Microfluidic studies of temperature dependent phase transitions in aerosol droplets

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Priyatanu Roy

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Professor Cari S. Dutcher, *advisor*

June 2021

Acknowledgements

I would like to thank my colleagues, classmates and family for all their support and motivation during my doctoral thesis work. Throughout the PhD program I have interacted with several great minds and gained wonderful experiences.

First of all, I owe my understanding of the fields of atmospheric aerosols and microfluidics, to my PhD advisor, Professor Cari Dutcher. I thank her for excellent technical expertise and knowledge. But above everything I acknowledge her positive reinforcement and efforts to maintain a supportive work environment during a stressful graduate school experience. I thank her for her valuable comments, excellent suggestions and inputs while preparing oral presentations and writing research articles, proposals, and my thesis. Her unwavering optimism and empathy for her students has been invaluable during this time.

I thank Prof. Andrew Metcalf for allowing me to take up the project he had initiated and take it to completion with helpful hints and insights. My collaborators Liora Mael and Prof. Vicki at Grassian University of California, San Diego and Russell Perkins, Kathryn Moore, Tom Hill and Paul Demott at Colorado State University also deserve special thanks for their excellent technical knowhow during projects and help with writing papers and presentations.

I am also grateful to Prof. Chris Hogan who helped shepherd me through a difficult time during graduate school. I must also thank Prof. David Wood for his microfluidics expertise as I learnt a lot from taking his class and through working tips and tricks from his students.

I thank my colleagues from the ditcher group: Shweta Narayan, Shihao Liu, Benjamin Micklavzina and Yun Chen for their help and expertise during execution of my projects. I would also love to thank Vishal Panwar, Athena Metaxas, Tom Neumiller, Lucy Nandy, Nikhil Sethia, Rana Bachnak, Maggie House and Iaroslav Makhnenko for being wonderful labmates.

Finally, I would love to thank my wife, Shannen, for her patience, encouragement, and unwavering belief in me as we made the journey together. And I am grateful to my parents for their love, support and advice throughout different stages of my life including grad school.

This work was supported by National Science Foundation (NSF) through the NSF Center for Aerosol Impacts on Chemistry of the Environment (CAICE), an NSF Funded Center for Chemical Innovation (CHE-1801971). Portions of this work were conducted in the Minnesota Nano Center, which is supported by the National Science Foundation through the National Nano Coordinated Infrastructure Network (NNCI) under Award Number ECCS-1542202.

Abstract

Atmospheric aerosols are suspensions of microscopic chemically complex solid or liquid particles in the atmosphere. The composition and phase of aerosols play important roles in determining radiative forcing, cloud formation, atmospheric chemistry, visibility and human health. Temperature and relative humidity (RH) dependent aerosol particle phase states and phase transitions control interactions with the surrounding gas phase as well as with other particles, and the way the particles evolve with age. Due to acceleration of global warming, there is an urgent need to develop more accurate particle-resolved climate models to improve climate prediction. Aerosols remain the largest source of uncertainty in climate predictions. The main goal of this dissertation is to *develop microfluidic instrumentation to measure aerosol droplet phase transitions such as liquid-liquid phase separation and ice nucleation as a function of temperature and relative humidity.*

First, liquid-liquid phase separation (LLPS) similar to that observed in atmospheric aerosol droplets is investigated with aqueous droplets containing organic and inorganic solutes in a static trap based microfluidic device. LLPS in an aerosol particle directly affects aerosol water uptake and formation of cloud drops. Temperature and RH dependence of LLPS and crystallization for model aerosol droplets with varying composition is explored. It is observed that temperature has a significant effect on some systems while having no effect on others depending on the organic to inorganic ratio (OIR) as well as the identity of the organic and inorganic phases.

Second, a high-throughput droplet freezing counter based on flow-through droplet microfluidics was developed to estimate ice nucleation (IN) in liquid samples relevant to atmospheric cloud droplets. Automated detection and classification of frozen droplets from liquid drops was implemented through machine learning with a deep neural network. A case study with an ideal biological ice nucleating particle (INP), Snomax, was performed. Heating and aging of the sample were also performed to identify the molecular nature of ice nucleation. The device benchmarked well against literature data and provided the highest throughput of any existing INP counters.

Finally, a large array based static trap microfluidic device was implemented to study both RH dependent phase and temperature dependent INP concentration of the same sample *in situ* using bulk sea water and sea surface microlayer (SSML) from a simulated waveflume experiment (SeaSCAPE). This study has implications in identifying origins of INPs in sea spray which make up a significant portion of atmospheric aerosols. Correlation between ice nucleation temperature and residual dry particle morphology showed that the bulk sample had lower INPs than SSML and the residual particles were significantly different between the samples.

In this dissertation, instrumentation development and case studies have been performed to show the suitability of microfluidics as versatile, adaptable and highly customizable devices, which are applicable to studying phases of aerosols and has broad implications in climate science.

Table of Contents

List of Tables

List of Figures

List of Symbols

Symbol	Meaning
V	Droplet volume
h	Channel height
V_0	Initial droplet volume
V_f	Final droplet volume
C_0	Initial droplet concentration
C_f	Final droplet concentration
I	Filtered light intensity
I_0	Incident light intensity
θ	Angle between optical axes of polarizer and analyzer
T	Temperature
$f(T)$	Frozen fraction at temperature T
$N_f(T)$	Number of frozen droplets at temperature T
N_{total}	Total number of droplets
C_m	Snomax concentration in droplet

List of Abbreviations

Abbreviation	Meaning
RH	Relative Humidity
SOA	Secondary Organic Aerosol
CCN	Cloud Condensation Nuclei
IPCC	Intergovernmental Panel on Climate Change
LLPS	Liquid-Liquid Phase Separation
IN	Ice Nucleation
INP	Ice Nucleating Particle
SRH	Separation Relative Humidity
PDMS	polydimethylsiloxane
OIR	Organic to Inorganic Ratio
3-MGA	3-methylglutaric acid
HPLC	High Performance Liquid Chromatography
O:C	Oxygen to carbon ratio
AS	Ammonium Sulfate
PEG	polyethylene glycol
PEGDia	polyethylene glycol diacrylate
PPG	polypropylene glycol
CAD	Computer Aided Design
ERH	Efflorescence Relative Humidity
SLWD	Super Long Working Distance
DNN	Deep Neural Network
PID	Proportional Integral Derivative
DAC	Digital Analog Converter
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor

PWM	Pulse Width Modulated
PCB	Printed Circuit Board
ROI	Region Of Interest
RTD	Resistive Temperature Device
PRTD	Platinum Resistive Temperature Device
HMDS	hexamethyldisilazane
UV	Ultraviolet
PFMD	perfluoromethyldecalin
FC	Fully Connected
FTIR	Fourier Transform InfraRed
ATR	Attenuated Total Reflection
FWHM	Full Width at Half Maximum
SSA	Sea Spray Aerosol
SSML	Sea Surface Micro Layer
SeaSCAPE	Sea Spray Chemistry And Particle Evolution
PTFE	polytetrafluoroethylene
IS	Ice Spectrometer
SSW	Synthetic Sea Water
DOM	Dissolved Organic Matter
mDOM	marine Dissolved Organic Matter
PMMA	polymethylmethacrylate

Chapter 1

Introduction to atmospheric aerosols and microfluidic measurement techniques

Portions of this chapter are adapted with permission from permission from Roy, P., Liu, S. and Dutcher, C. S. (2021) 'Droplet Interfacial Tensions and Phase Transitions Measured in Microfluidic Channels', Annual Review of Physical Chemistry, 72(1), pp. 73–97. doi: 10.1146/annurev-physchem-090419-105522.

1.1 What are atmospheric aerosols?

The Earth's atmosphere, classified into different layers by altitude, varies widely in both composition and temperature. In general, the gaseous component of earth's atmosphere is comprised of mainly nitrogen (~78%), oxygen (~21%) and argon (~1%) with trace gases making up the remaining <1%. Water vapor is the next most abundant component and can reach concentrations of up to 3% locally in the lower troposphere [1].Of the atmosphere layers, it is the troposphere, or the lowest 10-15 km of the atmosphere, that contains 99% of the total ambient water vapor and almost 75% of the atmosphere by mass.

Hence, it is the region where the interactions between suspended atmospheric particles and water vapor result in cloud formation and weather patterns. Temperature variation in the troposphere is about -10°C per 1 km increase in altitude with a corresponding increase in relative humidity (RH). Temperatures in the upper troposphere can be as low as -60°C.

The atmosphere contains more than simply gas molecules. Atmospheric aerosols are microscopic solid or liquid particles suspended in the atmosphere with sizes ranging from nanometers to micrometers [2]. They typically consist of multiple inorganic (ammonium sulfate, nitrates, sodium chloride, other salts, soot, mineral dust) and organic (sea surface microlayer, combustion products) phases and water (liquid, ice) [3]. The composition and size of the particles depend on the particle source, gas phase composition (including water molecules), and temperature.

Aerosols can be directly emitted by varying sources such as the sea surface, dust storms, volcanic eruptions, forest fires, desert winds or human activity. They can also be created in a process called new particle formation, which occurs when gas phase species in the atmosphere chemically react and nucleate to form particles. Once the primary particle has been emitted or created, they can change chemically through oxidative reactions with the surrounding gas phase, evolving into secondary organic aerosols (SOA) which can be composed of organics with different functional groups such as acids, alcohols, esters etc. [4,5].

Regardless of their origin, certain aerosol particles can grow into larger cloud droplets by rapidly taking up water vapor in the atmosphere. When this occurs, the

particles are acting as cloud condensation nuclei (CCN). Whether a particle can become a CCN depends on a multitude of factors, including its size, chemical composition, surface properties and, ultimately, the degree of supersaturation of water vapor in the ambient [6,7]. According to the Clausius-Clapeyron equation [8], saturation vapor pressure of water is inversely proportional to temperature. Which leads to frequent supersaturation of the ambient water vapor at lower temperatures in the troposphere and results in vapor condensing on CCNs [6,9–11] and their growth into liquid cloud droplets. However, if the temperature is low enough for ice to form, then these droplets turn into ice particles which become ice clouds [12].

Finally, a purely liquid aerosol droplet can also undergo liquid phase separation [13–16] i.e. separation of a single mixed liquid phase droplet into two or more immiscible liquid phases. The phase state of a particle or droplet can alter its surface and bulk properties, including the ability to uptake water, atmospheric reactivity as well as interaction with other particles and gases in the atmosphere. Understanding the dynamics governing these liquid-liquid phase separation (LLPS) events is therefore also critical to understanding the role of aerosols in key atmospheric processes.

1.2 Aerosol impacts on climate and human health

Aerosols play a major role in our current climate and how our climate changes. They affect the climate either directly or indirectly, so called the “direct effect” and “indirect effect”. Direct effects of aerosols contribute to the planet’s radiative balance by absorption and scattering of solar irradiation [17–19], while indirect effects include cloud

formation and cloud-aerosol interactions which impact cloud microphysical processes. In the first Intergovernmental Panel on Climate Change (IPCC) assessment report (1990) [20], aerosol effects were mentioned but the theory and measurement techniques were not advanced enough to estimate the magnitude or even determine the net direction of global temperature change due to these effects. Even though the number of studies incorporating aerosols in the climate models and estimates of their cumulative effect have grown exponentially in the intervening years, Figure 1-1, created with data from the fifth IPCC assessment report (2013) [21] shows that aerosol direct and indirect effects and cloud response still remain by far the largest sources of uncertainty and speculation [22] in both global and regional models [23].

Beyond controlling the climate, aerosols can also affect visibility, air quality and human health. For example, locations with a large density of aerosols in the atmosphere has been shown to result in higher than average rates of heart problems [24], lung cancer [25] and miscarriages [26]. And more recently, airborne transmission of the SARS-CoV-2 virus in aerosol droplets has received worldwide attention due to its potency leading to a pandemic [27]. Considering these factors, there is a need for improved understanding of aerosol processes, specifically the effect of particle composition on the phase state and the direct and indirect effects on the atmosphere.

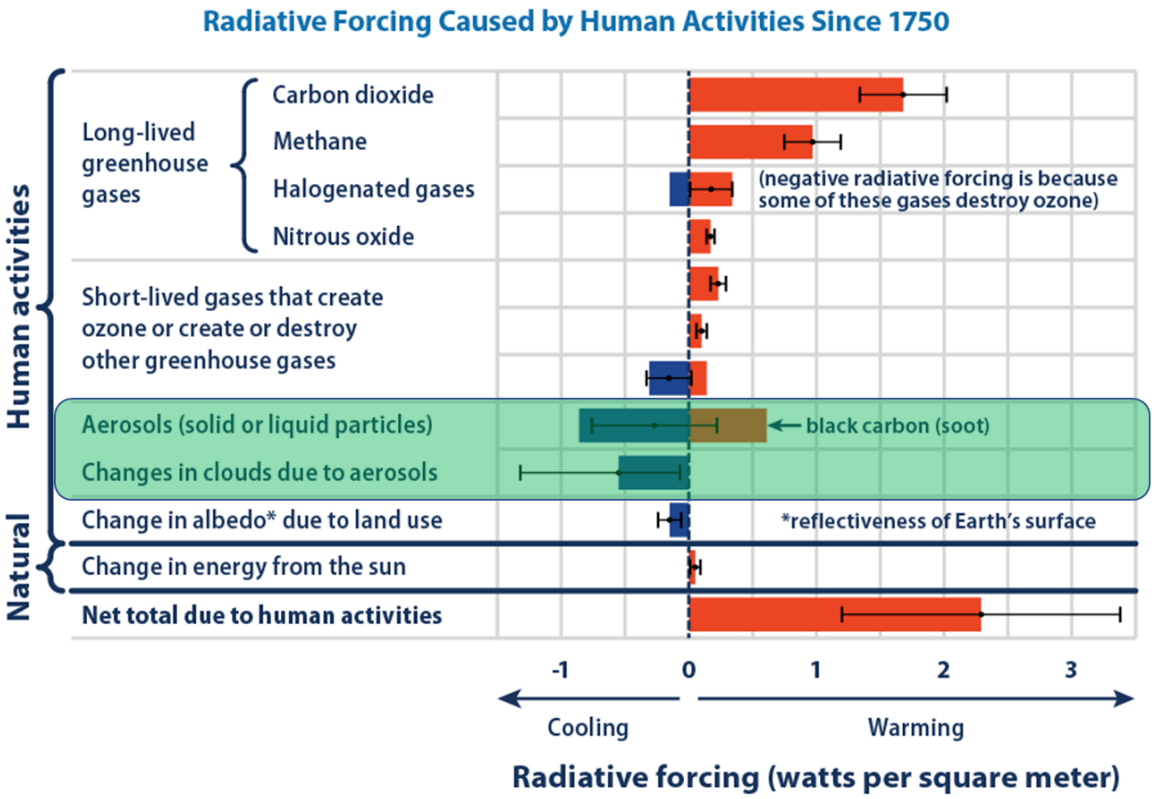


Figure 1-1: Summary of radiative forcing caused by human and natural activities from 1750-2013. Data source: IPCC fifth assessment report (2013) [21]. Highlighted in green are contributions to aerosol total forcing both directly (radiative forcing) and indirectly (cloud properties and lifetimes), which remain the largest sources of uncertainty in atmospheric radiative forcing.

1.3 Aerosol phase states and ice nucleation

1.3.1 Phase transitions: Efflorescence, deliquescence, and liquid-liquid phase separation (LLPS)

Aerosol particles can exist at multiple phase states and morphologies depending on ambient conditions [28] which affects their interaction with the atmosphere. For example, a completely liquid particle when exposed to very low RH turns into a dry particle and in a process called efflorescence. The opposite process of a dry particle taking up water vapor from the environment and becoming a liquid droplet is termed as deliquescence, occurring at a higher RH than efflorescence. The processes are shown in Figure 1-2. The red and blue arrows show the particle drying and wetting respectively. In the region from close to 100% RH to around 75%, the two pathways overlap, and this shows the extents of the solubility of the salt. But during drying, the droplet can get smaller, i.e. supersaturated with respect to the salt before it undergoes efflorescence or solidification at a very low RH. During the wetting process, the solid particle stays solid until the droplet reaches a high RH. This gap between the two processes is a metastable region which is a characteristic of the length scale and composition of the aerosol particles.

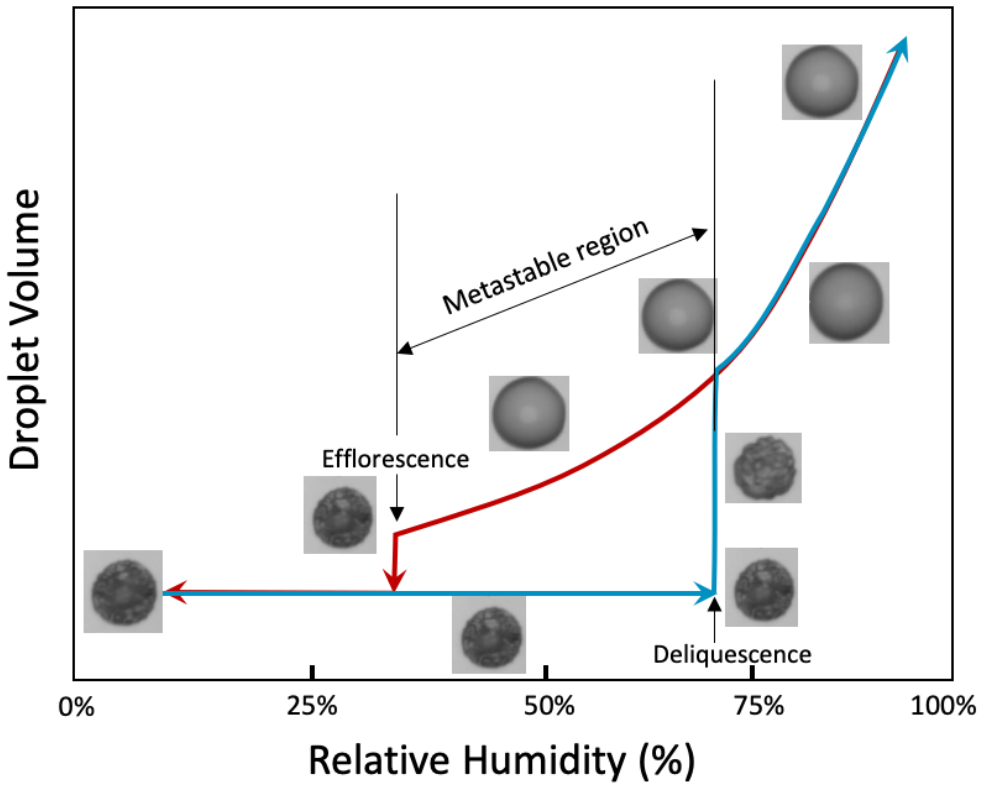


Figure 1-2: Aerosol phase states as a function of relative humidity for a droplet containing water and ammonium sulfate. While drying, the droplet enters a metastable state where it is supersaturated with respect to the salt.

