIn,intTimeIn )
%PINESTDADD Assists in generation of standard addition curves for BPED data
%-----
% This function will import a multi-set, multi-curve group of data files,
% zip them together and then processes them. The function will assume
% the 1st three files are blanks, and that the rest are data in sequence
% - this is subject to edit in later versions. The data will be parsed
% by user input as numSets = number of data sets total (including blank)
% and numCurvesPerSet = number of replicates in the data set, with delay
% time of delTimeIn and integration time of intTimeIn.
%-----Written 8/28/2014 by Ben Cuning - willcun1@umbc.edu-----
```

```
%Let's zip files!
dataIn = pineTextZipper;
```

```
%Okay, we have zipped data - let's strip the x and y out for processing
xData = dataIn(:,1);
%convert delTime from ms to sec same with intTime
delTime = delTimeIn/1000;
intTime = (intTimeIn/1000)+delTime;
```

```

%find indices for delTime and intTime
delRow = find(dataIn(:,1)==delTime);
intRow = find(dataIn(:,1)==intTime);
%Get y Data
yData = dataIn(:,2:end);
%set data to actual integration time set
xDat(delRow:intRow,:) = [];
yData(delRow:intRow,:) = [];
%get num rows in x data for posterity
[numRows,~] = size(xData);

%Assuming the first numCurvesPerSet are blank runs - let's deal with those.
%Calculate the mean
blankAvgCurrent = mean(yData(:,1:numCurvesPerSet),2);
%std of raw current - why not?
blankStdCurrent = std(yData(:,1:numCurvesPerSet),1,2);
%Let's get the blank average area and the blank area standard deviation as
%well, 1st - set blank data
blankDat = [xDat,yData(:,1:numCurvesPerSet)];
%get areas
blankAreas = pineMultiAreaCalc(blankDat,0)';
%avg and std
blankAvgArea = mean(blankAreas(1,:));
blankStdArea = std(blankAreas(1,:));

%lets get the copper runs seperated from the blanks - just to make life
%easier
copperData = yData(:,numCurvesPerSet+1:numSets*numCurvesPerSet);

%calculate non-current corrected areas
nonCorAreas = zeros(numSets-1,3);
for nonCorLp = 1:(numSets-1)
    %re-combine with x data
    calcDat = [xDat,copperData(:,...
        (1+((nonCorLp-1)*numCurvesPerSet)...
        :numCurvesPerSet+((nonCorLp-1)*numCurvesPerSet))]);
    %calc areas
    nonCorAreas(nonCorLp,1:3) = pineMultiAreaCalc(calcDat,0)';
end
%clear loop counter and temporary matrices
clear('nonCorLp','calcDat');

%now correct copper data for the blank current -> subtraction!

```

```

%pre-allocate storage variable
blankCorrected = zeros(numRows,((numSets-1)*numCurvesPerSet));

for blankSubLoop = 1:(numSets-1)*numCurvesPerSet
    blankCorrected(:,blankSubLoop) = copperData(:,blankSubLoop)...
        -blankAvgCurrent;
end
%clear loop counter
clear blankSubLoop;
%get average copper currents
%pre-allocate loop variable
copperAvgData = zeros(numRows,(numSets-1));
for copperCurrentAvgLoop = 1:numSets-1
    copperAvgData(:,copperCurrentAvgLoop) = ...
        mean(...
            blankCorrected(:,(1+((copperCurrentAvgLoop-1)*numCurvesPerSet)...
                :numCurvesPerSet+((copperCurrentAvgLoop-1)*numCurvesPerSet))),2);
end
%clear loop index
clear copperCurrentAvgLoop;

%so data has been blank corrected, now need to get average areas for
%copper data
%have to do it in a for loop
%preallocate loop variables
blankCorareas = zeros(numSets-1,3);
for iLoop = 1:(numSets-1)
    %re-combine with x data
    calcDat = [xData,blankCorrected(:,...
        (1+((iLoop-1)*numCurvesPerSet)...
            :numCurvesPerSet+((iLoop-1)*numCurvesPerSet))];
    %calc areas
    blankCorareas(iLoop,1:3) = pineMultiAreaCalc(calcDat,0)';
end
%clear loop counter and temporary matrices
clear('iLoop','calcDat');

%assemble output matrices
areasOut = [blankAvgArea,blankStdArea];
blankCurrentOut = [xData,blankAvgCurrent,blankStdCurrent];
copCurrentOut = [xData,copperAvgData];
%export data!
%get outpath

```