Other than liquid to solid transitions, aerosols can also undergo liquid-liquid phase separation (LLPS) [13]. Since aerosols contain multiple organic and inorganic species, with reduced water content, multiple immiscible liquid phases can exist, and they can separate out into visually distinct phases. This transition can occur when all of the phases in the particle are still completely liquid or in a more complicated manner where some species in the particle has already effloresced and become solid while the rest is still liquid [29,30]. Usually, the immiscible phases are either organic-rich and inorganic-rich. The morphology is governed by the sizes of the interfaces and the respective surface or interfacial tensions

[31–33], with the organic-rich phase typically forming a shell or lens around the inorganic-rich phase droplet.

Phase is a key parameter in determining an aerosol particle's optical properties and its ability to act as a CCN. For example, LLPS in an aerosol containing a mixture of organic and inorganic chemicals impacts the way it directly absorbs and scatters light by changing optical properties such as extinction cross section and single scattering albedo [34]. It also affects the particle's interaction with gas species and water vapor in the atmosphere and growth into a cloud droplet. Formation of an organic coating after phase separation leads to a decrease in the reactive uptake of N_2O_5 thus reducing the rate at which the particle ages in the atmosphere. Traditionally global climate models have treated the mixing state of aerosols in an overly simplified manner, i.e. either by simulating mixing or conversion of an hydrophobic particle to a hydrophilic one by using a time dependent exponential decay [35] or in some cases correlating bins of composition [36] to approximate mixing states, assuming that all particles within a bin have the same mixing state. For optical properties, the refractive index of a particle is calculated based on the assumed well-mixed or core-shell morphology along with an assumption of the particle being spherical and the Mie theory [37] is used to calculate the single scattering albedo.

Currently, there is an increased focus on particle-resolved models (such as PartMC-MOSAIC [38] or its modifications like WRF-PartMC-MOSAIC-SCM [39] or MOSAIC-mix [40]), to improve the accuracy of the mixing state of aerosols depending on the atmospheric ambient conditions. However, the effects of temperatures on phase separation and phase transitions in general in aerosols has not been widely studied and hence, ***it is essential to***

study the phase transitions of aerosols in tropospherically relevant temperatures (down to -40°C). The assumption of internally mixed particles can lead to an overestimation of optical absorption coefficient by almost a factor of two [41] and errors as high as 34% in activated cloud droplet numbers [42]. The tools, methods and results developed in this thesis can potentially improve the treatment of the aerosols in these particle-resolved models. Particularly, it can provide information about what fraction of the particles undergo LLPS as a function of temperature, composition (overall organic to inorganic ratio, organic oxygen to carbon ratio, and salt type).

1.3.2 Ice nucleation (IN)

Ice nucleation (IN) is a highly atmospherically relevant phase transition in cloud droplets and usually occurs at temperatures well below 0°C which is the thermodynamic freezing point of pure water. Clouds can be supercooled down to -38°C before ice nucleates in water droplets homogeneously [43]. In addition, foreign solid particles or macromolecules internal to the droplets can act as heterogeneous nucleation sources or ice nucleating particles (INPs) by facilitating ice formation at temperatures anywhere in the range of -38°C to 0°C [12]. Commonly, INPs are submicron particles generated from terrestrial biological, urban, mineral dust or marine emission sources [12,44–48]. INPs constitute less than 1 in 10^6 of all aerosol particles. These particles can nevertheless exert a disproportionate influence on the observed precipitation rates as well as indirectly affect radiative forcing by forming mixed or ice phase clouds [49]. However, the impact of increased INP concentration on ice cloud formation is not intuitive, because lowering of total frozen droplets at higher INP concentrations is possible due to heterogeneous ice formation at

higher temperatures which competes with more abundant homogeneous ice formation at lower temperatures in aerosols not containing INPs [50].

INPs nucleate ice via different pathways: immersion, deposition, condensation, and contact [51]. Immersion freezing, where crystals nucleate from the surface of a solid particle internal to a liquid droplet, has been theorized to be the dominant mode for lower tropospheric mixed phase clouds [47,52,53], while deposition nucleation is more dominant in upper tropospheric ice clouds [54]. Contact freezing is rarer and occurs mainly in volcanic eruption events where a lot of solid particles are ejected into supercooled clouds [55]. Presence of condensation nucleation in the atmosphere and whether it is substantially different from deposition nucleation is still under debate [56]. Commonly, INPs are submicron particles generated from terrestrial biological, urban, mineral dust or marine emission sources [12,44–48]. Out of these modes, immersion freezing is the most relevant in terms of IN in supercooled clouds, making up over 85% of these events [57].

Like the simplified treatment of aerosol phase above its freezing point, current models of radiative forcing due to ice particles either depend only on ambient temperature and ice saturation ratio (a thermodynamic parameter) [58] or use laboratory results on specific ideal particle types to extrapolate to IN concentrations in the atmosphere. Both methods lead to orders of magnitude errors in prediction of actual INP concentration when compared to field data. ***Since oceans cover the dominant portion of earth's surface area, there is a need to study INP emissions from marine sources*** [59]**.** The temperature controlled static and flow-through microfluidic ice nucleation counter devices developed in this thesis will help simplify and accelerate fundamental studies on INPs collected from field sources.

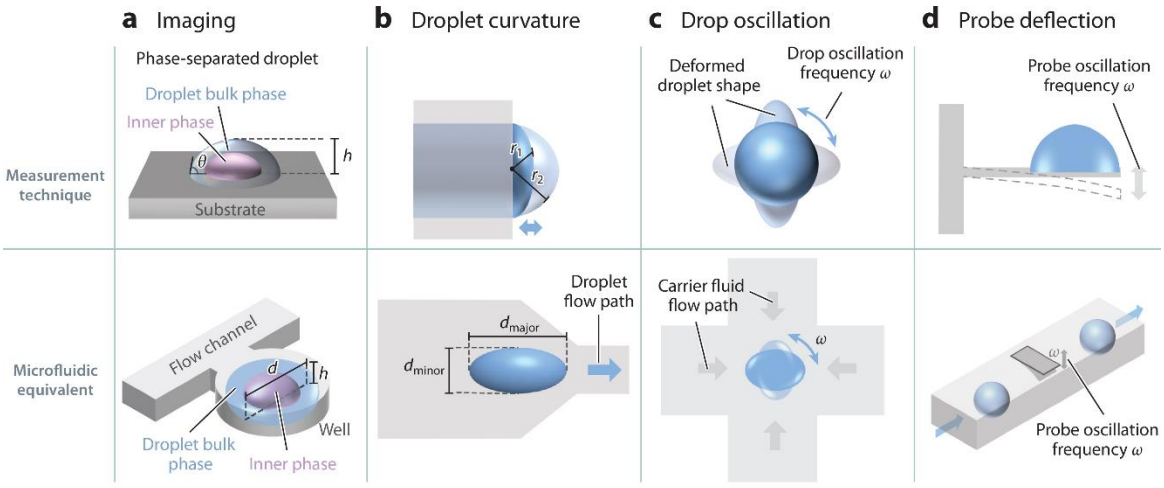
1.4 Introduction to microfluidic devices

Predictive modeling of aerosol processes requires precise knowledge of chemical and physical properties of individual aerosol particles such as surface tension [60], viscosity [61], and phase [13]. For accurate measurements, shrinking down bulk measurement techniques is often necessary to observe isolated single [62] or multiple particles at the relevant length scales [63]. Measurements of physical properties of fluids are traditionally done in the bulk with large sample volumes. Recent advancements in droplet generation, manipulation and observation techniques have extended these measurements to the micrometer length scales [64,65]. This microscopic length scale becomes relevant in complex and multiphase fluid systems such as aerosols, emulsions, suspensions and biological fluidic mixtures, where measurement of bulk fluids is not often feasible either due to the lack of substantial sample volumes, or due to the small characteristic length scales governing the dynamics of the system. Atmospheric aerosol particles and droplets fall in both categories – their sample volume is often extremely limited and their characteristic length scale ranges from micrometers to nanometers [66].

Microfluidic devices are a special class of liquid flow channels with micrometer length scales in one or more directions and are usually fabricated in some transparent material for optical observation. These devices are used extensively in biomedical [67], industrial [68], and environmental monitoring [69] applications as well as in fundamental research on multiphase systems such as emulsions [70,71]. Advantages of such small length scales and confined geometry include shorter thermal and material transport length scales which allows access to flow regimes usually not accessible in bulk scale experiments.

Additionally, the small sample volume required, the ease of design, and the affordable manufacturing cost of these devices make them ideal for applications where the sample is much rarer and more precious than the sampling equipment.

Various physical principles of droplet measurements are shown in Figure 1-3. The top row shows microscale procedures and the bottom row shows their corresponding microfluidic-based equivalent approaches for high-throughput or multiplexed measurements. Figure 1-3a, shows measurements on sessile drops on a solid substrate, where imaging techniques are used to measure phase state or viscosity of the droplets. Figure 1-3b shows force tensiometry methods where droplet curvature or overall distortion from circularity is used to infer surface tension and viscosity. Figure 1-3c shows droplet oscillation methods where the physical properties are calculated from the oscillation frequency and amplitude. Finally, Figure 1-3c shows deflection-based methods where external probes are used for physical property measurements.



Roy P, et al. 2021
Annu. Rev. Phys. Chem. 72:73–97

Figure 1-3: (Top row) Surface tension and phase measurement principles at the microscale including (a) imaging-based, (b) droplet curvature– or overall shape–based, (c) drop oscillation–based, and (d) probe deflection–based methods. (Bottom row) Representative equivalent microfluidic setups. θ, d, and h are the droplet contact angle, diameter, and height, respectively, for imaging-based methods. r1 and r2 are the droplet radii corresponding to different upstream pressures. d_{major} and d_{minor} are the major and minor axes of a deformed droplet as it enters a microfluidic constriction during surface/interfacial tension measurement for droplet curvature–based methods. ω refers to oscillation frequency of either the droplet or a probe in drop oscillation– and probe deflection–based methods. Reprinted with permission from Roy, P., Liu, S. and Dutcher, C. S. (2021) 'Droplet Interfacial Tensions and Phase Transitions Measured in Microfluidic Channels', Annual Review of Physical Chemistry, 72(1), pp. 73–97. doi: 10.1146/annurev-physchem-090419-105522. Copyright 2021 Annual Reviews.

1.4.1 Development of microfluidic devices for aerosol measurement

Microfluidics is still in its nascent stages in aerosol science applications [14,30,32,72–76]. Devices already developed range from microfluidic droplet generators for off-chip analysis [73,77], and on-chip analysis with static arrays of droplet traps [78,79] to flow-through channel based devices [74,80]. A comparison of different conventional instruments used for measuring aerosol properties in terms of their attributes is given in Table 1-1. Here, the color green indicates suitable, orange indicates mediocre and red indicates challenging. It is apparent that, while microfluidics has proven itself to be an indispensable tool in the field of biomedical application, it still has stiff competition in aerosol science as it is not widely adopted yet, and versatility of the method remains unproven as there are still many potential challenges to overcome.

Table 1-1: Microfluidics compared against some traditional aerosol measurement techniques

Instrument	Size	Cost	Complexity	Throughput	Versatility	Ref.
Droplet on slide						81
Differential mobility analyzer						82–84
Optical trap						85
Cloud chamber						86–88
Electrodynamic balance						89
Cloud condensation nuclei counter						90,91
Microfluidics					Early stages	30,32,72,92

However, one big advantage of microfluidics is lower cost, smaller space requirements and quick prototyping compared to building flow devices with multiple flow lines and chambers. For example, precise temperature and RH control in aerosol instruments becomes costly when a large space needs to be controlled compared to the small volume occupied by the microfluidic chip. As a result, ingenious microfluidic devices can be designed so that instead of competing with these instruments, microfluidics can fill in the gaps in the studies where traditional aerosol instrumentation prove difficult to operate, be it due to space, cost or time constraints.

The goal of this thesis is to design, build and benchmark custom temperature dependent microfluidic devices for aerosol applications with case studies in model and

real aerosols. The microfluidic platforms were adapted from literature and modified to fit the requirements of the study. This thesis is organized into six chapters. Chapter 2 details the adaption of microfluidic static traps to study temperature and composition dependent liquid-liquid phase separation in model ternary aerosol droplets as well as nucleation rate dependent crystal formation in model binary aerosol droplets. The temperatures studied are above the freezing point of the aqueous droplets. Chapter 3 focuses on the development of a high-throughput droplet freezing microfluidic device and a custom temperature-controlled platform with sensors for precise temperature measurement of the droplets. Also detailed are two methods for automated droplet freezing detection including one state-of-the-art deep neural network-based image recognition algorithm. Chapter 4 elaborates on a case study of the ice nucleating property measurement of a biological ice nucleating particle Snomax and delves deeper into the molecular mechanism behind its efficient ice nucleation behavior. Chapter 5 details the development of a large array based microfluidic static trap device to study combined ice nucleation and dehydration related efflorescence of individual droplets of samples collected from a marine source. The instrument is benchmarked against a more established ice nuclei counter operating at a different length scale of sample. Dehydrated droplet morphology is correlated with ice nucleation temperature using the device. Finally, in Chapter 6, preliminary designs of a simpler flow-through droplet freezing counter, a relative humidity-controlled chamber for particle deliquescence experiments and a modified static well device which allows for in situ chemical analysis for studying complex organic marine samples are discussed.

Chapter 2

Temperature dependent phase transitions in model aerosol mimics

Portions of this chapter are adapted with permission from Roy, P., Mael, L. E., Makhnenko, I., Martz, R., Grassian, V. H., Dutcher, C. S. (2020) 'Temperature-Dependent Phase Transitions of Aqueous Aerosol Droplet Systems in Microfluidic Traps', ACS Earth and Space Chemistry, 4(9), pp. 1527–1539. doi: 10.1021/acsearthspacechem.0c00114.

2.1 Introduction

Aerosol in the atmosphere can exist in multiple phase states including liquid, semi-solid and solid [93,94]. Particles can transition among these states depending on their chemical composition along with the ambient conditions such as temperature and relative humidity. LLPS and efflorescence/deliquescence of particles play a crucial role in aerosol properties like their optical properties [95,96], hygroscopicity [93,97] and heterogeneous chemistry including atmospheric aging or uptake of gas phase molecules like N_2O_5 [98,99]. Additionally, the phase state of a particle can also influence its surface properties and hence its ability to activate into a CCN [100] or ice nucleating capabilities [101].

Methods to quantitatively measure the phase state of aerosol particles include bulk measurements in cases where separation relative humidity (SRH) of the particle is higher

than the deliquescence relative humidity. However, this is a rare occurrence and supersaturation is needed to study LLPS in most relevant aerosol systems. Supersaturation can be achieved in single particle phase state measurements, at the supermicron scale using optical microscopy on droplets on substrate [15], electrodynamic balancing [102], or optical traps [103]. Single particle phase at submicron scale can be studied using scanning electron microscopy with energy dispersive X-ray analysis [104], transmission electron microscopy [105] and scanning transmission X-ray microscopy/near edge X-ray absorption fine structure methods [106]. Phases of larger populations of single particles can be measured directly using particle water uptake and growth in hygroscopicity tandem differential mobility analyzers [82–84], microresonators [107], interdigitated electrodes [108], quartz crystal microbalances [109] and indirectly by measuring particle bounce using impactors [110–113]. For a review of comprehensive physical and chemical characterization techniques in submicron aerosols, see McMurry et al. [114]. The substrate used in contact-based methods can influence the particle morphology during the initial impaction as well as phase transitions. Hydrophobic surfaces are commonly used to minimize this effect. Single particle trap-based methods often require highly complex instrumentation to obtain measurements and can become cost prohibitive and time consuming.

Recently, microfluidic measurements have been used to look at statically trapped supermicron (pico-to-nanoliter volume) single droplet phase transitions [14,30,115] and ice nucleation [76]. Aerosol droplets are trapped in microfluidic wells which are surrounded by a thin layer of carrier fluid and sandwiched between highly permeable poly-dimethylsiloxane (PDMS) and non-permeable glass layers. Morphological changes

including LLPS and efflorescence can be observed as droplets evaporate slowly over time in a quasi-equilibrium manner and the changes are recorded in a time-lapse video. Microfluidics offers an advantage over the substrate-based methods since the droplets are encased in layer of liquid and hence reduce the probability of heterogeneous nucleation from a solid substrate. It also provides a 2-D Hele-Shaw type geometry which makes it easier to determine LLPS and resultant morphology using optical microscopy. In addition, there is usually no need for external relative humidity control during an experiment as the drying process occurs slowly until the droplet RH equilibrates with the surrounding RH. By careful design of flow channels and wells it is even possible to rehydrate droplets or control the dehydration and rehydration rate by flowing water through channels adjacent to the trapped droplets [115,116]. The disadvantages of this method are that droplet dehydration usually takes a long time, on the order of days. So, collection of a large number of data points can be challenging. Also, the droplet RH needs to be calculated based on droplet concentration using equilibrium thermodynamic models as direct measurement of water activity or RH in a microfluidic well is not readily feasible.

Factors including temperature, as well as the initial composition of a mixed organic-inorganic liquid droplet itself can decide whether an aerosol droplet will undergo LLPS and at what SRH. Among the chemical composition aspects, a universally acknowledged key factor is the oxygen to carbon ratio (O:C) of the organic component [117–119]. Other factors which have been studied are sulfate ratios, molecular weight of the organic [120], multiple organics [121], and secondary organic aerosols created from atmospheric aging [122] of the primary aerosols. Certainly, not all systems undergo LLPS before efflorescence. You

et al. showed particles with O:C below 0.57 readily undergo LLPS, whereas particles with O:C between 0.83 and 0.57 undergo LLPS only for certain organics, and particles with O:C above 0.83 do not undergo LLPS [120]. A recent study by Ott et al. [123] on ternary aerosols with added sucrose has expanded the upper limit of the O:C ratio for LLPS to 0.92. Results from more complex systems with more than three components and field samples also agree with this narrow range of O:C necessary for LLPS [117,121,122].

However, while the effect of chemical composition on LLPS is becoming more extensively understood, the effect of temperature on LLPS is less well-characterized. It was theorized that droplet viscosities at lower temperatures in the troposphere (<20°C) would increase and lead to inhibition of phase separation [124]. In contrast, a recent study on the effect of low temperatures down to -29°C showed that for most of the systems that do undergo LLPS at room temperature, inhibition of LLPS was not noticeable, i.e. SRH was not dependent on the droplet temperature. Furthermore, in some cases the SRH even increased with reduction in temperature[120]. Organic to inorganic mass ratio (OIR) of the aerosol also plays a part in the morphology of these phase transitions and generally varies between 0.2 to 3.5 in atmospheric particles [125–127].