```

outPath = uigetdir;
%set filenames
blankAreaFileName = strcat(outPath,'\Blank_Area_Output.csv');
blankCurrentFileName = strcat(outPath,'\Blank_Current_Output.csv');
areaFileName = strcat(outPath,'\BlankCor_Copper_Area_Out.csv');
nonCorAreaFN = strcat(outPath,'\nonCor_Cop_Area_Out.csv');
copCurrentFileName = strcat(outPath,'\Copper_Current_Output.csv');
%perform export
dlmwrite(blankAreaFileName,areasOut);
dlmwrite(nonCorAreaFN,nonCorAreas);
dlmwrite(areaFileName,blankCorareas);
dlmwrite(blankCurrentFileName,blankCurrentOut);
dlmwrite(copCurrentFileName,copCurrentOut);

end

```

Function: tOxOptimization.m

Purpose: Function was written to assist in determining optimal values for oxidation time components (t_{int} , t_{del} , t_{clean}) based on a series of I vs. t curves.

Code:

```

function [optdel,optclean] = tOxOptimization(time,currents)
%TOXOPTIMIZATION Calculates optimal values for tint, tdel, and tclean
% Written 2/25/2015 by Ben Cuning - willcun1@umbc.edu: Function will
% determine the optimal values for integration time, delay time, and
% cleaning time based on areas using
% current (currents), and time (time) data put into the function.

[~,iCol] = size(currents);
%figure out cleaning time - time at which I is less than 95% of max (most
%stripping complete)
cleanInd(1,iCol) = 0;
for tClLp = 1:iCol
    maxI = max(currents(:,tClLp));
    cleanI = maxI*0.05;
    cleanInd(1,tClLp) = find(currents(:,tClLp)<cleanI,1,'first');
end
cleanTimeInd = max(cleanInd);
optclean = time(cleanTimeInd,1);

%delay time is best at whatever time the blank current is < 95% initial
%value (non-faradaic current has settled)
blankI = currents(:,1);
peakI = max(blankI);

```

```

delI = 0.05*peakI;
delRow = find(blankI<delI,1,'first');
optdel = time(delRow,1);

%now that we have our limits of delay time and cleaning time, we can
%calculate S/N values for integration time - treating blank signal as
%noise.
intTimes = time(delRow:cleanTimeInd,1);
intAreas = areas(delRow:cleanTimeInd,:);
[intRows,intCols] = size(intAreas(:,2:end));
blankInt = intAreas(:,1);
sigInt = intAreas(:,2:end);
S2N(intRows,intCols) = 0;
intTime(intRows) = 0;
for intTimeCalc = 1:intRows
    noise = sum(blankInt(1:(1+(intRows-1)),1));
    signals = sum(sigInt(1:(1+(intRows-1)),:),1);
    S2N(intTimeCalc,:) = signals/noise;
    intTime(intTimeCalc) = intTimes(intTimeCalc) - intTimes(1,1);
    if intTimeCalc == 1
        intTime = 0.001;
    end
end
plot(intTime,S2N);
inputTime = ...
    inputdlg('Please select Optimum integration time length','Int. Time.');
```

end

Function: wmc0096DataAnalysis.m

Purpose: This function was written to determine what effects various BPED waveform parameters had on the linear least-squares correlation coefficient (R^2).

Code:

```

function [ ] = wmc0096DataAnalysis( numCon,conVec )
%WMC0096DATAANALYSIS calculates the areas and errors of data and fits R^2's
%-----
% numCon = number of Concentrations; conVec = column vector of [Cu]'s
%-----Written 1/14/2015 by Ben Cuning - willcun1@umbc.edu -----

if nargin < 2
    %create concentration vector
    conVec = zeros(numCon,1);
    for cnLp = 1:numCon
```

```

    prompt = strcat...
        ('Please enter the ',int2str(cnLp),'th concentration. ');
    conVec(cnLp,1) = cell2mat(inputdlg(prompt,'Concentrations'));
end
clear cnLp
end

%import data
inDat = pineMultiDataImport(numCon);
%get number of rows in cell portion {#} using column 1 -> (:,1).
sampDat = inDat{1};
[numRows,numCols] = size(sampDat);
clear sampDat

%preallocate current matrices and R-squared matrix
avgCur = zeros(numRows,numCon);
stdCur = zeros(numRows,numCon);
rSqs = zeros(numRows,1);

%get average current and std dev current
for IcalcLp = 1:numCon
    calcDat = inDat{IcalcLp};
    yDat = calcDat(:,2:end);
    avgCur(:,IcalcLp) = mean(yDat,2);
    stdCur(:,IcalcLp) = std(yDat,0,2);
    if IcalcLp == 1
        blkCur = avgCur(:,1);
        blkStdCur = stdCur(:,1);
    end
end
avgCorCur = zeros(numRows,numCon);
stdCorCur = zeros(numRows,numCon);
arblkcorAr = zeros(numCon,1);
arblkcorStd = zeros(numCon,1);
%correct for background, propagate errors
for IcorLp = 1:numCon
    avgCorCur(:,IcorLp) = avgCur(:,IcorLp) - blkCur;
    stdCorCur(:,IcorLp) = ((stdCur(:,IcorLp).^2)+(blkStdCur.^2)).^(1/2);
end
clear IcalcLp; clear IcorLp