In addition to LLPS, the crystal structure and subsequently the morphology of aqueous droplet upon efflorescence has also been known to depend upon temperature and drying conditions, specifically, the dehydration rate. Sodium chloride forms two possible crystal structures upon efflorescence, namely, anhydrate ($NaCl_{(s)}$) and dihydrate ($NaCl{\cdot}2H_2O_{(s)}$). The equilibrium phase diagram shows that at temperatures above 0°C, the $NaCl_{(s)}$ form and below 0°C, the $NaCl{\cdot}2H_2O_{(s)}$ form is thermodynamically favorable [128].

However, in experiments on droplets on substrate, it was shown that only $NaCl_{(s)}$ is formed at temperatures as low as -20°C [129]. In another study [130] the range of temperatures where the transition from $NaCl_{(s)}$ to $NaCl{\cdot}2H_2O_{(s)}$ happens was found to be between about -20°C, where 0% of the particles were in the dihydrate form, and -32°C, where 100% of the particles observed were the dihydrate. Another study on particles suspended in an electrodynamic balance [101] reported the transition to be in a similar temperature range. Some studies have also reported a different efflorescence crystal structure in seawater depending on the rate of decrease of RH due to different number of water molecules trapped in the dried particle [131,132].

Here, we investigate a range of model aerosol systems in the range -20°C to +25°C which covers a significant portion of tropospheric temperature (-60°C to +25°C) for O:C ratios between 0.38 to 0.67 which are below the critical O:C < 0.92 for LLPS. We varied the organic to inorganic ratio (OIR) of the systems to estimate the effect of the organic and inorganic components on the LLPS morphology temperature-dependence. Systems studied consisted of either ammonium sulfate or sodium chloride as inorganics and organic acid and polyalcohol organics in an aqueous solution with OIRs ranging from 1:10 to 10:1. We also studied binary solution of sodium chloride and water to find the effect of temperature and drying rate on the efflorescent crystal structure. This work uses a novel and low cost, quasi-equilibrium technique to shed new light on how temperature and the OIR of these systems affect the phase separation morphology as well as the total solute concentration or conversely the SRH at which these systems undergo phase separation.

2.2 Materials and methods

2.2.1 Chemicals used

Solutions of ammonium sulfate (Fisher Scientific) and sodium chloride (Fisher Scientific) were prepared by dissolving crystals into high performance liquid chromatography (HPLC) grade water (Fisher Scientific). Organics with O:C ratio below 0.83 were mixed with this salt solution to make experimental samples. The organics used were 3-methylglutaric acid (3-MGA) (Tokyo Chemical Industry) with O:C = 0.67, polyethylene glycol (PEG) (Molecular weight = 200 g/mol) (Sigma Aldrich) with O:C = 0.63, polyethylene glycol diacrylate (PEGDia) (Sigma Aldrich) with O:C = 0.50 and polypropylene glycol (PPG) (Sigma Aldrich) with O:C = 0.38. The organics were mixed with the salt solution in three different ratios by mass, namely OIR 1:10, 1:1 and 10:1.

Experiments for the ammonium sulfate (AS) and 3-MGA system were performed at four different temperatures: +25°C, +5°C, -10°C, -20°C and three different OIRs: 1:10, 1:1 and 10:1. The same initial concentration C_0 for a given mixture system was used at all temperatures. In the AS+3-MGA system C_0 values were 1.42, 2.33 and 0.92 M for OIR 1:10, 1:1 and 10:1 respectively. Ammonium sulfate and other organics were studied at +25°C and -20°C at OIR 1:1. C_0 was 2.56 M, 0.98 M and 0.67 M for AS+PEG, AS+PPG and AS+PEGDia systems respectively. Sodium chloride and 3-MGA systems were studied at all three OIRs but only at +25°C and -20°C, with C_0 of 1.67 M, 1.15 M and 0.56 M for OIR 1:10, 1:1 and 10:1 respectively.

2.2.2 Microfluidic device fabrication

The microfluidic well design used for this study was adapted from [133] and has been used previously in a study [14] from our group to examine LLPS and efflorescence in different models and real systems at room temperature. For this work, a new device was designed by replicating this single well into a 3x3 grid on a single chip. Each device consists of a single inlet and outlet, in series with the microfluidic wells (Figure 2-1a). Each well consists of a 200µm wide inlet, a 450µm diameter well and a 40µm wide restriction. Additionally, there is a bypass path for the carrier oil to pass through. Heights of all individual wells were measured to be between 85 to 95µm with a P16 surface profilometer (KLA Tencor).

To fabricate the device, the photomask shown in Figure 2-1a was designed in a computer aided design (CAD) software (AutoCAD, Autodesk Corp.) and printed on a vinyl transparency (CAD/Art Services Inc.). A device master on a 4-inch diameter silicon wafer was fabricated using well-established photolithography techniques 65,66 using the photomask. Subsequently, the microfluidic device was made from pouring a mix of uncured PDMS (Sylgard 184 Silicone Elastomer, Dow Corning Corp.) and a linking agent in 10:1 ratio by weight into the mold. Then the mold was degassed before baking in a 75°C oven to cure the PDMS. Next, the PDMS device was cut out from the mold, and holes were punched for the inlet and outlet. Finally, the device was sealed to a 25x75x1mm glass slide (Corning) after oxidative plasma etching of both the glass and PDMS surfaces.

2.2.3 Droplet loading

The as prepared mixed organic and inorganic aqueous sample was injected into the device to fill the microfluidic channel completely, including the wells and the bypass paths. Silicone oil (Sigma Aldrich) was then carefully injected into the device which scavenged all the excess sample from the bypass channels and pushed it out through the outlet. The sample droplet already inside the well was not pushed out, since the restriction in front of the sample was small enough to prevent the droplet from squeezing through. This resulted in droplets trapped inside the wells and the rest of the channel being filled with silicone oil.

2.2.4 Dehydration experiment

The device was placed on the cold plate of a temperature-controlled stage (LTS 420, Linkam Scientific, UK) and the lid was closed. The cold stage was placed on the XY stage of a reflective microscope (SZX10, Olympus) and jpeg images were captured at 1600x1200px using an Ace 1600gm area scan camera (Basler) every 10 seconds throughout the duration of the experiment. Water from the droplets slowly diffused through the silicone oil and pervaporated through the PDMS during an experiment. A schematic of the droplet trapped inside the device is shown in Figure 2-1b.

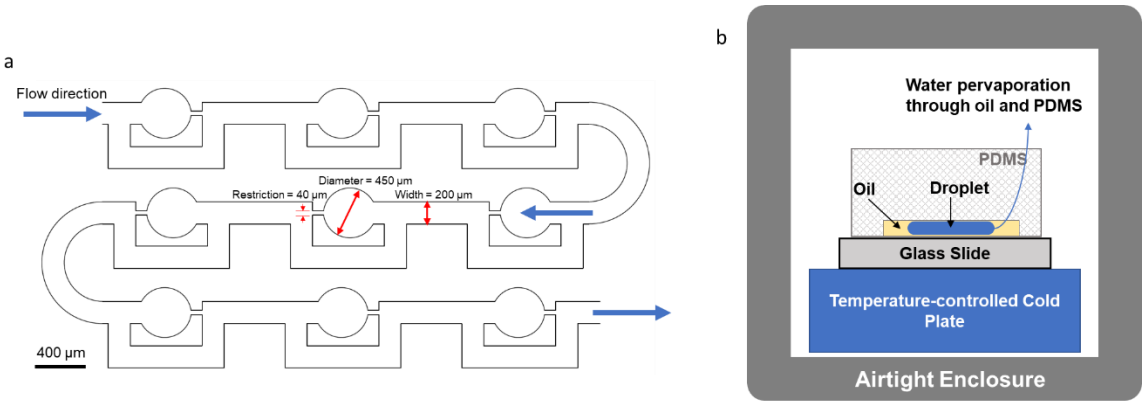


Figure 2-1: (a) Microfluidic static well device schematic. (b) Schematic of microfluidic device inside cold stage showing pancake shaped droplet surrounded by silicone oil in a PDMS device and the water pervaporation path through the oil and PDMS layers. Device fabrication details are provided on page 22.

2.2.5 Droplet temperature simulation

Before fabricating the devices and running experiments, a numerical heat transfer simulation of the microfluidic device and the cold plate was performed to ensure that there was not a large amount of temperature variation inside the droplet and that the droplet temperature closely approximated the temperature of the cold stage. A 3D finite element model was constructed in COMSOL Multiphysics. The glass substrate was placed in thermal contact with the cold plate and meshing was performed with free tetrahedral elements using the finest density settings in the software. The number of mesh elements was 1,218,959, with a minimum element quality of 0.06136 and average element quality of 0.6559 using a skewness quality measure. The geometry of the complete domain is shown in Figure 2-2a and the mesh is shown in Figure 2-2b. Note that three separate

devices on the same slide was fabricated since the individual device size was small and by repeating devices on single slide, more of them could be fabricated at one go.

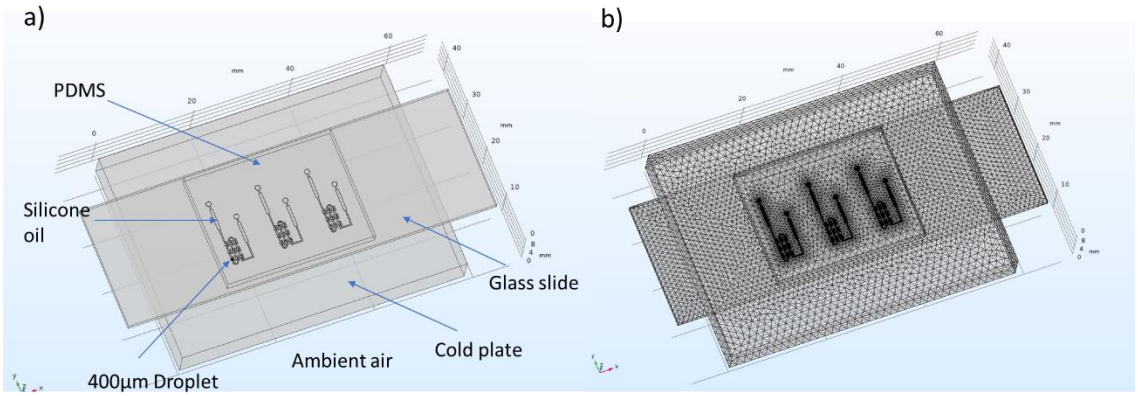


Figure 2-2: (a) Geometry domain for COMSOL 3D heat transfer simulation of microfluidic static wells. (b) Free tetrahedral mesh of the domain.

The boundary conditions for this simulation as follows: A constant temperature was applied to all surfaces of the cold plate. The device was placed in contact with the cold plate surface at -20°C, the lowest temperature used in subsequent experiments and thus simulating worst case scenario for any temperature variation. All exposed surfaces had natural convection with a heat transfer coefficient of h = 10W/m^2 and an ambient air temperature of 20°C.

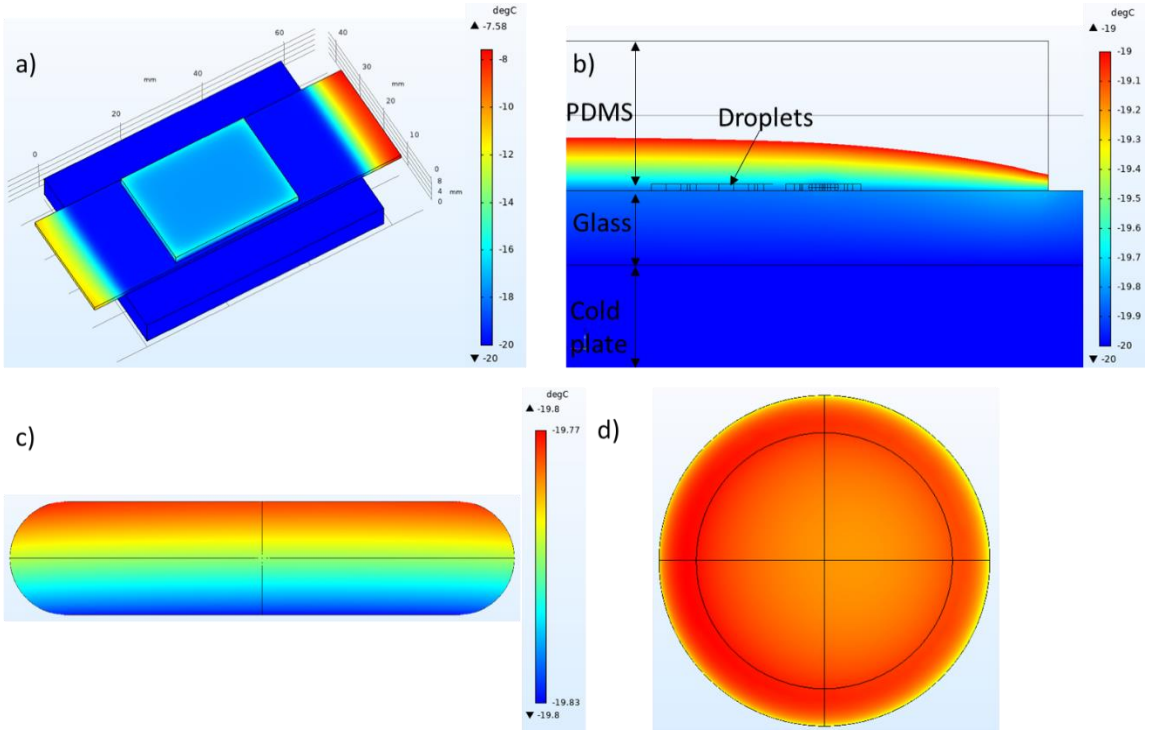


Figure 2-3: Results of steady state heat transfer simulation on the static well 3D model in COMSOL. (a) The whole domain showing the overall temperature variation in the microfluidic device. (b) A cross section through the centerline of the device along the long axis of the device.

Result from the simulation is shown in Figure 2-3. The edges of the glass slide overhanging the cold plate are warmer than the center of the slide where the PDMS device is located. Figure 2-3b shows the cross section of the device, the displayed contours have been clipped at -19°C. Figure 2-3c-d show the temperature distribution across the surface of a droplet. The droplet temperature varies between -19.77°C at the top to -19.83°C at the bottom. This essentially means that the variation inside the droplets is much lower than 1°C in the most extreme case and hence for all experiments, the temperature of the cold plate was reported as the droplet temperature. Please refer to Appendix A for more

numerical simulations with different thermal contact gaps between the slide and the cold plate and with different ambient air temperatures.

2.2.6 Droplet volume and water activity calculation

Trapped droplets were pancake shaped and the volumes, V, were calculated using the method described in.

$$V = \frac{\pi h^3}{6} + \frac{\pi h}{4}(D - h)\left(\frac{\pi h}{2} + D - h\right) \qquad \text{Equation 2-1}$$

In Equation 2-1, D is the diameter of the droplet, and h is the channel height. The volume of the droplet is used in Equation 2-2 to calculate the solute molarity, C (M or moles / liter), of the droplet at any point during the experiment [14] by assuming only water leaves the droplet during the experiment

$$V_0 C_0 = V_f C_f \qquad \text{Equation 2-2}$$

where the subscripts 0 and f correspond to measurements taken initially and finally at the phase transition point respectively. The droplet solutions were always prepared with a fixed initial concentration at a given OIR and to allow for a comparison between different temperatures at the phase transition points. The RH is not being directly measured; instead the droplet solute concentration at what RH the phase transition would occur, via a thermodynamic model [134] that relates concentration to water activity, similar to Nandy and Dutcher [14] in an identical microfluidic setup. Using this method, they calculated efflorescence relative humidity (ERH) of 33-35% for binary ammonium sulfate solution at

room temperature matches closely to literature data [135,136]. We followed a similar methodology to calculate the RH of phase transitions in ternary droplets using the thermodynamic models E-AIM [137,138] (http://www.aim.env.uea.ac.uk/aim/aim.php) model II and III and AIOMFAC [139,140] (https://aiomfac.lab.mcgill.ca/model.html) between the initial droplet loading point and the point where the first phase transition (LLPS or nucleation of crystal/efflorescence) occurred. The modeling could not be extended beyond this as the droplet did not remain well-mixed and the composition of the phases were unknown after this transition point. Both models are widely used in aerosol science. E-AIM has been used previously to study mixed organic inorganic systems in [141–144]. AIOMFAC is also often used for studies of mixed systems [143,145–147], with long and medium range interactions between organic and inorganics in aqueous solutions incorporated. For the RH calculation, 3-MGA and other organics were created as new compounds in E-AIM with activity equations defined using the group contribution method from UNIFAC. E-AIM density output was used to convert the solution molarity from Equation 2-2 to molality and mass fraction which was used for calculating the equilibrium the water activity of the droplet, which for a droplet in the atmosphere is equivalent to the corresponding RH of the gas phase and referred to as the RH of the droplet with the assumption of equilibrium. The E-AIM density was used to calculate the mass fraction of solutes and solvents in the droplet phase transition point and used as an input to AIOMFAC. The RH of the droplet was calculated from both thermodynamic models. More details on the calculation methodology and results are reported in Appendix B.

2.2.7 Microfluidic confocal Raman microscopy

For some of the samples, 2D confocal Raman spectroscopy was also performed on an ammonium sulfate and 3-MGA droplet trapped inside microfluidic wells at room temperature to qualitatively map the organic and salt before and after LLPS inside the droplet. The system consisted of an Alpha 300R microscope with a UHTS300 spectrometer and DV401 CCD detector with 600/mm grating (WITec, Germany). A 532nm Nd:YAG laser was used for the excitation with a 100x objective (Nikon Instruments, NY). Each image had a resolution of 58x58 pixels with a pitch of 3μm and an integration time of 0.1 seconds per pixel.

2.2.8 Environmental cell Raman microscopy

For fast drying experiments, a Raman microscope (Horiba, LabRAM HR Evolution) coupled to an environmental cell (Linkam, LTS 120) was used to collect spectra and optical images of substrate deposited sodium chloride (NaCl) particles following deliquescence and efflorescence at both -21.0 and -25.0 + 0.4°C. A detailed description of the experimental set up has been described in Mael et al. [148]. In this experiment, a 10 % (w/w) aqueous solution of NaCl (> 99% Fischer Scientific) was passed through an atomizer (TSI Inc, model 3076) and impacted on to a hydrophobically coated (RainX) quartz substrate (Ted Pella Inc., no 16001-1). The sample was then placed in the environmental cell and first dried under a flow of N2 until the dew point had stabilized (measured using a hygrometer, Buck, CR-4), following which the temperature was reduced to either -21 or -25□C. Once equilibrium had been reached for both temperature and relative humidity in the cell, the relative humidity was increased to 85 + 2% by altering the ratio of wet:dry N2, and the particles were deliquesced. Following this the wet N2 line was turned off and the

samples were rapidly effloresced, which took between 30 to 45 minutes at -21°C and 75 to 90 minutes at -25°C. Once the relative humidity stabilized, optical images of the particle ensemble and spectra of individual particles were collected. Images were collected with an optical microscope (Olympus BX41) and a 100x super long working distance (SLWD) objective. Spectra were taken using a 532 nm Nd:YAG laser excitation source with seven exposures of fifteen seconds each averaged to obtain individual spectra.

2.3 Results

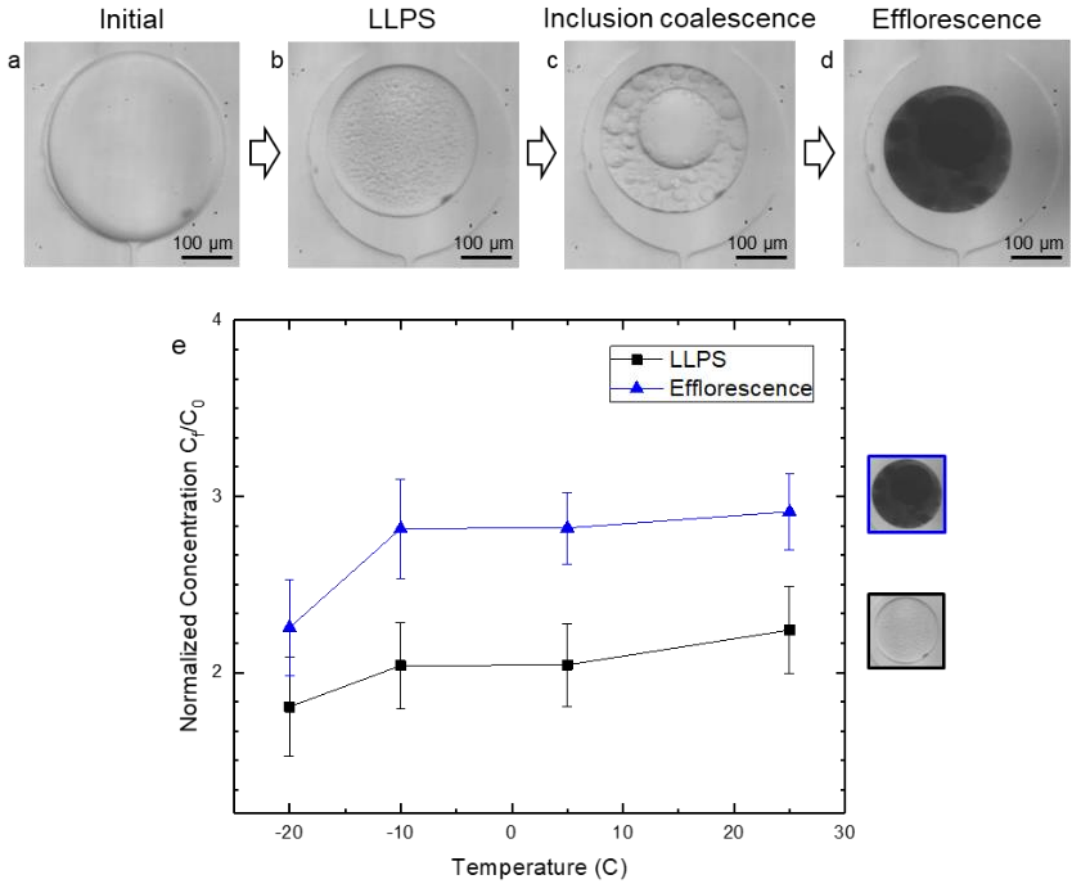


Figure 2-4: (a-d) Typical phase transitions in a aqueous AS and 3-MGA droplet with organic to inorganic ratio (OIR) of 1:1 at 25°C inside a static microfluidic trap. (e) Phase transitions reported as a function of normalized total droplet solute concentration at different temperatures. C_0 was 2.33 M for this OIR.

Droplets consisting of water, AS and 3-MGA mixed with OIR 1:1 undergo LLPS while continuously dehydrating in the microfluidic trap. Shown in Figure 2-4(a-d), following the initial phase transition, the phases undergo a bulk transition into a core-shell morphology with the core comprising of an inorganic-rich phase and the shell comprising of an organic-rich phase. This morphology was also observed in previous work on mixtures of AS and 3-MGA [14]. Finally, the droplets undergo efflorescence at a sufficiently low water content. Similar transitions occur in all experimental temperatures. Figure 2-4e shows the normalized concentration (C_f / C_0) from Equation 2-2. Since the C_0 was constant for all droplets for a given OIR, this figure shows that droplets become comparatively more concentrated at higher temperatures before undergoing phase separation. Later in Figure 2-8, after application of the thermodynamic model calculator, the results are recast in terms of RH. Error bars in all figures indicate the standard deviation of measurement between 3 to 9 droplets.

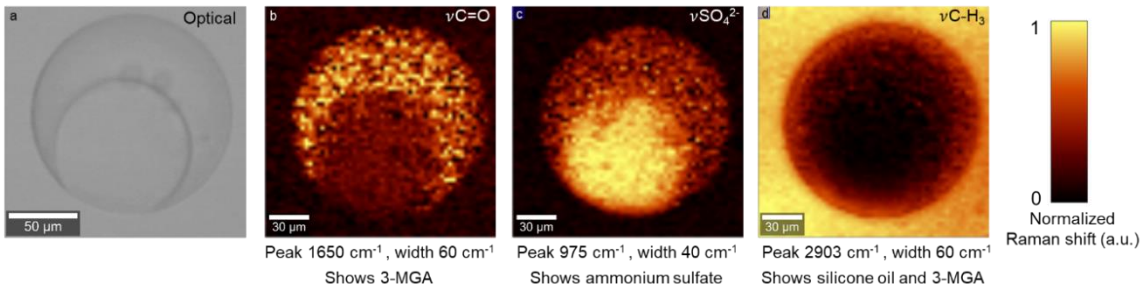


Figure 2-5: (a) Optical image of a phase separated droplet of AS and 3-MGA with OIR 1:1. (b-d) Raman images for different chemical species in the sample. The spectral region used for generating the heatmaps are indicated in the images.

Figure 2-5a shows a bright field image of trapped droplet with ammonium sulfate and 3-MGA and an OIR 1:1 in the Raman setup. The droplet has undergone LLPS with a core-shell morphology. The contour plot in Figure 2-5b shows the normalized sum intensity of a 60cm^{-1} wide band centered around the peak at 1650cm^{-1} (for the carboxylic acid group) from 2D confocal Raman microscopy. The colors indicate that the 3-MGA is distributed mostly in the outer shell phase. The 40cm^{-1} wide band centered around the peak at 975cm^{-1} is indicative of the SO_4^{2-} ion and Figure 2-5c shows that ammonium sulfate is present in a larger concentration in the core structure. Interestingly, the 60cm^{-1} band centered around the peak at 2903cm^{-1} indicates the presence of CH_3 groups and is present is very high concentrations around the droplet and at somewhat lower concentrations inside the outer shell structure of the droplet (Figure 2-5d). The outer high concentration region is indicative of the silicone oil the droplet is immersed in, while the inner low concentration region indicates the CH_3 groups of 3-MGA. The nonuniform halo of CH_3 encroaching inside the circular edge of the droplet in Figure 2-5d is an artifact of the confocal imaging system due to the pancake shape of the droplet, with a circular edge of hemispherical cross section. This causes the edge to have silicone oil at the top and bottom of the image focal plane and it shows up as nonuniformity of the signal around the edges.

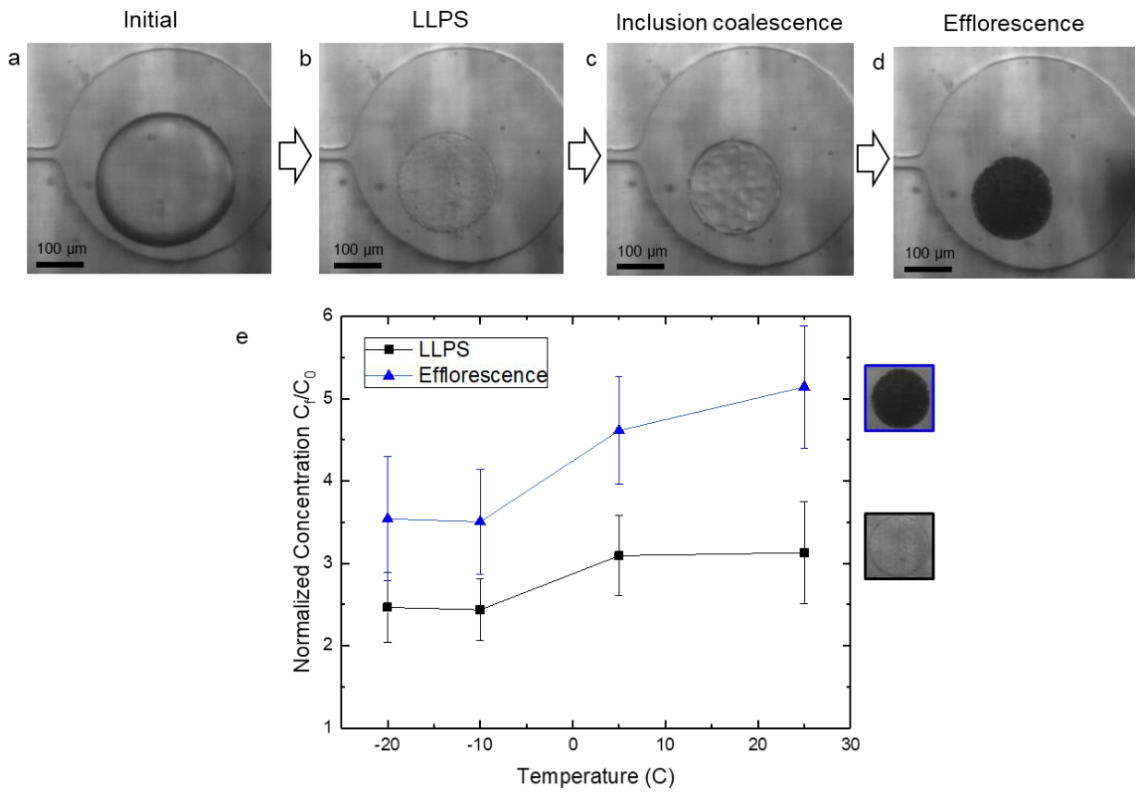


Figure 2-6: (a-d) Typical phase transitions in an aqueous AS and 3-MGA droplet with organic to inorganic ratio (OIR) of 1:10 at 25°C. The droplet undergoes LLPS but there is no subsequent core-shell formation. (e) Phase transitions reported as a function of normalized total droplet solute concentration at different temperatures. C_0 was 1.42 M for this OIR.

Like OIR 1:1, droplets with AS and 3-MGA with OIR 1:10 also undergo LLPS prior to efflorescence at all temperatures during experiments (Figure 2-6a-d). The inorganic rich tiny inclusions undergo some coalescence but do not form a large inner core structure. We also could not discern any differences among the inclusion coalescence behavior at different temperatures. Once again, the LLPS and efflorescence happen at lower concentrations at lower temperatures (Figure 2-6e).

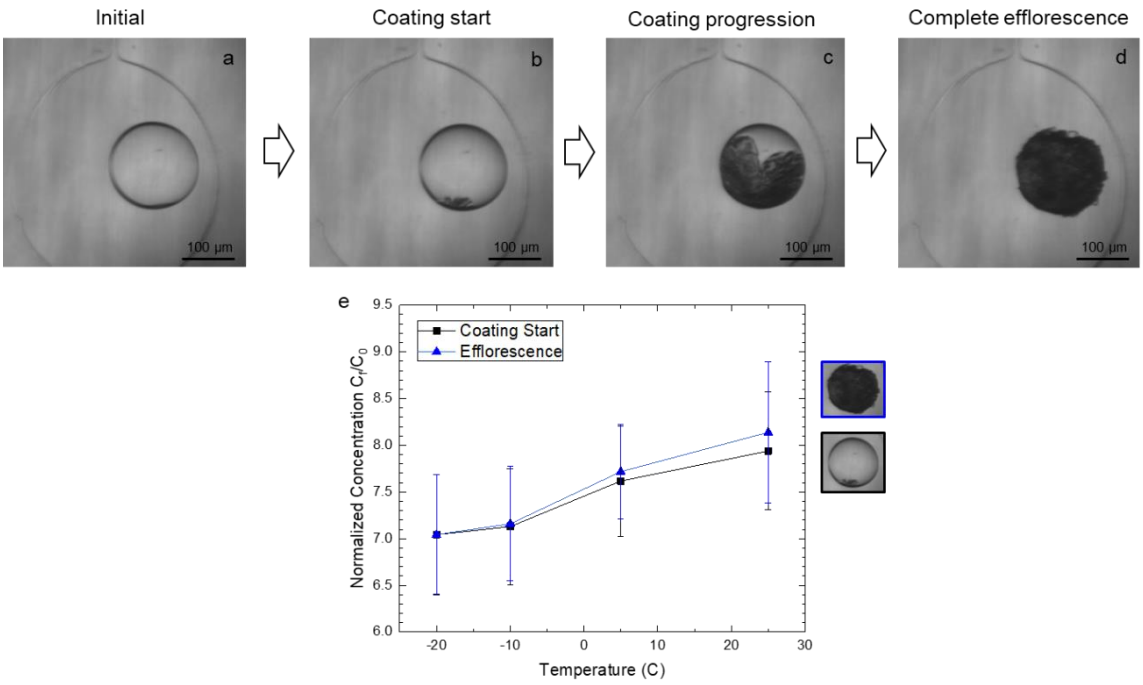


Figure 2-7: (a-d) Typical phase transitions in an aqueous AS and 3-MGA droplet with organic to inorganic ratio (OIR) of 10:1 at 25°C. The droplets do not undergo a LLPS but instead a solid coating forms directly from a well-mixed state. (e) Phase transitions reported as a function of normalized total droplet solute concentration at different temperatures. C_0 was 0.92 M for this OIR.

However, in droplets of AS and 3-MGA at an OIR 10:1, no LLPS is observed (Figure 2-7a-d). The droplets initiate an apparent opaque coating which progresses to efflorescence. We use the term "coating" here to indicate an event where there appears to be a progressing solid sheet growing over or through the droplet, while a portion of the droplet remains liquid. This is similar to observations in [14] for cases where the organic component was present in larger amounts than the salt. Even though these phase transitions are dissimilar to the other two OIRs, Figure 2-7e shows that there is still a temperature dependence of the RH at which the coating starts.

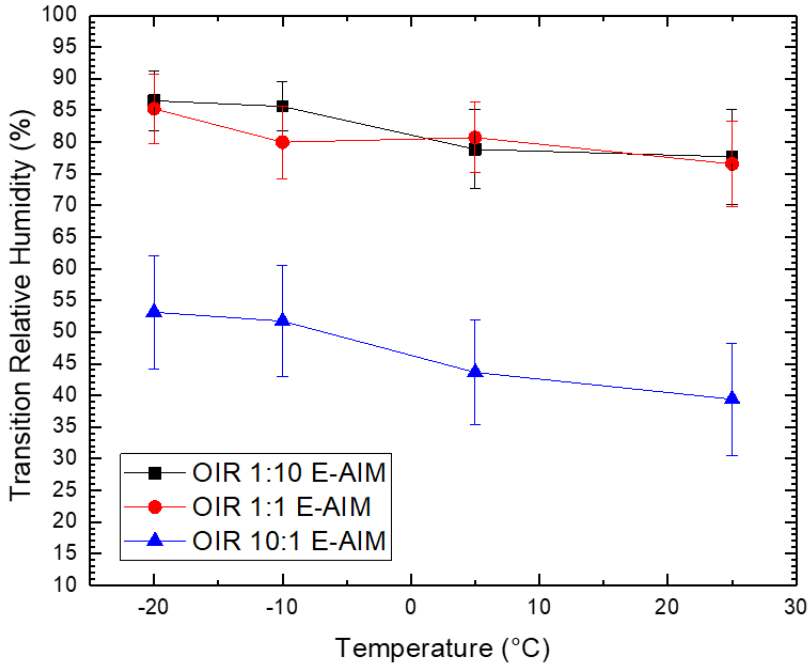


Figure 2-8: Relative humidities of phase transition at different organic inorganic ratios and at different temperatures for AS+3-MGA droplets. For the OIR 1:10 and 1:1 phase transition refers to LLPS and for OIR 10:1 phase transition refers to efflorescence by nucleation of solid coating as there is no LLPS.

The phase transition morphologies observed for different OIRs correspond well to a study by Song et al. [146] who studied ternary mixtures of AS as the inorganic phase and 3 different carboxylic acids with 6 carbon atoms as the organic phase. At OIR 1:1 the authors observed core-shell formation similar to our microfluidic droplet. At an OIR of 11:1, droplets effloresced with needle like growths of ammonium sulfate being initiated from the edge of the droplet, in a similar morphology to our OIR 10:1 droplet.

The SRH for all these cases calculated from E-AIM are plotted as a function of temperature in Figure 2-8. The RH values reported here for OIR 1:1 are close to the SRH reported in similar model aerosol systems in[117,120]. Additionally, these three graphs show that regardless of the OIR, the phase transitions happen at a higher RH at lower temperatures. The systems with OIR 1:1 and 1:10 have a much lower RH of phase separation than the system with OIR 10:1. A comparison between E-AIM and AIOMFAC models for these points show a difference of about 2 to 5% for the cases studied here (Further details are given in Appendix B). This discrepancy is well within the margin of uncertainty of our experimental methods described later on as well as the uncertainties within the models themselves related to the activity of group contribution, temperature dependence of activity coefficients and the temperature dependence of the density of the ternary solutions. However, we note that both methods show a comparable trend in the transition RH values as a function of temperature.