%using each row, calculate the R-squared value with concentrations for I
for IlinLp = 1:numRows

```

```

    curDat = avgCorCur(IlinLp,:);
    linMod = fitlm(conVec,curDat');
    rSqs(IlinLp,1) = linMod.Rsquared.Ordinary;
end
clear IlinLp;
disp('Currents finished.')
tic;
%need to calculate areas 1 trapezoid at a time (so like trapz but no sum)
%needs to be done y by y, x's constant. Since data is [x|y1|y2|y3] must
%use nested loops. Only working on 1st 1000 rows since majority of signal
%is there.
%pre-allocate output matrix of trapezoid areas that is (numRows-1,numCon*3)
trapezoids = zeros(999,3*numCon);
%first call on proper matrix and isolate x and y data
for datLp = 1:numCon
    time = inDat{datLp}(1:1000,1);
    current = inDat{datLp}(1:1000,2:end);
    [numTimes,numYs] = size(current);
    strips = zeros(numTimes-1,numYs);
    %Now run calculation for as many y columns exist
    for trapCalLp = 1:numYs
        %To get individual trapezoids must do another loop for rows - yay.
        for trapRows = 1:numTimes-1
            deltaTime = time(trapRows+1)-time(trapRows);
            strips(trapRows,trapCalLp) = ...
                ((current(trapRows+1,trapCalLp)+...
                    current(trapRows,trapCalLp))/2)*deltaTime;
        end
    end
    trapezoids(1:numTimes,(1+(numYs*(datLp-1)):numYs+(numYs*(datLp-1))) =
strips;
end
clear datLp trapCalLp trapRows strips

%Now that areas are individual strips, the integration time is given by
%summing the rows for that length of time and delay time is performed by
%ignoring a certain number of rows. Since deltaTime is always the same,
%will just use that variable here to calculate time values for delay and
%integration times, but loops will be based on the number of rows. Output 3
%column matrix is going to be MASSIVE unless only the first 1000 rows are
%used as delays, after that a significant portion of signal is gone. This
%will give almost 500k calculations, and will probably take a week.
areaRsquares = zeros(498501,3);

```

```

%initialize variable to count how many calculations have been performed
calcCount = 0;
%ca-chunk ca-chunk
for delayLp = 0:998
    for integrateLp = 1:999-delayLp
        delayTime = delayLp*deltaTime;
        integrationTime = deltaTime*integrateLp;
        integratedSums = sum(trapezoids(1+delayLp:integrateLp,:),1);
        %now have integrated sums, need to get averages for individual
        %concentrations so that I can build the data to get R^2's.
        averageSums = zeros(1,numCon);
        blkCorSums = averageSums;
        for avgArLp = 1:numCon
            averageSums(1,avgArLp) = mean(integratedSums(1,...
                1+(3*(avgArLp-1)):3+(3*(avgArLp-1))));
            if avgArLp > 1
                %do blank correction and stuff
                blkCorSums(1,avgArLp) =...
                    averageSums(1,avgArLp) - averageSums(1,1);
            else
                blkCorSums(1,avgArLp) =...
                    averageSums(1,1);
            end
        end
        linFit = fitlm(conVec,blkCorSums');
        Rsq = linFit.Rsquared.Ordinary;
        %increment calculation counter variable
        calcCount = calcCount + 1;
        areaRsquares(calcCount,:) = [delayTime, integrationTime,Rsq];
    end
end
toc;
%output the following: {time|avgCorCur|stdCorCur|rSqs} and
%{concMat|avgAreas|stdAreas}

time = sampDat(:,1);
cursOut = [time,avgCorCur,stdCorCur,rSqs];
arsOut = [conVec,arblkcorAr,arblkcorStd];
[nameOut,pathOut] = uiputfile('*.csv','Save Current Data...');
dlmwrite(strcat(pathOut,nameOut),cursOut,',');
[nameOut,pathOut] = uiputfile('*.csv','Save Area Data...');
dlmwrite(strcat(pathOut,nameOut),arsOut,',');
[nameOut,pathOut] = uiputfile('*.csv','Save Area Rsquares...');

```



```
dlmwrite(strcat(pathOut,nameOut),areaRsquares,',' );
end
```

Scripts

Script: SignaltoNoiseCalcwmc0095.m

Purpose: Quickly analyze data from the experiment listed in the script name and get average and standard deviation of areas.

Code:

```
%preallocate area vector
calcAreas = zeros(66,1);
%import data parse into blanks (blkData) and signals (sigData), and then background
%correct (subData).
dataIn = pineTextZipper();
blkData = dataIn(:,2:34);
sigData = dataIn(:,35:end);
subData = sigData-blkData;
data = [dataIn(:,1),blkData,subData];
for areaCalc = 1:66
    calcAreas(areaCalc,1) = trapz(data(:,1),data(:,2+(areaCalc-1)));
end
clear areaCalc
avgArea = zeros(22,1);
stdArea = zeros(22,1);
for avgLp = 1:22
    mat = [calcAreas(1+(3*(avgLp-1)),1),calcAreas(2+(3*(avgLp-1)),1),...
        calcAreas(3+(3*(avgLp-1)),1)];
    avgArea(avgLp,1) = mean(mat);
    stdArea(avgLp,1) = std(mat);
end
clear avgLp
%pull appropriate averages and noise to calculate S/N
signal = avgArea(12:22,1);
noise = stdArea(1:11,1);
sigNoi = signal./noise;
```

Script: AnalysisScriptGCSensorDelTimeCal.m

Purpose: Script quickly analyzed a series of delay and integration times to assist in optimization attempts.

Code:

```
%create tint vector
tint = [10,15,20,25,30];
%Get data
```

```

in = pineDataImport(pwd);
%Make vector of time data
tvec = in(:,1);
%remove time data from input
in(:,1) = [];
%get number of data sets
[~,curves] = size(in);
numSets = int32(curves/3);
%preallocate loop variables
mnAr = zeros(numSets,6);
stAr = zeros(numSets,6);
areas = zeros(numSets,3);
rawAr = cell(6,1);
%compute areas, averages, standard deviations, and output
for delLp = 1:6
    delTim = tint(1,delLp);
    for x = 1:numSets
        areas(x,:) = pineMultiAreaCalc([tvec(delTim:delTim+10,1),...
            in(delTim:delTim+10,(1+((x-1)*3)):(3+((x-1)*3)))],0);
        mnAr(x,delLp) = mean(areas(x,:));
        stAr(x,delLp) = std(areas(x,:));
    end
    rawAr{delLp,1} = areas;
    clear areas;
end
out = [mnAr(:,1),stAr(:,1),mnAr(:,2),stAr(:,2),mnAr(:,3),stAr(:,3),...
    mnAr(:,4),stAr(:,4),mnAr(:,5),stAr(:,5),mnAr(:,6),stAr(:,6)]];

```

Script: AnalysisScriptGCSensorIntTimeCal.m

Purpose: Similar to above function, the purpose of this function was to assist in waveform optimization.

Code:

```

%create tint vector
tint = [10,15,20,25,30,50];
%Get data
in = pineDataImport(pwd);
%Make vector of time data
tvec = in(:,1);
%remove time data from input
in(:,1) = [];
%get number of data sets
[~,curves] = size(in);
numSets = int32(curves/3);

```

```


```
%preallocate loop variables
mnAr = zeros(numSets,6);
stAr = zeros(numSets,6);
areas = zeros(numSets,3);
rawAr = cell(6,1);
%compute areas, averages, standard deviations, and output
for intLp = 1:6
 intTim = tint(1,intLp);
 for x = 1:numSets
 areas(x,:) = pineMultiAreaCalc([tvec(5:5+intTim,1),...
 in(5:5+intTim,(1+((x-1)*3)):(3+((x-1)*3)))],0);
 mnAr(x,intLp) = mean(areas(x,:));
 stAr(x,intLp) = std(areas(x,:));
 end
 rawAr{intLp,1} = areas;
 clear areas;
end
out = [mnAr(:,1),stAr(:,1),mnAr(:,2),stAr(:,2),mnAr(:,3),stAr(:,3),...
 mnAr(:,4),stAr(:,4),mnAr(:,5),stAr(:,5),mnAr(:,6),stAr(:,6)]];

```


```

Script: AnalysisScriptAuBatch.m

Purpose: Rapid analysis of collected data.

Code:

```


```
%Get data
in = pineDataImport(pwd);
%Make vector of time data
tvec = in(:,1);
%remove time data from input
in(:,1) = [];
%get number of data sets
[~,curves] = size(in);
numSets = int32(curves/3);
%preallocate loop variables
areas = zeros(numSets,3);
mnAr = zeros(numSets,1);
stAr = zeros(numSets,1);
out = zeros(numSets,5);
%compute areas, averages, standard deviations, and output
for x = 1:numSets
 areas(x,:) = pineMultiAreaCalc([tvec(21:41,1),...
 in(21:41,(1+((x-1)*3)):(3+((x-1)*3)))],0);
 mnAr(x,:) = mean(areas(x,:));
 stAr(x,:) = std(areas(x,:));
end