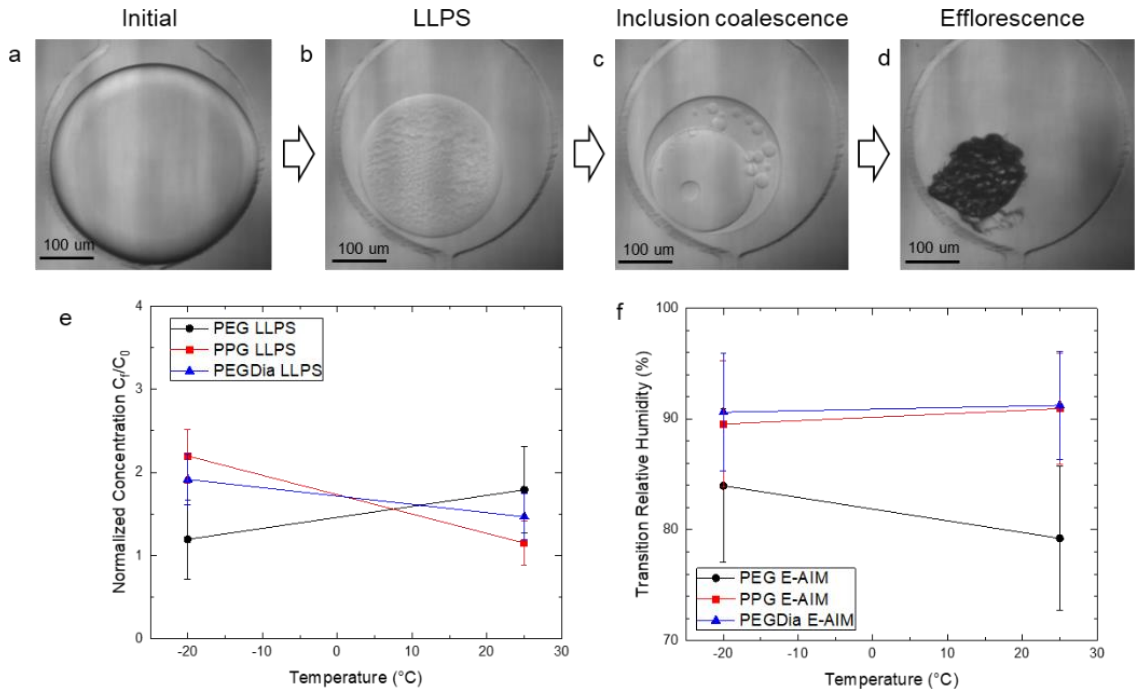


Figure 2-9: (a-d) Typical phase transitions in an aqueous AS and PEG droplet with organic to inorganic ratio (OIR) of 1:1 at 25°C. Droplets undergo LLPS followed by core-shell formation similar to the AS+3MGA systems. Similar behavior is observed in AS+PPG and AS+PEGDia systems as well. (e) LLPS concentrations normalized by initial concentrations for AS and PEG, PPG and PEGDia solutions as a function of temperature. C_0 was 2.56 M, 0.98 M and 0.67 M for AS+PEG, AS+PPG and AS+PEGDia systems respectively. (f) Relative humidity of phase transition to an LLPS state reported for these systems at different temperatures.

Results from experiments performed with AS as the inorganic phase and polyethylene glycol with a molecular weight of 200 g/mol (PEG) as the organic phase with an OIR of 1:1 are also plotted in Figure 2-9(a-d). The SRH of the PEG 200 system in the temperatures studied is in the range of 79% to 83% according to E-AIM, similar to that by You et al. [50] at room temperature for a 2:1 OIR system. The SRH is lower than reported by

Ciobanu et al. [149] in AS and PEG 400, though the differences in higher PEG molecular weight and lower O:C ratio used in their study may explain the higher tendency to phase separate leading to larger SRH. Normalized concentrations of the LLPS transition of this system along with additional systems where the organic phase was swapped with PPG and PEGDia are plotted in Figure 2-9e. The lines in Figure 2-9f show that the SRH for the PPG and PEGDia systems do not increase, and perhaps even decrease with lowered temperature, but the SRH of the PEG system does rise with reduction in temperature.

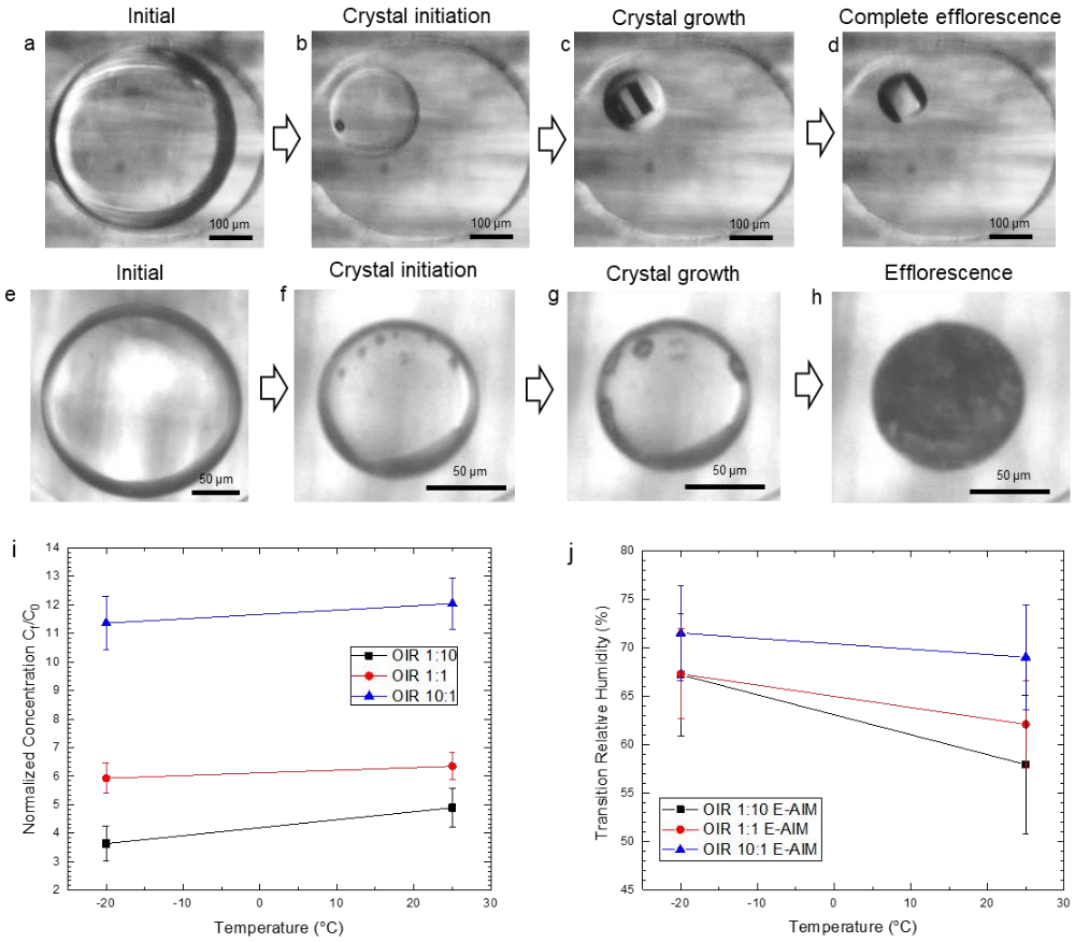


Figure 2-10: (a-d) Typical phase transitions in an aqueous NaCl and 3-MGA droplet with organic to inorganic ratio (OIR) of 1:1 at 25°C. Droplets undergo a single crystal growth at efflorescence.

(e-h) Phase transitions for same compounds at OIR 10:1 at 25°C. Droplets undergo multiple salt crystal nucleation triggered possibly by organic efflorescence followed by complete efflorescence. (i) Normalized concentrations of NaCl and 3-MGA at initial crystal growth. C0 was 1.67 M, 1.15 M and 0.56 M for OIR 1:10, 1:1 and 10:1 respectively. (j) Relative humidity of initial visible crystal formation or efflorescence is reported for these systems at different temperatures.

In the sodium chloride and 3-MGA system, there is no LLPS at any OIRs and at any temperatures studied. Instead, Figure 2-10(a-d) shows that there is crystal nucleation which eventually leads to drying out of the complete droplet for both OIR 1:1 and 1:10. In this case, the first appearance of the crystal (i.e. the efflorescence point) was designated as the phase transition point in our subsequent calculations. The crystal that grew at -20°C at OIR 1:1 and 1:10 visually resembled anhydrous cubic crystals. The crystal eventually led to complete droplet efflorescence with some organic efflorescing on the crystal faces as the water evaporated. There was no LLPS in the 10:1 OIR system either, however, the crystal growth morphology was quite different with multiple crystals nucleating inside the droplet followed by an amorphous efflorescent structure formation as shown in Figure 2-10(e-i). Figure 2-10i shows that, like the AS and 3-MGA systems, the concentration at which the initial crystal appeared is affected by the temperature of the experiment quite monotonically across all OIRs. Finally, Figure 2-10j shows that the RH for this phase transition also increases with lowered temperature. Unlike AS and 3-MGA, these OIR 1:1 and 1:10 systems have much lower RH of phase transition than the OIR 10:1 system.

A sensitivity analysis for measurement uncertainty in different parameters and their influence on the model calculated RH values were also carried out. Typical uncertainty in

droplet diameter measurement was about ±5μm due to the available camera resolution and contrast. Uncertainty was about ±1% in the initial solution concentration. Uncertainty in microfluidic channel height was about ±5μm due to variations during fabrication between different wells in a single microfluidic device. As a result, the uncertainty in droplet final concentration was ±10.4%. This led to an average of 5.2%, 4.1% and 9% uncertainties in RH prediction for AS + 3MGA systems at OIR 1:1, 1:10 and 10:1 respectively using the thermodynamic models. Similar analysis led to about 5.7%, 4.5% and 5.2% uncertainties in RH prediction for the NaCl + 3MGA system at OIR 1:1, 1:10 and 10:1 respectively. Another source of uncertainty in the thermodynamic models can be the density of the solutions where an error of about 0.01 g/cm^3 leads to an error in concentration prediction of 6% on average and final RH prediction error of about 3%, 2.5% and 5.2% for the AS + 3MGA systems at OIR 1:1, 1:10 and 10:1 respectively.

Droplet dehydration time series curves were generated by tracking the droplet volume as a function of time from the time-lapse images taken during the experiment using the method described in the main paper. Dehydration curves for the AS + 3-MGA droplets in microfluidic wells are plotted in Figure 2-11. Dehydration rates were almost equal among all organic to inorganic ratio samples at a given temperature as shown in Figure 2-11a. Variations among different OIRs could be caused due to several reasons including droplet hygroscopicity and initial size but the likely biggest cause is the variability among the different device thicknesses used as the PDMS layer on top of the droplet is the biggest factor restricting water evaporation. However, the dehydration rate is a strong function of the temperature of the experiment as shown in Figure 2-11b. e.g. the experiment at 25°C

took only about 11 hours from initial loading to efflorescence compared to almost 5 days for the same system at -20°C.

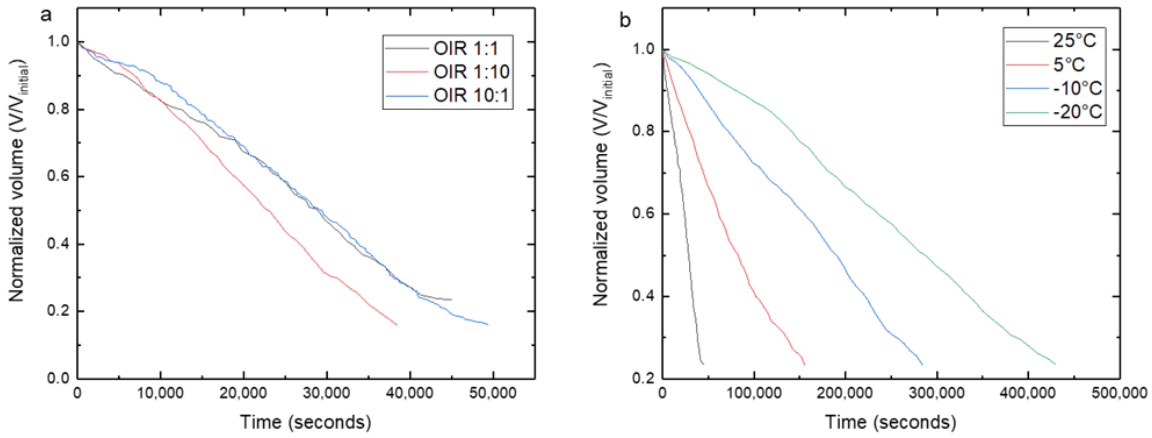


Figure 2-11: AS + 3-MGA droplet dehydration time curves. (a) droplet normalized volume vs. time plotted for three different OIRs at 25°C. (b) droplet normalized volume vs. time plotted for OIR 1:1 at different temperatures.

Experiments were also performed to investigate the effect of temperature and dehydration rate on the efflorescence of binary sodium chloride water solution. Faster dehydration rate experiments (~2.5% RH/min at -21°C and ~1% RH/min at -25°C) were performed on the environmental cell. Figure 2-12b,c show at -21°C, 15.3 ± 0.1% of the droplets formed $NaCl{\cdot}2H_2O_{(s)}$ which went up to 98.5 ± 0.6% at -25°C. Raman spectra (Figure 2-12a) were used to confirm the hydration state of the particles. Slow dehydration rate experiments were performed over a period of about 5-6 days in the microfluidic trap and Figure 2-12e,f show that both at -25° and -35°C, the efflorescent form was NaCl(s).

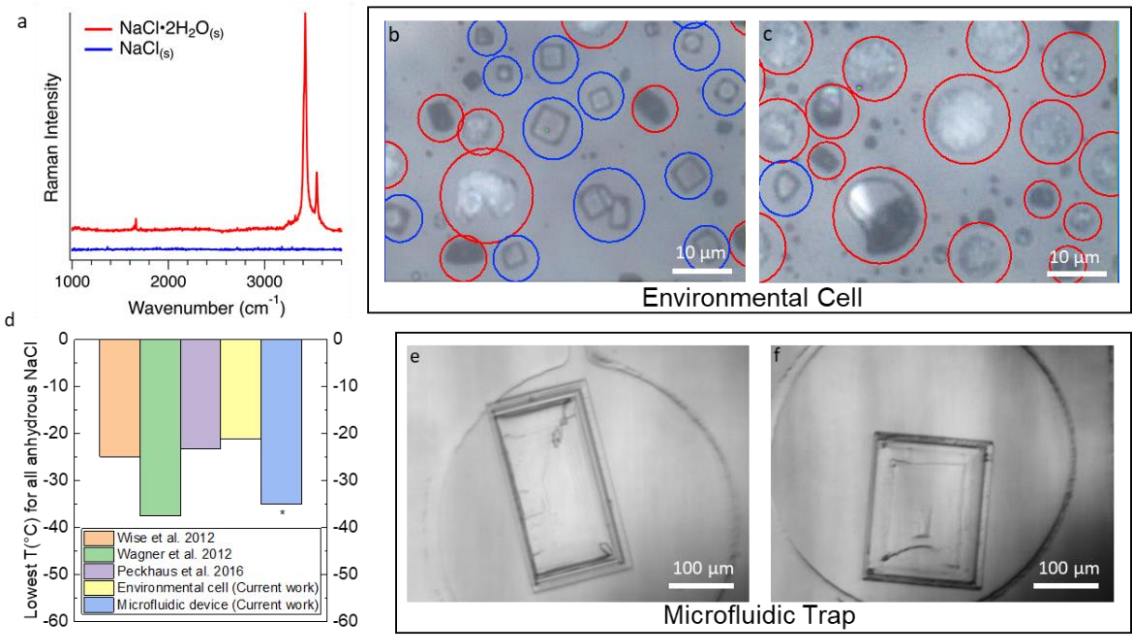


Figure 2-12: (a) Raman spectra of NaCl·2H2O(s) (red) and NaCl(s) (blue) and optical images of the these particles deposited on the environmental cell substrate and fast dehydrated at (b) -21°C and (c) -25°C with colored circles indicating the different crystals. NaCl(s) efflorescence observed at (e) -25°C and (f) -35°C in a microfluidic well at very slow dehydration rate. (d) Comparison of the lowest temperature at which all of crystals were anhydrous with literature. The asterisk over the microfluidic device column indicates the lowest temperature experiment performed. Dihydrate crystals were not observed at this temperature. Lower temperatures may produce dihydrate crystals in this experiment but were not performed.

2.4 Discussion

Previous studies on model aerosols have shown that the SRH is affected only weakly by temperature and molecular weight of the organic compound [120], and instead dominated by the O:C ratio of the organic compound. While no direct comparisons are

possible between the studies due to the vastly different molar ratios of solutes and dehydration rates as well as specific temperatures studied, the trend in the results can be commented upon. e.g. efflorescence RH of AS with different organics shows a slightly increasing or constant trend at lower temperatures [150,151]. Similarly the deliquescence RH of mixed organic and AS systems show an increasing trend with reduced temperatures [152]. Here, in a similar fashion, we see a clear trend towards higher SRH at lower temperatures for the AS + 3-MGA systems studied. This could be explained by reduced solubility of pure organics as well as salts at lower temperatures [153]. However, this trend is not always present across all systems studied and it cannot be positively asserted that reducing the temperature of LLPS would lead to a higher SRH across the board which is in agreement with [120]. More likely, the type of organic including the functional group (alcohol, carboxylic acid etc.), molecular weight as well as the solubility of the organic in the salty droplet leads to a nuanced behavior with no generalizable trend in terms of temperature dependence.

While we did not see any dependence of actual series of morphological phase transitions on temperature, we did see different morphologies depending on the droplet OIR. Droplets with more salt than organic resulted in small inclusions containing the salt rich phase formed inside the droplet which did not fully separate from the outer organic phase. We noticed this same trend where metastable inclusions in the OIR 1:10 droplets formed after LLPS which coalesced to form larger inclusions but not a central core before efflorescence. Fard et al. [154] previously observed temperature and drying rate dependent inorganic ion diffusivity and organic inclusion coalescence in mixed AS and carminic acid

droplets at OIR 1:3 after LLPS. We could not discern any temperature dependence of the inclusion coalescence in our drops. This could be due to several reasons. The fastest rate of dehydration in our experiments were an order of magnitude slower than the slowest rate reported by Fard et al. Also, the OIR and the organic acid itself is very different which perhaps suggests that a coalescence temperature dependence is possible at a different temperature rate and OIR in our system. We also cannot rule out the possibility that we missed out on small variations in coalescence behavior due to limited resolution and contrast in the droplets. In systems with comparable amounts of salt and acid, fully separated core-shell formation is observed. For droplets which contained significantly larger amounts of acid than salt, in was observed previously [14,146] that the droplet did not spontaneously change phase all at once, but rather a progressive solidification or coating formed through crystallization from a given point on the droplet surface. A similar behavior was also observed here for OIR 10:1, where the crystallization started from a given point on the droplet and no LLPS was observed. Finally, in addition to the phase transition events, a noticeable difference in the critical value of the SRH was observed here for systems with different OIR. In AS + 3-MGA systems containing more organic than salt, specifically for OIR 10:1, the ERH was much lower than the SRH reported for the systems with OIR 1:1 and 1:10. This is probably caused by the efflorescence of the AS phase as has been previously suggested by Song et al. [146]. After the initiation of the coating at a low RH, the whole droplet effloresces quickly which we believe is the organic crystallizing heterogeneously on the already effloresced AS since the amount of organic is 10 times larger in the droplet than the salt.