```


```

```
    out(x,:) = [areas(x,:),mnAr(x,:),stAr(x,:)];  
end
```

Bibliography

- (1) A Metals Primer - Dartmouth Toxic Metals Superfund Research Program <http://www.dartmouth.edu/~toxmetal/toxic-metals/metals-primer.html> (accessed Apr 10, 2017).
- (2) Ancient History Lesson: Copper Discovery and Early Uses <https://www.thebalance.com/copper-history-pt-i-2340112> (accessed Apr 10, 2017).
- (3) History of Metals Timeline Infographic <http://www.makin-metals.com/about/history-of-metals-infographic/> (accessed Apr 10, 2017).
- (4) In *Metallurgy for the Non-Metallurgist*; Reardon, A. C., Ed.; ASM International: Materials Park, OH, 2011; pp 73–84.
- (5) Thurman, R. B.; Gerba, C. P. *Crit. Rev. Environ. Control* **1989**, 18 (4), 295–315.
- (6) Grass, G.; Rensing, C.; Solioz, M. *Appl. Environ. Microbiol.* **2011**, 77 (5), 1541–1547.
- (7) Nguyen, T. T. M.; Park, H.-J.; Kim, J. Y.; Kim, H.-E.; Lee, H.; Yoon, J.; Lee, C. *Environ. Sci. Technol.* **2013**, 47 (23), 13661–13667.
- (8) Brunel, F.; El Gueddari, N. E.; Moerschbacher, B. M. *Carbohydr. Polym.* **2013**, 92 (2), 1348–1356.
- (9) CDA Press Releases: March 25, 2008, U.S. EPA Approves Registration of Antimicrobial Copper Alloys http://www.copper.org/about/pressreleases/2008/pr2008_Mar_25.html (accessed Feb 20, 2016).
- (10) Spotte, S. In *Captive Seawater Fishes: Science and Technology*; John Wiley & Sons: New York, 1992; pp 469–474.
- (11) Yanong, R. P. E. *Uses of Copper in Marine Systems*; Fisheries and Aquatic Sciences; FA165; University of Florida IFAS Extension, 2009.
- (12) Cardeilhac, P. T.; Whitaker, B. R. *Vet. Clin. North Am. Small Anim. Pract.* **1988**, 18 (2), 435–448.
- (13) U.S. National Aquaculture Association. Drugs Used in the US Aquaculture Industry.
- (14) Southern Agricultural Insecticides, Inc. .
- (15) Table of Regulated Drinking Water Contaminants | Your Drinking Water | US EPA <http://www.epa.gov/your-drinking-water/table-regulated-drinking-water-contaminants#Inorganic> (accessed Feb 20, 2016).
- (16) Blossom, N. *Am Chemet Corp* **2007**, 1–8.
- (17) Kidwell, D. A. *Measuring Copper in Seawater-An Automated Detection of Copper Binding Capacity*; DTIC Document, 2003.
- (18) Watson, C.; Yanong, R. P. *Fact Sheet FA-13 Ser. Dep. Fish. Aquat. Sci. Fla. Coop. Ext. Serv. Inst. Food Agric. Sci. Univ. Fla.* **2006**.
- (19) Boyd, C. *Glob. Aquac. Advocate* **2015**, No. Jan/Feb, 38–41.
- (20) Bruland, K. W.; Coale, K. H.; Mart, L. *Mar. Chem.* **1985**, 17 (4), 285–300.

- (21) Kodama, H.; Fujisawa, C. *Metallomics* **2009**, *1* (1), 42–52.
- (22) Jaiser, S. R.; Winston, G. P. *J. Neurol.* **2010**, *257* (6), 869–881.
- (23) *Guidelines for drinking-water quality*, 4th ed.; World Health Organization, Ed.; World Health Organization: Geneva, 2011.
- (24) Dorsey, A.; Ingerman, L.; Swarts, S. *Toxicological Profile for Copper*; U.S. Department of Health and Human Services: Agency for Toxic Substances and Disease Registry, 2004.
- (25) Wilson Disease | National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) <http://www.niddk.nih.gov/health-information/health-topics/digestive-diseases/wilson-disease/Pages/facts.aspx> (accessed Feb 20, 2016).