Experiments performed on the sodium chloride and 3-MGA systems showed a completely different phase transition morphology in contrast with AS, as expected. For the systems studied here, the NaCl droplets did not undergo any LLPS, but started an anhydrous crystal growth before efflorescence resulting in an organic coating on and around the salt crystal. It has been shown previously [155] that AS has a larger SRH than sodium chloride in a large study of mixed salt and organic droplets. This indicates that AS has a larger tendency of salting out the organics from the droplet than sodium chloride thus favoring LLPS over crystallization, potentially explained by the Hofmeister series[156] where the abilities of ions to salt out proteins are classified in a linear order. Since there was no LLPS in our study, we were able to report the concentration and RH at which the sodium chloride crystal first appeared or the start of efflorescence as a function of temperature. At OIR 10:1, multiple NaCl crystals started efflorescing at a larger RH close to the DRH of pure NaCl (~75%) than the other OIRs indicating possible heterogeneous nucleation of NaCl due to formation of a few organic crystals. Studies have shown that presence of a nucleating surface is able to crystallize inorganic salts close to their DRH [157,158]. Following this, droplets rapidly formed an opaque amorphous coating without significant crystal growth or droplet dehydration, indicating an organic coating due to the comparatively large amount of organic in the droplet. This contrasts with the RH of the crystallization for AS system where the droplets crystallized closer to the ERH of pure AS, indicating that the AS crystallized first. More work is needed to understand this differing behavior between the two systems. At all OIRs, the appearance of the crystal was at a lower concentration (and higher RH) at lower temperatures indicating again perhaps the reduced solubility of the components as the temperature reduced. It is interesting to note that we observed the

anhydrous form of the NaCl crystal during the experiment rather than the thermodynamically stable dihydrous form which resembles a circular shape with irregularities [101].

We noticed a similar behavior in binary NaCl water systems. In this case, NaCl droplets without a ternary organic phase and dehydrated at two very different rates showed a dependence of the rate of change of RH as well as the droplet surroundings on the crystal structure formed. The slower dehydrating microfluidic droplets formed cubic $NaCl_{(s)}$ crystals down to -35°C whereas the faster dehydrating environmental cell droplets formed irregular $NaCl{\cdot}2H_2O_{(s)}$ crystals even at -25°C. This is perhaps a combined effect of the vastly different dehydration rates as well as the droplet surroundings in the two setups. It has been reported previously [159] that at higher rates of dehydration NaCl droplets form a dendritic and irregular crystal structure upon nucleation whereas slower dehydration leads to the single cubic crystal structure. The authors postulated that slower dehydration leads to related slower homogenous nucleation and growth which favors the singular anhydrous crystal rather than rapid crystal growth inducing trapped water molecules and a random structure. It has also been theorized previously that the presence of heterogeneous nucleation sites on a solid-droplet surface and higher dehydration rates are capable of inducing the transition to $NaCl{\cdot}2H_2O_{(s)}$ at higher temperatures by reducing the energy barrier for the transition to this state [101,160]. No consensus exists in literature about the temperature range of this transition in droplets with the range being dependent on size, dehydration rate and overall method, for example in the study by Wagner et al. [160], the highest temperature for the transition was about -37.45°C whereas Peckhaus et al. [101] and

Wise et al. reported -23°C and -21°C respectively for the same transition. This highlights that fact that while speculative, differences in droplet size, shape, substrates as well as the other factors discussed above could all potentially be contributing factors while comparing the transition temperatures between the methods presented here. The rate of dehydration has also been shown to be a major factor affecting the morphology of efflorescing particles in previous studies on seawater [131,132] as well as studies in pharmaceutical compounds [161,162] and could be the biggest cause of the different crystal structures at the same temperature seen here and needs to be investigated further.

2.5 Conclusions

In this chapter, we used a PDMS based microfluidic device to trap aqueous chemical systems relevant to atmospheric aerosol droplets. The device was placed inside a hermetically sealed temperature-controlled cell, and the temperature of the cell was held fixed throughout the experiment. The droplets dehydrated as water pervaporated through the PDMS and left the microfluidic device in a quasi-static manner. Droplet solute concentrations were calculated based on optical microscopy data and the concentration was used with the E-AIM model to calculate relative humidities at the first phase transition. Concentrations and relative humidities of phase changes of the droplet were reported as a function of temperatures ranging from +25°C to -20°C. Additionally, a study on effect of dehydration rates on efflorescent crystal structure was performed on binary sodium chloride and water systems with two vastly different dehydration rates.

It was observed that the changes in phase occurred at a lower total solute concentration at lower temperatures in most droplets. The relative humidity of the first phase transition in all systems were calculated using the E-AIM model, showing that the SRH increased as the phase transition temperature was lowered. The SRH results are somewhat contradictory to previous studies with similar systems which show weak or no appreciable trend in SRH at lower temperatures. While the SRH for supermicron droplets does not depend on its size, slow time scales for water transport may play a role in phase transitions in a microfluidic well when compared to other methods. However, the trend of lower total solute concentration and higher RH at phase separation with lowered temperature is clearly visible in the AS and 3-MGA system. The role of the organic to inorganic ratio on SRH was also showed to be temperature dependent. Rate of dehydration and the droplet surroundings also had an impact on the phase transition with NaCl crystals formed at slower dehydration rates in our microfluidic device favoring the metastable anhydrous form than the dihydrous form which was predominant at faster dehydration rates at the same temperature on a cold plate device.

A shortcoming in the methodology includes the need to use a thermodynamic model to calculate the equilibrium RH of the droplets instead of directly measuring it. This adds to the experimental uncertainties (droplet concentration, volume and density) by introducing modeling related uncertainties for activities of the mixed organic inorganic systems at low temperatures. Future device designs with either a controlled RH environment surrounding the PDMS device, or flow channels that allow for droplet

rehydration to identify deliquescence behavior for further benchmarking to known transitions would be helpful to overcome this shortcoming.

This study provides a novel look at the temperature, OIR and dehydration rate dependence of phase changes in atmospheric aerosol relevant chemical systems using a low-cost microfluidic apparatus. Ultimately, the temperature and RH-dependent phase of aerosol droplets influences various cloud and atmospheric processes like gas-particle partitioning of certain compounds, cloud condensation nuclei activation, optical properties, and heterogeneous chemistry.

Chapter 3

Development of an automated high-throughput droplet freezing counter microfluidic device

3.1 Introduction

Clouds are key contributors in modulating Earth's climate through their albedo and atmospheric lifetimes [51,163,164]. Ice nucleating particles (INPs) in cloud drops facilitate the formation of ice crystals at temperatures warmer than the homogeneous ice nucleation temperature of pure water of -38°C [43]. INPs control the size and concentration of liquid vs. frozen cloud drops [12,51], where clouds comprised of more ice crystals have larger particle sizes and have lower albedo than those composed of mainly liquid drops [50,165,166], thus shortening cloud lifetime. Ice nucleation also controls precipitation and the water content

of clouds by scavenging water vapor from liquid droplets [51,66]. INPs are emitted from various sources, including mineral dusts from arid regions [52,167], from plants and soils, anthropogenic activity [48,168] and, marine sources [12,59].

IN temperatures of droplets are measured using online or offline instruments. Continuous flow diffusion chambers [169,170], and continuous flow tubes [171] are typical online equipment which continuously measure ice nucleating particles (INPs) at a fixed temperature by connecting to air sampling lines from the ambient environment. However, they are complex and expensive. Among the offline equipment, large cloud expansion chambers can be used to simulate whole clouds in the upper troposphere [86]. However, there are other less complex and expensive classes of benchtop offline experiments which allow IN studies in a laboratory. For instance, differential scanning calorimeters [124], environmental Raman stages [148,172], electrodynamic balances [173], optical traps [174], static cold plate based freezing assays using well plates [175,176], printed droplet arrays on hydrophobic substrates [177], droplets in microwells [178] and droplet arrays on a pyroelectric polymer substrate [179] have been used to study ice nucleation from atmospheric and laboratory samples.

This chapter documents the development of a microfluidic platform with on-chip droplet generation and continuous high-throughput ice nucleation detection. The platform and methods described here could be adapted to count INPs in atmospheric aerosol samples collected in liquid. There the high throughput nature of the device would allow measuring minuscule amounts of INP in field samples more accurately at warmer temperatures. The stage could also be modified into a new class of liquid particle counter, analogous to

condensation particle counters for aerosols, where a small number of particles in ultrapure liquid phase samples can be detected by measuring ice nucleation temperatures of a large number of droplets rapidly. One should note, however, that while there are advantages to high-throughput methods, there are also limitations, including difficulty in accessing low cooling rates (~1°C/min) as well as observing phase change events in individual droplets with highly precise temperature measurements.

Two automated detection approaches are presented here: a polarized light method and a deep neural network (DNN) based machine learning method. The appropriate detection algorithm is implemented depending on the throughput rate needed, and optical contrast available. A case study on a model ice nucleating particle dissolved in water droplets with the platform developed here is presented in the next chapter. To our knowledge, this is the first high-throughput microfluidic device of its kind where automated machine learning and polarized light-based detection algorithms have been used to detect ice nucleation.

3.2 Materials and methods

3.2.1 Temperature controlled platform

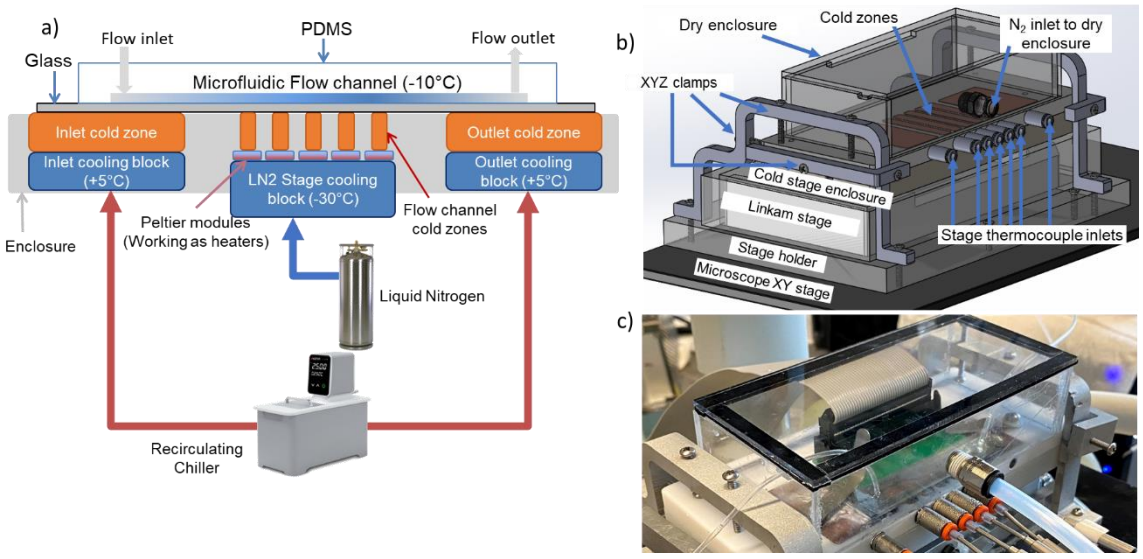


Figure 3-1: Temperature-controlled microfluidic platform. (a) Schematic showing the complete platform with multiple copper blocks for individually controllable cold zones and the cooling blocks underneath which act as heat sinks for the cold zones. (b) CAD model showing the platform on a microscope stage with a dry, optically transparent enclosure on top surrounding the region where the microfluidic device is placed. (c) A photograph of the stage in action with all fluid and electrical connections.

A temperature-controlled platform based on Stan et al. [80,180] was constructed with multiple independent temperature zones, to control the droplet temperatures inside the microfluidic channel. The design allows a temperature gradient along the flow channel. Figure 3-1a shows our platform design with seven upper copper blocks (called cold zones) with the microfluidic device placed on top. The cold zones are in thermal contact with

separate heat exchangers (called cooling blocks) with cooled liquid flow going through them to maintain a fixed temperature.

Two separate sets of cooling blocks are used. The inlet and outlet cold zones are in contact with two cooling blocks maintained at above freezing temperatures by a recirculating chiller (Anova R10, Anova Inc.), pumping ethanol at 5°C through the cooling blocks. The flow channel cold zones are in contact with a programmable cooling block (LTS 420, Linkam Scientific) maintained at cold temperatures (-30°C) by pumping liquid nitrogen through the block. Thermal grease (Ceramique 2 Thermal Compound, Arctic Silver) was applied between all mating surfaces to ensure adequate heat transfer at low temperatures.

Sandwiched between the flow channel cold zones and the liquid nitrogen cold block are Peltier elements (926-1209-ND, Laird Technologies). These elements act as temperature controllers and maintain the desired temperature in the cold zones on top, independent of the liquid nitrogen cold block temperature underneath. A custom PID loop-based control circuit was built using both discrete electronic components and off-the-shelf printed circuit boards (PCBs). T-type thermocouples (TJC36-CASS-062U-2, Omega), which were inserted into holes drilled into each cold zone. Thermocouple readings were taken with thermostats (TEC-9100, Tempco). Target values for each cold block were selected and the thermostat provided a PID control signal to the Peltier elements to maintain the target temperature. This thermostat output was insufficient to power the Peltier elements and hence this signal was amplified using the circuit shown in Figure 3-2a.

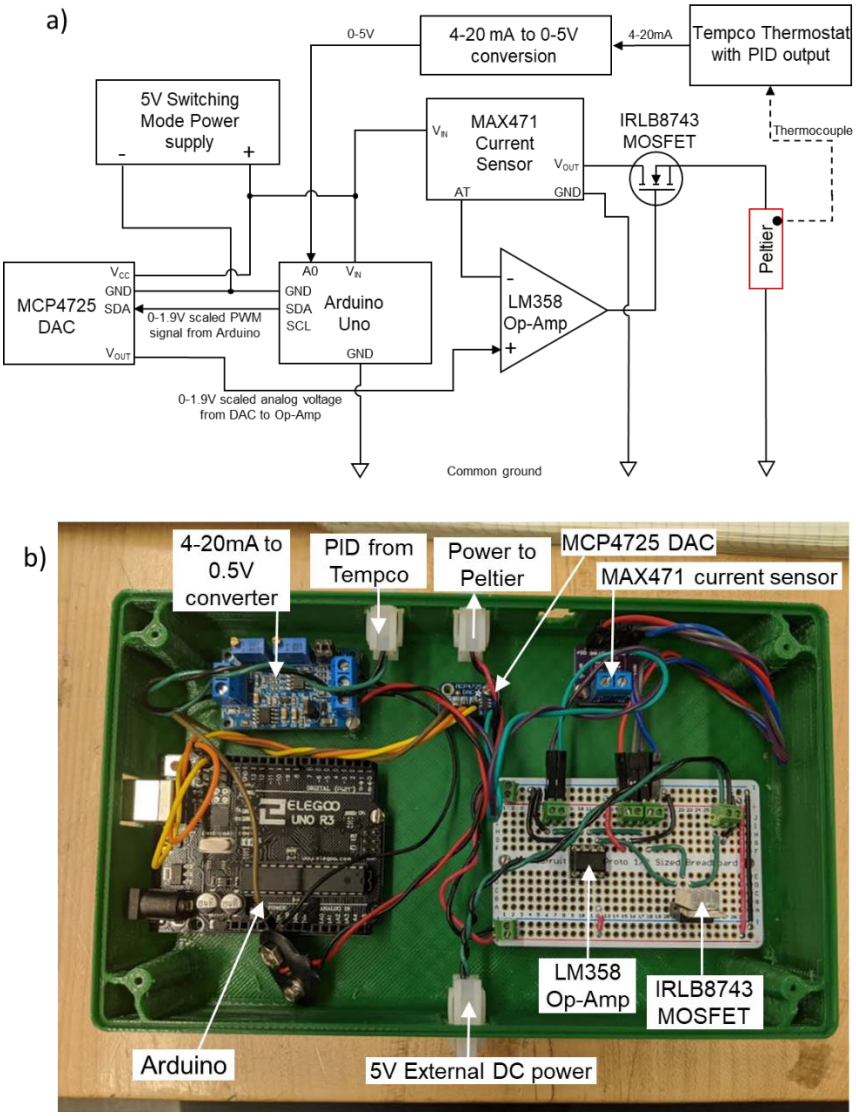


Figure 3-2: (a) Custom designed control circuit for the temperature-controlled platform. (b) A labeled photograph of the components installed inside a 3-D printed enclosure.

Briefly, the 4-20mA current signal was first converted to a 0-5V voltage signal. This voltage signal could not be directly con-verted to a 0-5A current and fed to the Peltier elements since they operate within a 0-1.9A current limit. An Arduino Uno (Ardunio.cc) was used to scale the 0-5V signal to a 0-1.9V pulse width modulated (PWM) signal which

was converted to a smooth DC signal using a digital-to-analog converter (DAC) (MCP4725, Adafruit). A 0-30V Lab Benchtop DC power supply (Amazon) was then used to drive the Peltier elements by modulating its output to a 4.5V, 0-1.9A current source using a current sensor (MAX471, DigiKey), Op-Amp (LM358, DigiKey) and an n-channel metal oxide semiconductor field effect transistor (MOSFET) (IRLB8743, Digikey). These components were placed inside a custom 3-D printed enclosure (CSE Anderson Labs, University of Minnesota), and standard cable connectors were installed in the enclosure to ensure easier connection between electrical wiring.

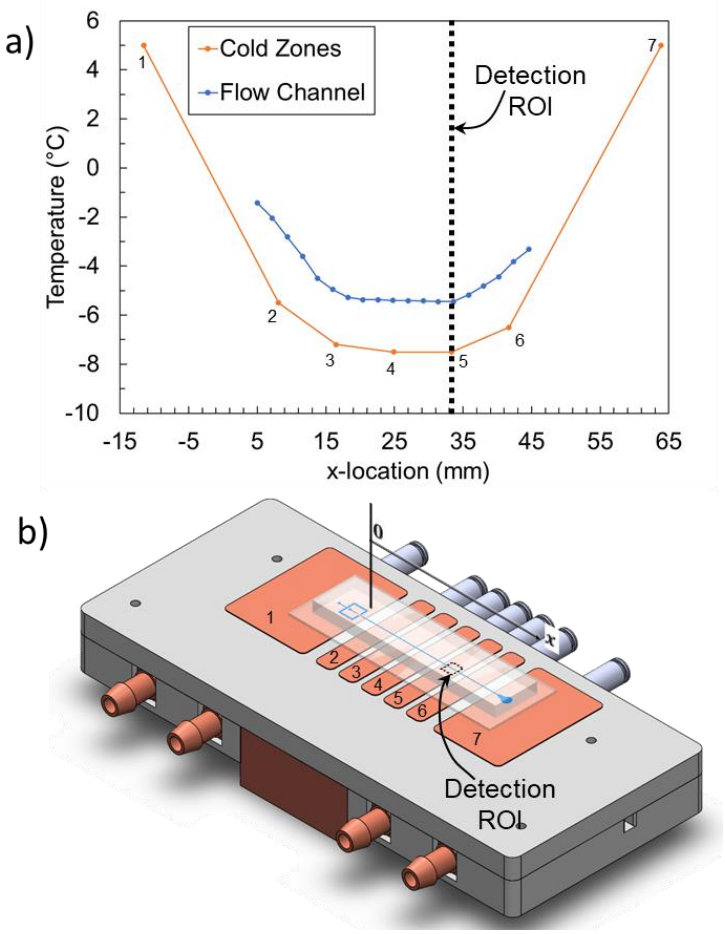


Figure 3-3: (a) Cold zone temperatures measured using thermocouples and microfluidic channel temperatures measured using on-chip thin film sensors. The number labels next to the orange line indicate the corresponding cold zone in the schematic, and the temperature values are measured using thermocouples inserted into the copper blocks. The blue line indicates the temperature in the flow channel. (b) The location of the detection region of interest (ROI) during droplet freezing experiments and the numbered cold zones.