- (26) Wilson Disease - National Library of Medicine - PubMed Health <http://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0024926/> (accessed Feb 20, 2016).
- (27) Huheey, J. E.; Keiter, E. A.; Keiter, Richard L. In *Inorganic Chemistry: Principles of Structure and Reactivity*; HarperCollins College Publishers, 1993; pp 588–599.
- (28) Moffett, J. W.; Zika, R. G. *Mar. Chem.* **1983**, *13*, 239–251.
- (29) Sadiq, M. In *Toxic Metal Chemistry in Marine Environments*; Marcel Dekker, Inc.: New York, 1992; pp 198–249.
- (30) Gibbon-Walsh, K.; Salaün, P.; van den Berg, C. M. G. *J. Phys. Chem. A* **2012**, *116* (25), 6609–6620.
- (31) Richardson, S. D. *Anal. Chem.* **1999**, *71* (12), 181–216.
- (32) Richardson, S. D. *Anal. Chem.* **2001**, *73* (12), 2719–2734.
- (33) Richardson, S. D. *Anal. Chem.* **2003**, *75* (12), 2831–2857.
- (34) Richardson, S. D.; Ternes, T. A. *Anal. Chem.* **2005**, *77* (12), 3807–3838.
- (35) Richardson, S. D. *Anal. Chem.* **2007**, *79* (12), 4295–4324.
- (36) Richardson, S. D. *Anal. Chem.* **2009**, *81* (12), 4645–4677.
- (37) Richardson, S. D.; Ternes, T. A. *Anal. Chem.* **2011**, *83* (12), 4614–4648.
- (38) Richardson, S. D.; Ternes, T. A. *Anal. Chem.* **2014**, *86* (6), 2813–2848.
- (39) Richardson, S. D.; Kimura, S. Y. *Anal. Chem.* **2016**, *88* (1), 546–582.
- (40) Koester, C. J.; Simonich, S. L.; Esser, B. K. *Anal. Chem.* **2003**, *75* (12), 2813–2829.
- (41) Wang, J.; Tian, B.; Wang, J.; Lu, J.; Olsen, C.; Yarnitzky, C.; Olsen, K.; Hammerstrom, D.; Bennett, W. *Anal. Chim. Acta* **1999**, *385* (1), 429–435.
- (42) Martin, T. D. *US EPA* **2003**.
- (43) Martin, T. D.; Brockhoff, C. A.; Creed, J. T.; EMMC Methods Work Group. U.S. Environmental Protection Agency 1994.
- (44) Creed, J. T.; Brockhoff, C. A.; Martin, T. D. U.S. Environmental Protection Agency 1994.
- (45) Creed, J. T.; Martin, T. D.; O'Dell, J. W. U.S. Environmental Protection Agency 1994.
- (46) Søndergaard, J.; Asmund, G.; Larsen, M. M. *MethodsX* **2015**, *2*, 323–330.

- (47) Minami, T.; Konagaya, W.; Zheng, L.; Takano, S.; Sasaki, M.; Murata, R.; Nakaguchi, Y.; Sohrin, Y. *Anal. Chim. Acta* **2015**, 854, 183–190.
- (48) Nicolai, M.; Rosin, C.; Tousset, N.; Nicolai, Y. *Talanta* **1999**, 50 (2), 433–444.
- (49) Bond, A. M.; Wallace, G. G. *Anal. Chem.* **1981**, 53 (8), 1209–1213.
- (50) Bond, A. M.; Wallace, G. G. *Anal. Chem.* **1982**, 54 (11), 1706–1712.
- (51) Hojabri, H.; Lavin, A. G.; Wallace, G. G.; Riviello, J. M. *Anal. Chem.* **1987**, 59 (1), 54–57.
- (52) Bauer, H.; Ottenlinger, D.; Yan, D. *Chromatographia* **1989**, 28 (5/8), 315–317.
- (53) Ohta, K.; Tanaka, K.; Paull, B.; Haddad, P. R. *J. Chromatogr. A* **1997**, 770, 219–227.
- (54) Lopez-Ruiz, B. *J. Chromatogr. A* **2000**, 881, 607–627.
- (55) LaCourse, W. R. *Pulsed Electrochemical Detection in High-Performance Liquid Chromatography*; Techniques in Analytical Chemistry; John Wiley & Sons: New York, 1997.
- (56) Lu, H.; Mou, S.; Yan, Y.; Tong, S.; Rivello, J. M. *J. Chromatogr. A* **1998**, 800, 247–255.