Figure 3-3b shows the temperatures in all the cold zones during a typical experiment. The temperature of each cold zone is adjusted until a desired temperature distribution is achieved inside the microfluidic channel. Droplet freezing detection was performed using this stage on a reflective microscope (SZX10, Olympus) and 128x256 pixel videos were recorded with a high-speed camera (AX100 mini, Photron) at 3,000 frames per second. A region of interest (ROI) of 250x500 μm was recorded for a continuous 4 minutes of experiment time, and a constant temperature was maintained in the flow channel.

3.2.2 Platinum thin film temperature sensor fabrication and calibration

Embedded thin film temperature sensors similar to those in Stan et al. [80] were used. Figure 3-4 shows the design and fabrication process of the platinum resistive temperature device (PRTD) linear array. The array had 19 PRTDs in a row and a pitch of 2.2 mm between each PRTD. These arrays were fabricated onto 4” diameter and 0.5 mm thick soda-lime glass wafers (University Wafer) using a modified version of the lift-off lithography process described in [80,181].

The steps described here are pictorially represented in Figure 3-4a and corresponding design or photographs are shown in Figure 3-4b. The wafer was piranha cleaned in a 5:1 H_2SO_4:H_2O_2 bath at 120°C for 15 minutes. It was then rinsed with DI water with 3 cycles in a dump rinser and dried in a spin-rinse-drier with a stream of N_2 gas. To complete the drying process, the wafer was heated on a hot plate at 120°C for 10 minutes. Following cleaning and drying, the wafer needed to be coated with a photoresist for photolithography. The process started with covering the surface with an adhesion promoter hexamethyldisilazane (HMDS). The wafer was placed in a wafer holder and immersed in HMDS vapor for 3 minutes to deposit a thin layer on the surface. Then, the wafer was spin coated with a positive photoresist, AZ 1518 (Microchemicals GmBH), for 30 seconds at 3,300 rpm at a ramp rate of 3,000 rpm/s to achieve a coating thickness of 2μm. Then a soft-bake using a hotplate at 100°C for 50 seconds was performed to stabilize the photoresist by evaporating the solvents, resist mask adhesion and improve resist adhesion to the substrate. Then a chrome photomask of the PRTDs fabricated on order by the Minnesota Nano Center was used to expose the coated wafer to UV light on a mask aligner. The exposure was performed with hard contact mode with a 9 second exposure on a MA6-P mask aligner (Karl Süss). Development of the photoresist was done using Microposit 351 developer for 40 seconds. The wafer was rinsed with DI water and blow dried with N_2. A hard-bake at 115°C for 50 seconds was performed to completely cure the remaining photoresist and promote adhesion to the substrate. The step-by-step fabrication protocol is provided in Appendix C.

The next step following the preparation of the photoresist coated wafer was metal thin film deposition. This was done on an ATC 2000 sputterer (AJA International). Titanium and platinum targets were installed DC powered guns. The wafer was loaded into the sputter chamber and the system was pumped down to the range of 10-5 bar. A 2nm adhesion promoter layer of titanium was deposited by running the sputterer for 40 seconds and a 200nm layer of platinum was deposited by running the sputterer for 480 seconds. To perform lift-off of the metal layer, the completely metal coated wafer was immersed upside down by placing it in a cassette holder in an acetone bath on a hotplate at 60°C overnight. The bath was sealed with an aluminum foil to make sure the acetone did not evaporate. Finally, the wafer was annealed for 24 hours at 500°C in a furnace to stabilize the metal layer resistivity and extend its lifespan.

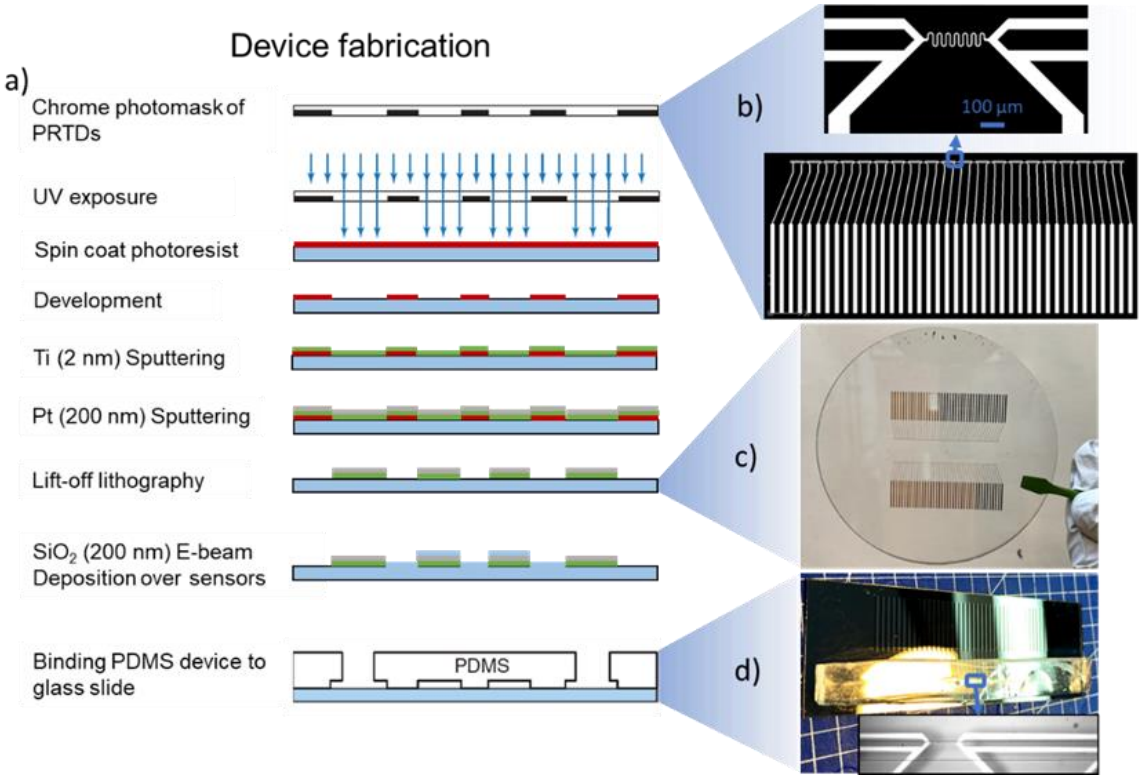


Figure 3-4: PRTD array fabrication and integration with microfluidic PDMS channels. (a) PRTD array fabrication and bonding process schematic. (b) Array mask design, showing 19 PRTD arrays in a row. (c) After deposition onto a 0.5mm glass wafer. (d) Completed device after bonding to a PDMS channel and coating the rear face with aluminum for using the device on a reflective microscope stage.

Following metal deposition, each circular wafer was cut into two 50.8x27 mm rectangular pieces using a DISCO wafer saw, each containing a single PRTD array. To measure the resistance of the PRTD array, a custom designed PCB with 40 contact pads (Sunstone Circuits) was attached to the leads on the substrate using electrically conductive adhesive transfer tape (9703, 3M) which only conducts electricity across its thickness through embedded discrete conductive particles. A programmable digital multimeter (Keithley 2701 with a 7710 multiplexer card, Tektronix) was used to read individual PRTD voltages which were converted to resistance using a modified four-wire protocol. A 5V DC power source was connected in series to the PRTD array using the leftmost and rightmost leads. A 100 ohm high precision reference resistor (RWPB03U100R0TS, Johanson Dielectrics) labeled R_{ref} or was connected in series and the voltage drop across this was also wired to the digital multimeter. The voltage drop across this R_{ref} provided an accurate measure of the current flowing through the circuit which was assumed to be equal in all 19 PRTDs in series. The schematic of the measurement setup and the physical arrangement of the electrical components are shown in Figure 3-5. The corresponding LabVIEW code used to collect resistance data can be found in Appendix C.

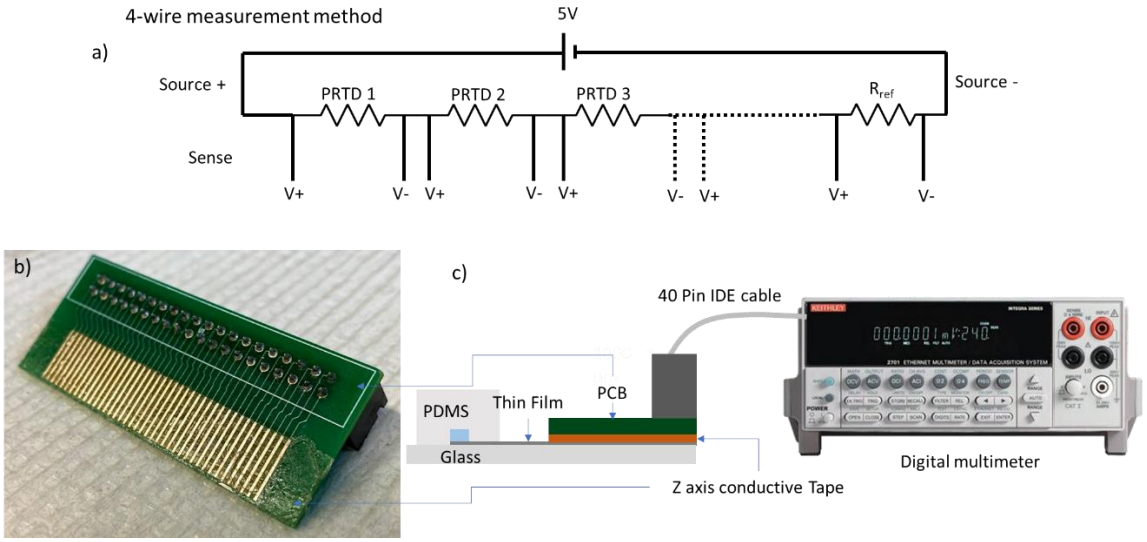


Figure 3-5: The resistance measurement setup for the thin film PRTDs. (a) The electrical connection diagram of the sensors for 4-wire sensing of the resistance. (b) The PCB with the z-axis electrically conductive tape applied. (c) The complete circuit with the digital multimeter to read the voltages and the interconnections.

For the calibration process, a factory calibrated standard RTD (5606-50-B, Fluke) was used. For this, metal box with a dimension of 64x46x15 mm was created from 1/16" sheet aluminum. The glass substrate with the PRTDs and the electrical connectors was placed inside, and the standard RTD was also placed inside the box. The box was filled with Novec 7500 (3M) which is an engineered heat transfer fluid but in not electrically conductive. The box was then placed on the cold plate of the LTS 420 stage, as shown in Figure 3-6a. The whole stage was placed inside a large foam box and openings were sealed with duct tape to keep water condensation on airflow from interfering with the cold temperature calibration. A stepped temperature ramp with a 2 hour hold at each 10°C interval from 30°C down to -50°C was performed. The voltages measured by the digital

multimeter was recorded and calibration curves unique to each PRTD in the array were created based on a linear fit. The array numbering, resistance data recorded during the calibration and calibration curves are shown in Figure 3-6b and Figure 3-7a,b. The accuracy of the sensor array from the calibration curve based on the linear fit was ±0.03°C.

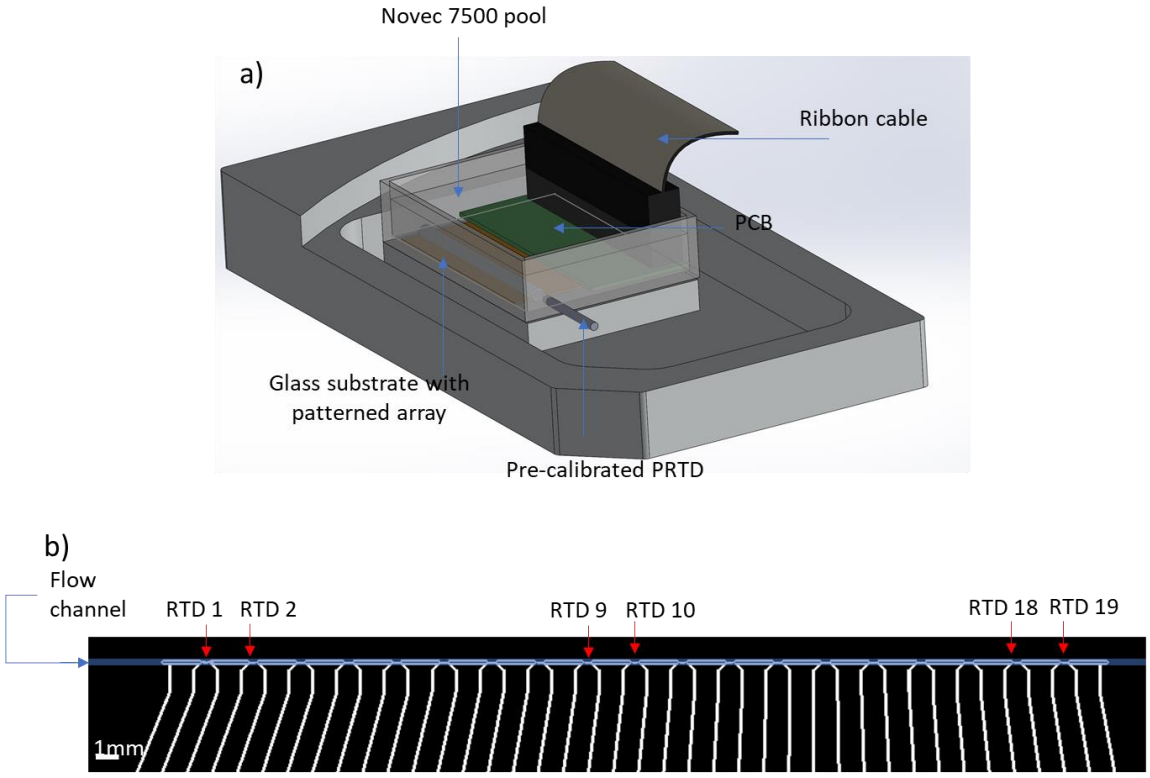


Figure 3-6: PRTD array calibration setup. (a) The calibration box placed on the cold plate of the Linkam stage. (b) The numerical designation of the arrays.

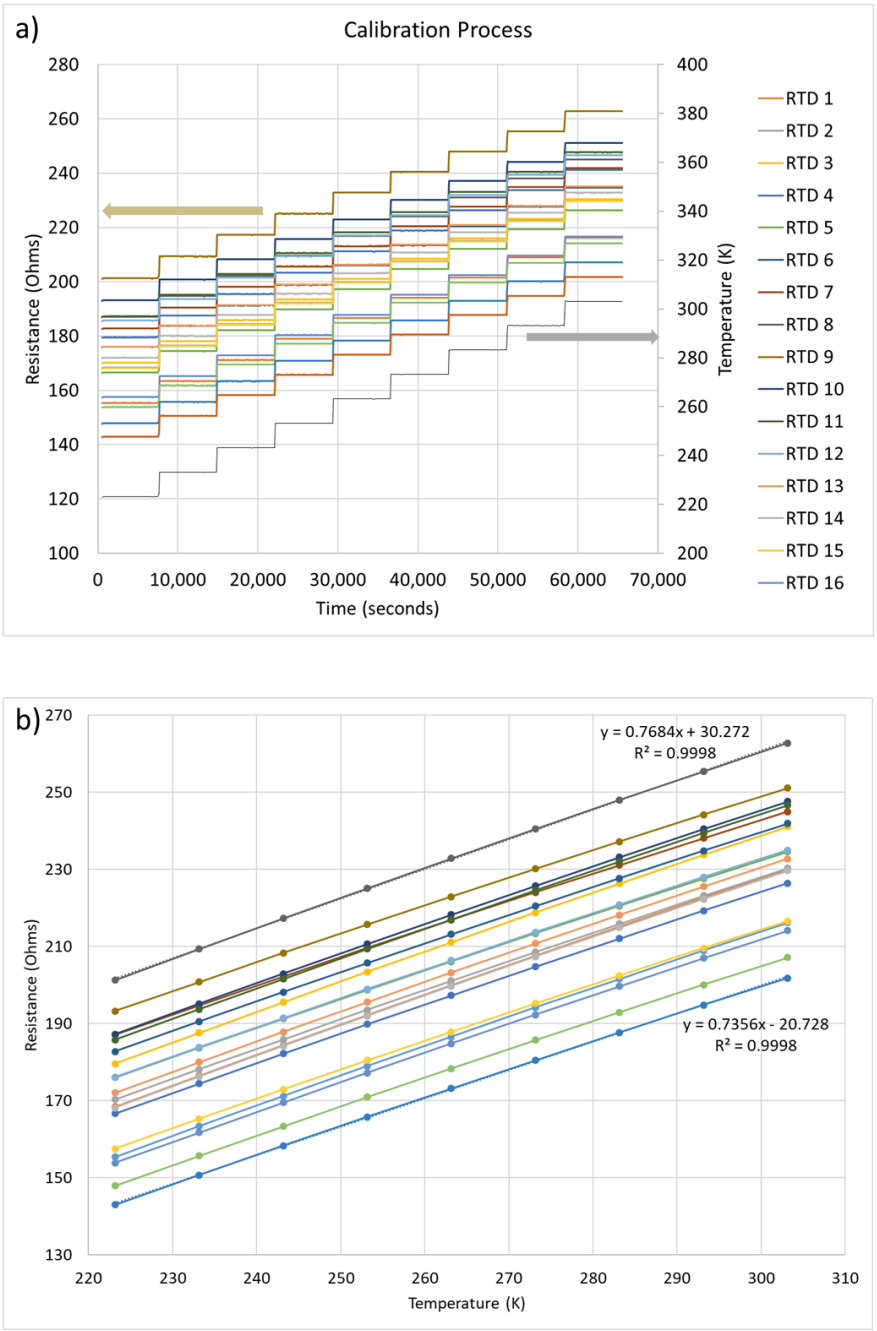


Figure 3-7: The PRTD array calibration results. (a) The calibration temperatures vs. time. The y-axis on the left corresponds to the resistance curves of each PRTD array and the y-axis on the right corresponds to the temperature of the cold stage indicated by the gray curve. (b) he resistances of

the PRTDs plotted as a function of temperature. Linear fit equations for two of the arrays are also shown. These curves are unique to each device and must be measured individually.