- (57) Osipova, E. A.; Shapovalova, E. N.; Ofitserova, M. N.; Podlesnykh, S. V. *J. Anal. Chem.* **2000**, 55 (1), 52–57.
- (58) Santoyo, E.; Santoyo-Gutierrez, S.; Verma, S. P. *J. Chromatogr. A* **2000**, 884, 229–241.
- (59) *Dionex Technical Note 25: Determination of Transition Metals in Complex Matrices by Chelation Ion Chromatography.*
- (60) *Dionex Application Update 168: Determination of Transition Metals in Complex Matrices Using Chelation Ion Chromatography.*
- (61) *Dionex Technical Note 10: Chromatography of Metal Ions.*
- (62) McGillicuddy, N.; Nesterenko, E. P.; Nesterenko, P. N.; Jones, P.; Paull, B. *J. Chromatogr. A* **2013**, 1276, 102–111.
- (63) Chen, G. J.; Lee, N. M.; Hu, C. C.; Liu, C. Y. *J. Chromatogr. A* **1995**, 699, 343–351.
- (64) Francois, C.; Morin, P.; Dreux, M. *J. Chromatogr. A* **1995**, 717, 393–408.
- (65) Wen, J.; Cassidy, R. M. *Anal. Chem.* **1996**, 68 (6), 1047–1053.
- (66) Bereza-Malcolm, L. T.; Mann, G.; Franks, A. E. *ACS Synth. Biol.* **2015**, 4 (5), 535–546.
- (67) Pal, S.; Chatterjee, N.; Bharadwaj, P. K. *RSC Adv.* **2014**, 4 (51), 26585.
- (68) Montes-Bayón, M.; DeNicola, K.; Caruso, J. A. *J. Chromatogr. A* **2003**, 1000 (1), 457–476.
- (69) Tsednee, M.; Huang, Y.-C.; Chen, Y.-R.; Yeh, K.-C. *Sci. Rep.* **2016**, 6 (1).
- (70) Gehrig, P. M.; You, C.; Dallinger, R.; Gruber, C.; Brouwer, M.; Kägi, J. H.; Hunziker, P. E. *Protein Sci. Publ. Protein Soc.* **2000**, 9 (2), 395.
- (71) Mota, A. M.; Pinheiro, J. P.; Simões Gonçalves, M. L. *J. Phys. Chem. A* **2012**, 116 (25), 6433–6442.
- (72) Suginta, W.; Khunkaewla, P.; Schulte, A. *Chem. Rev.* **2013**, 113 (7), 5458–5479.

- (73) Gumpu, M. B.; Sethuraman, S.; Krishnan, U. M.; Rayappan, J. B. B. *Sens. Actuators B Chem.* **2015**, *213*, 515–533.
- (74) Fedorowski, J.; LaCourse, W. R. *Anal. Chim. Acta* **2015**, *861*, 1–11.
- (75) Skoog, D. A.; Holler, F. J.; Crouch, S. R. *Principles of Instrumental Analysis*, 6th ed.; Brooks/Cole: New York.
- (76) Marple, R. L.; LaCourse, W. R. In *Ewing's Analytical Instrumentation Handbook*; Cazes, J., Ed.; Marcel Dekker, Inc.: New York, 2005; pp 509–527.
- (77) Westall, J. C.; Morel, F. M. M.; Hume, D. N. *Anal. Chem.* **1979**, *51* (11), 1792–1798.
- (78) Qin, W.; Liang, R.; Fu, X.; Wang, Q.; Yin, T.; Song, W. *Anal. Chem.* **2012**, *84* (24), 10509–10513.
- (79) Gelhaus, S. L.; LaCourse, W. R. In *Ewing's Analytical Instrumentation Handbook*; Cazes, J., Ed.; Marcel Dekker, Inc.: New York, 2005; pp 561–580.
- (80) Bajwa, S. Z.; Dumler, R.; Lieberzeit, P. A. *Sens. Actuators B Chem.* **2014**, *192*, 522–528.
- (81) Bard, A. J.; Faulkner, L. R. *Electrochemical Methods: Fundamentals and Applications*, 2nd ed.; John Wiley & Sons: New York, 2001.
- (82) *Laboratory Techniques in Electroanalytical Chemistry*, 2nd ed.; Kissinger, P. T., Heineman, W. R., Eds.; Marcel Dekker, Inc.: New York, 1996.
- (83) Olson, M. P.; LaCourse, W. R. In *Ewing's Analytical Instrumentation Handbook*; Cazes, J., Ed.; Marcel Dekker, Inc.: New York, 2005; pp 529–543.
- (84) LaCourse, W. R. In *Ewing's Analytical Instrumentation Handbook*; Cazes, J., Ed.; Marcel Dekker, Inc.: New York, 2005; pp 545–559.
- (85) Ramaley, L.; Krause Jr, M. S. *Anal. Chem.* **1969**, *41* (11), 1362–1365.
- (86) O'Dea, J. J.; Osteryoung, J.; Osteryoung, R. A. *Anal. Chem.* **1981**, *53* (4), 695–701.
- (87) Osteryoung, J. G.; Osteryoung, R. A. *Anal. Chem.* **1985**, *57* (1), 101A – 110A.
- (88) Liu, A.-C.; Chen, D.; Lin, C.-C.; Chou, H.-H.; Chen, C. *Anal. Chem.* **1999**, *71* (8), 1549–1552.
- (89) Zhihua, W.; Xiaole, L.; Jianming, Y.; Yaxin, Q.; Xiaoquan, L. *Electrochimica Acta* **2011**, *58*, 750–756.
- (90) Florence, T. M. *J. Electroanal. Chem. Interfacial Electrochem.* **1972**, *35*, 237–245.
- (91) Wang, J.; Lu, J.; Hocevar, S. B.; Farias, P. A. M.; Ogorevc, B. *Anal. Chem.* **2000**, *72* (14), 3218–3222.
- (92) Prado, C.; Wilkins, S. J.; Marken, F.; Compton, R. G. *Electroanalysis* **2002**, *14* (4), 262–272.
- (93) Dai, X.; Qiu, F.; Zhou, X.; Long, Y.; Li, W.; Tu, Y. *Anal. Chim. Acta* **2014**, *848*, 25–31.
- (94) Bonfil, Y.; Brand, M.; Kirowa-Eisner, E. *Anal. Chim. Acta* **1999**, *387*, 85–95.
- (95) Alves, G. M. S.; Magalhães, J. M. C. S.; Salaün, P.; van den Berg, C. M. G.; Soares, H. M. V. M. *Anal. Chim. Acta* **2011**, *703* (1), 1–7.

- (96) Herzog, G.; Moujahid, W.; Twomey, K.; Lyons, C.; Ogurtsov, V. I. *Talanta* **2013**, *116*, 26–32.
- (97) Mohadesi, A.; Taher, M. *Talanta* **2007**, *72* (1), 95–100.
- (98) Zhang, P.; Dong, S.; Gu, G.; Huang, T. *Bull. Korean Chem. Soc.* **2010**, *31* (10), 2949–2954.
- (99) Zhao, H.; Jiang, Y.; Ma, Y.; Wu, Z.; Cao, Q.; He, Y.; Li, X.; Yuan, Z. *Electrochimica Acta* **2010**, *55* (7), 2518–2521.
- (100) Fu, X.-C.; Wu, J.; Li, J.; Xie, C.-G.; Liu, Y.-S.; Zhong, Y.; Liu, J.-H. *Sens. Actuators B Chem.* **2013**, *182*, 382–389.
- (101) Cantalapiedra, A.; Gismera, M. J.; Procopio, J. R.; Sevilla, M. T. *Talanta* **2015**, *139*, 111–116.
- (102) Nezhadali, A.; Sadeghzadeh, S. *Sens. Actuators B Chem.* **2016**, *224*, 134–142.
- (103) Xing, H.; Xu, J.; Zhu, X.; Duan, X.; Lu, L.; Wang, W.; Zhang, Y.; Yang, T. *J. Electroanal. Chem.* **2016**, *760*, 52–58.
- (104) Etienne, M.; Bessiere, J.; Walcarius, A. *Sens. Actuators B* **2001**, *76*, 531–538.
- (105) Ashrafi, A. M.; Vytrās, K. *Electrochimica Acta* **2012**, *73*, 112–117.
- (106) Wei, Y.; Gao, C.; Meng, F.-L.; Li, H.-H.; Wang, L.; Liu, J.-H.; Huang, X.-J. *J. Phys. Chem. C* **2012**, *116* (1), 1034–1041.
- (107) Afkhami, A.; Soltani-Felehgari, F.; Madrakian, T.; Ghaedi, H.; Rezaeivala, M. *Anal. Chim. Acta* **2013**, *771*, 21–30.
- (108) Cai, W.; Feng, C.; Ma, X.; Chen, M.; Liu, J. *Electrochimica Acta* **2015**, *169*, 424–432.
- (109) Gan, X.; Zhao, H.; Quan, X.; Zhang, Y. *Electrochimica Acta* **2016**, *190*, 480–489.
- (110) Yang, F.; He, D.; Zheng, B.; Xiao, D.; Wu, L.; Guo, Y. *J. Electroanal. Chem.* **2016**, *767*, 100–107.
- (111) Marei, M. M.; Roussel, T. J.; Keynton, R. S.; Baldwin, R. P. *Anal. Chim. Acta* **2013**, *803*, 47–55.
- (112) Salaun, P.; van den Berg, C. M. G. *Anal. Chem.* **2006**, *78*, 5052–5060.