3.2.3 Microfluidic device design and fabrication

The microfluidic device used for this project is a simple straight channel with a flow-focusing droplet generator [80]. The main channel dimensions are 200x150 µm and has a length of 50mm. The width of the droplet phase channel upstream of the flow-focusing junction is 40 µm. The device design is shown in Figure 3-8a. A negative vinyl mask was printed (CAD/art services), and the mold for the PDMS devices was prepared using a soft photolithography process, slightly modified from the one described in Chapter 2, Solid state SUEX sheets (DJI Microlaminates) were bonded to standard 4" silicon wafers at 70°C using a thermal laminator. The soft-bake, exposure and development processes were performed according to the manufacturer specifications. SUEX sheets act as negative photoresists instead of SU-8. The sheets were used to ensure height uniformity as spin coating SU-8 was seen to cause slight alterations in the height across the long flow channel.

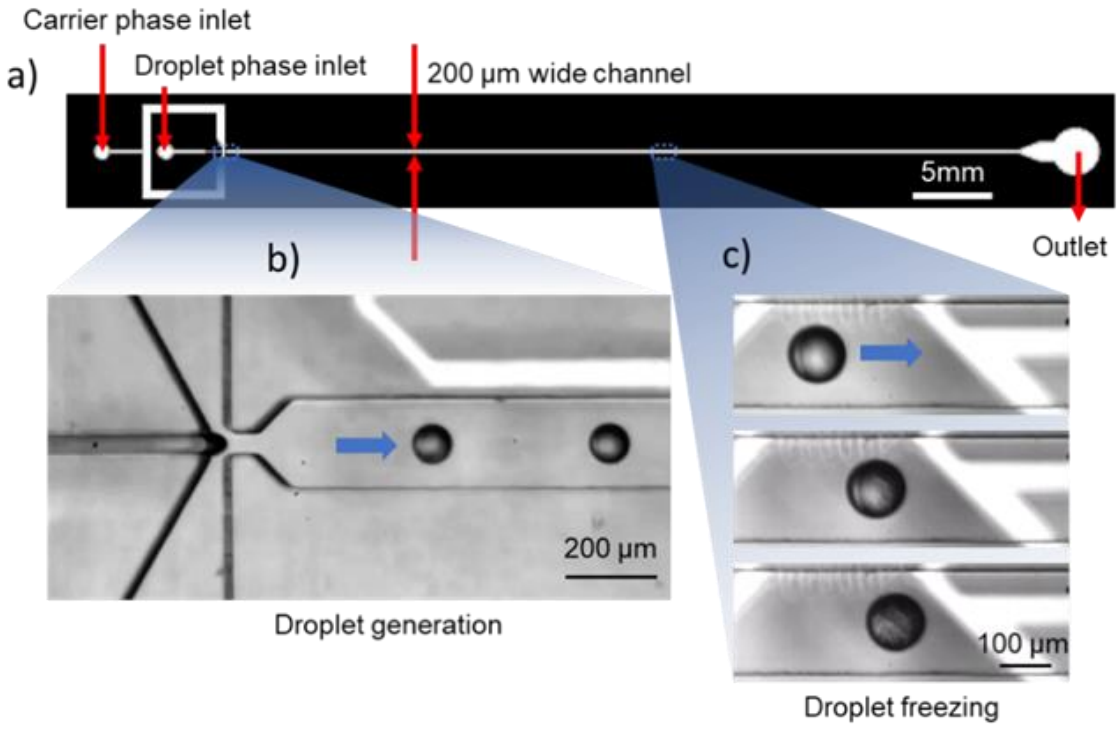


Figure 3-8: (a) Microfluidic channel design for the study. (b) Droplet generation at the flow-focusing junction. (c) Droplet ice nucleation and complete crystallization observed in the cold zone.

The leads region of the glass substrate with the PRTDs was masked off with Kapton tape (3M) and a 5µm layer of SiO2 was deposited onto the wavy sensor region using the Varian e-beam deposition system. The PDMS device was cut out from the mold using a razor, and 1.5mm diameter holes were punched for inlets and outlet with a biopsy punch (Integra Miltex Instruments) before being bonded to the substrate on the silica layer on top of the PRTDs following a 1-minute oxygen etching in a plasma cleaner. This was done to ensure that the PDMS can stick to the silica layer on the substrate since the metal film creates poor bonding with PDMS and leads to leaks.

The device was bonded to the substrate with the PRTDs at an offset from the centerline of the flow channel ensuring the wavy sensing region of the PRTD did not sit directly beneath the flowing droplets as, shown in Figure 3-8c. This bonding protocol prevents lensing of the sensor outline through the droplet from causing freezing detection errors in our DNN algorithm. Figure 3-8c also shows progression of the freezing front in a droplet, from the initiation of the droplet to complete crystallization. Events like this were rare, however, since the ROI in our studies was chosen so that droplets would freeze before entering the field of view of the camera.

Droplet generation was performed by feeding the sample of interest at the droplet phase inlet and a bulk oil phase at the carrier phase inlet (Figure 3-8b). The droplets were generated by the bulk phase pinching off the droplet phase at the droplet generation junction. Droplet size ranged from 70 to 85 μm in diameter and velocity ranged from 11.1 to 13.2 mm/s depending on the temperature but was constant at a given temperature. Additionally, the droplet spends ~1 second at the isothermal region inside the cold channel be-fore reaching the detection ROI. This relatively long residence time spent inside this isothermal region ensures that the droplet reaches thermal equilibrium with the channel before entering the ROI.

3.2.4 Temperature distribution inside the microfluidic channel

During a typical experiment, the temperatures of five cold zones were adjusted until an isothermal profile was reached. Since the freezing detection ROI was at the end of the cold region, small variations in the isothermal channel could lead to errors in reporting the

exact temperature of freezing. While multiple experimental videos for a given sample recorded a fixed temperature data point, due to small fluctuations in the flow channel in the isothermal region during the recording, the overall uncertainty in the droplet temperature was estimated as ±0.26°C.

We also wanted to check if there was any significant temperature difference between the bottom, middle and the top of the channel. To achieve this without using a second set of sensors at the top of the channel, the temperature distribution of the microfluidic channel was modeled in COMSOL Multiphysics.

The model consisted of two distinct steps. The first step was to calculate the channel temperature and check for large variations across the channel at a given location. This was done by using a steady state laminar viscous flow model in COMSOL assuming a continuous flow of carrier fluid. The input to the model was the measured PRTD array temperature on the floor of the channel and the flow inlet temperature of 5°C which was maintained by the inlet cooling block.

The second step was to use the calculated channel fluid temperatures from the first step and apply them as a transient temperature boundary condition on the droplet surface. The rationale behind this was that from the reference frame of the moving droplet, the channel temperature would be experienced as a function of time. For this step, the channel and the continuous fluid were not modeled. The reported droplet temperature was the average temperature of the whole droplet.

Figure 3-9 shows that there is a difference of less than 0.1°C between the bottom and the centerline of the channel, where the droplets flow through, at an isothermal channel temperature of about -5°C. This nonuniformity grew to about 0.2°C at -20°C but was still smaller than the experimental temperature uncertainty limit of 0.26°C as mentioned previously. As a result, we directly used the temperature measured by the PRTDs as the channel temperature for the droplet temperature calculation in step 2. Also shown in Figure 3-9 is the droplet temperature as it moves through the channel calculated from step 2. There is a temperature lag in the droplet compared to the channel temperature in the region where the temperature drops sharply before the droplet reaches the isothermal region maintained by the individually controlled cold zones. The approximately 1 second of residence time in the isothermal region enables the droplet to equilibrate with the channel temperature and lets us directly report this temperature as the droplet temperature for the frozen fraction curves.

The simulations aid in confirming the accuracy of the measured temperature, though empirical measurements of melting points of pure liquids such as water and hydrocarbons would also aid in calibrating the droplet temperatures accurately. However, these were not attempted due to significant changes required in the stage and flow device to freeze and melt droplets inside the same channel. The temperature of the droplet was also simulated at additional channel temperatures of -10, -15 and -20°C. The results are shown in Figure 3-10. There was no appreciable difference between the droplet temperature and the recorded PRTD temperatures.

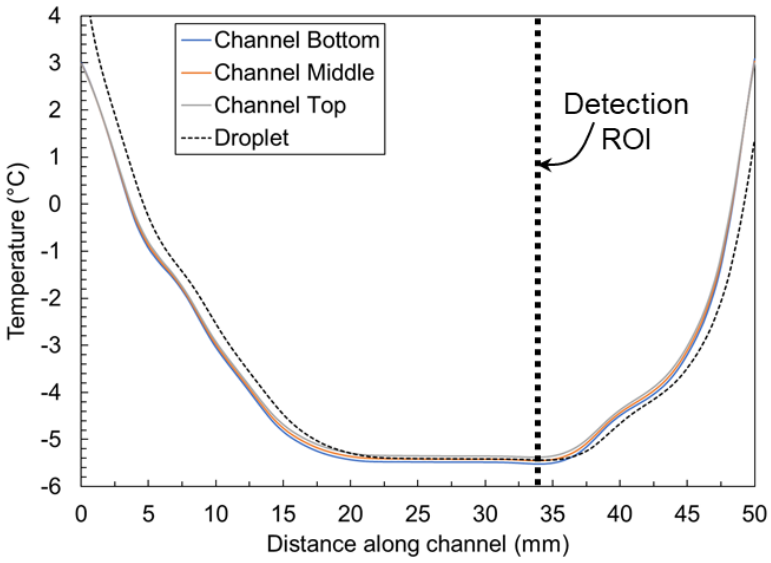


Figure 3-9: Temperature distribution in the microfluidic flow channel and droplet. Droplet diameter 70 µm, droplet and carrier fluid flow velocity 15 mm/s. The droplet resides inside the isothermal zone for ~1 second.

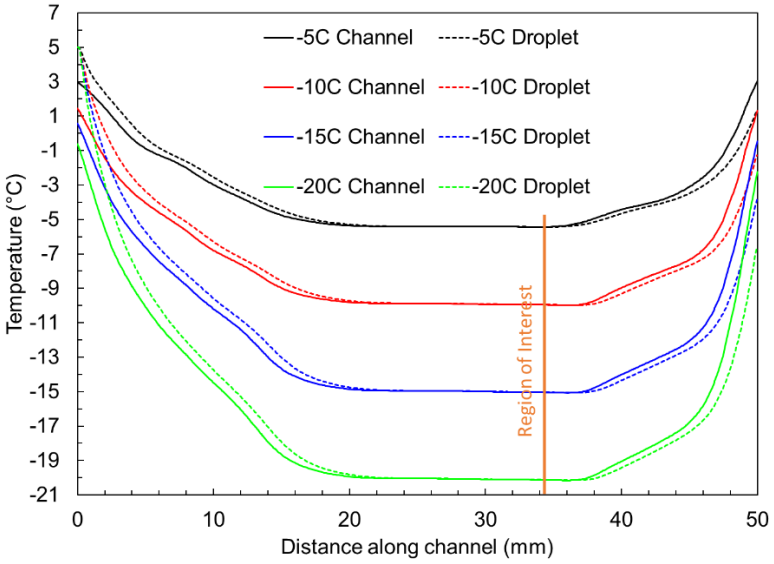


Figure 3-10: Droplet temperatures inside flow channel at different isothermal channel temperatures. The orange line indicates the ROI where frozen droplets were counted.

3.3 Results

3.3.1 Droplet freezing with different carrier fluids

Droplet freezing was initially attempted with perfluoromethyldecalin (PFMD), the same carrier fluid used by Stan et al. [80]. Additionally, Novec 7500, an engineered fluorinated heat transfer fluid manufactured by 3M, was also used. The rationale behind using these fluids was their biocompatibility and track record of being used in biomedical microfluidic devices. However, several operational challenges were encountered while using these oils instead of light mineral oil. Figure 3-11 shown the device being used with these two oils and droplets of Snomax in water being carried in the flow. The droplet phase and carrier phase flow rates for both studies were kept constant and the same microfluidic device was for both tests. The device was flushed with isopropanol and dried with air in between uses. The difference in droplet sizes between the two fluids is apparent. What is not apparent from the still pictures is that the velocities of the two cases also vary by a big amount and are almost 10 to 15 times higher than the ~10mm/s achieved with the light mineral oil.

Figure 3-11: (a) Droplets with PFMD as the carrier fluid. (b) Droplets with Novec 7500 as the carrier fluid.

To reduce the droplet velocity and allow droplets to freeze and equilibrate with the channel temperature more gradually, a sheath flow channel was added in a new device design shown in Figure 3-12. Two types of designs were created, one with a straight flow channel and one with a serpentine channel followed by a straight section. The channel width was increased to slow the droplets down. The idea behind the serpentine channel was to increase the pressure drop in the main flow channel and hence stabilize the droplet generation and flow rate by smoothing out fluctuations in the supply pressure provided by the syringe pumps. The droplet velocities with this new design was tested and results are reported in Table 3-1. Even with the increased channel cross section, the droplet velocities in Novec 7500 and PFMD were much larger than the light mineral oil.

Figure 3-12: Modified microfluidic device designs to mitigate high droplet speed and small droplet size. (a) Added sheath flow. (b) Added sheath flow and a serpentine flow resistor to increase pressure drop and flow stability.

Table 3-1: Droplet velocities in microfluidic freezing counter with different carrier fluids

Carrier fluid	**Channel width**	**Droplet speed**
Light Mineral Oil	200 μm, no sheath	~10 mm/s
Novec 7500	300 and 400 μm, with sheath	55 and 68 mm/s
PFMD	300 μm, with sheath	80 mm/s

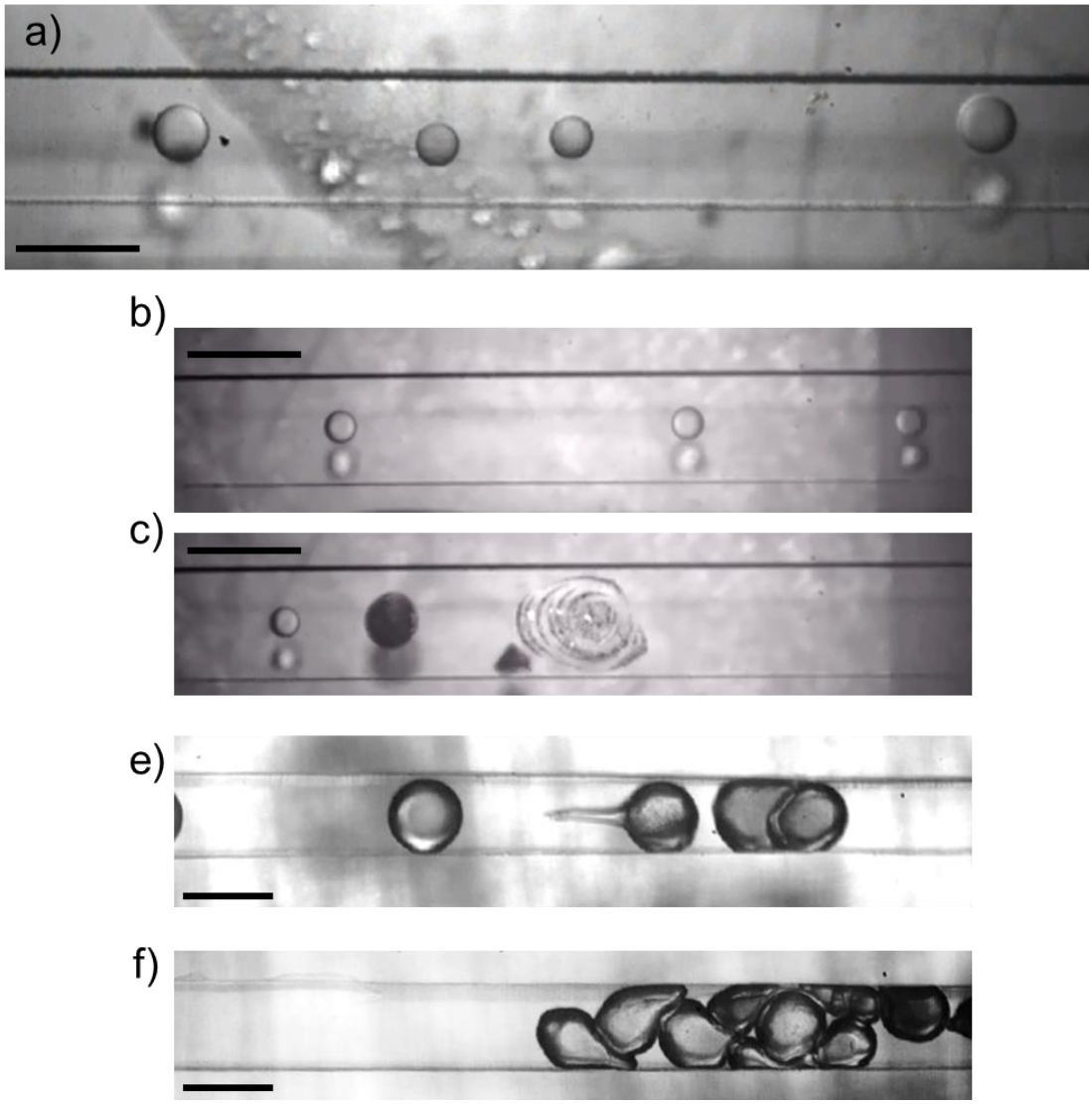


Figure 3-13: Operational issues with PFMD and Novec 7500 as carrier fluid. (a) Droplet coalescence leading to inconistent size and flow of droplets. (b-c) smaller droplets sticking together and forming larger ice plugs as they freeze. (d-e) Larger droplets creating individual ice plugs to stop the flow as they freeze. The scale bar represents 100 μm in all images.

In addition to the high velocities, there were issues with droplet size inconsistency owing to coalescence downstream of droplet formation. There was also droplet clogging

issues in the microfluidic channels after droplets frozen and became solid and started sticking to the channel walls. This has been shown in Figure 3-13. The channel clogging became a larger issue since a clogged device often led to huge pressure drops in the system. This often meant that either the tubing was pushed out of the microfluidic device, leading to the liquids dripping over the experimental setup or even worse, the device starting to leak through the bonded region between the PDMS and the substrate. And this often meant that the device became unusable after it clogged. Even if the device could be reused, it showed a tendency to clog very easily afterwards.

3.3.2 Automated freezing detection algorithms

The high-speed camera captured approximately 4,000-6,000 droplets in a single video of 4-minute duration. To determine the fraction of frozen droplets at a given temperature and determine the ice nucleation rate, a method for counting the number of liquid and frozen droplets is needed. Ideally, the method is robust and high-speed, ultimately allowing for on-chip detection with the potential for downstream actuation and sorting. Here, we explored two methods, automated detection with polarized light images and with a deep neural network from brightfield images. Both methods are benchmarked to simple manual counting from brightfield optical images.

3.3.3 Polarized intensity threshold algorithm for freezing detection

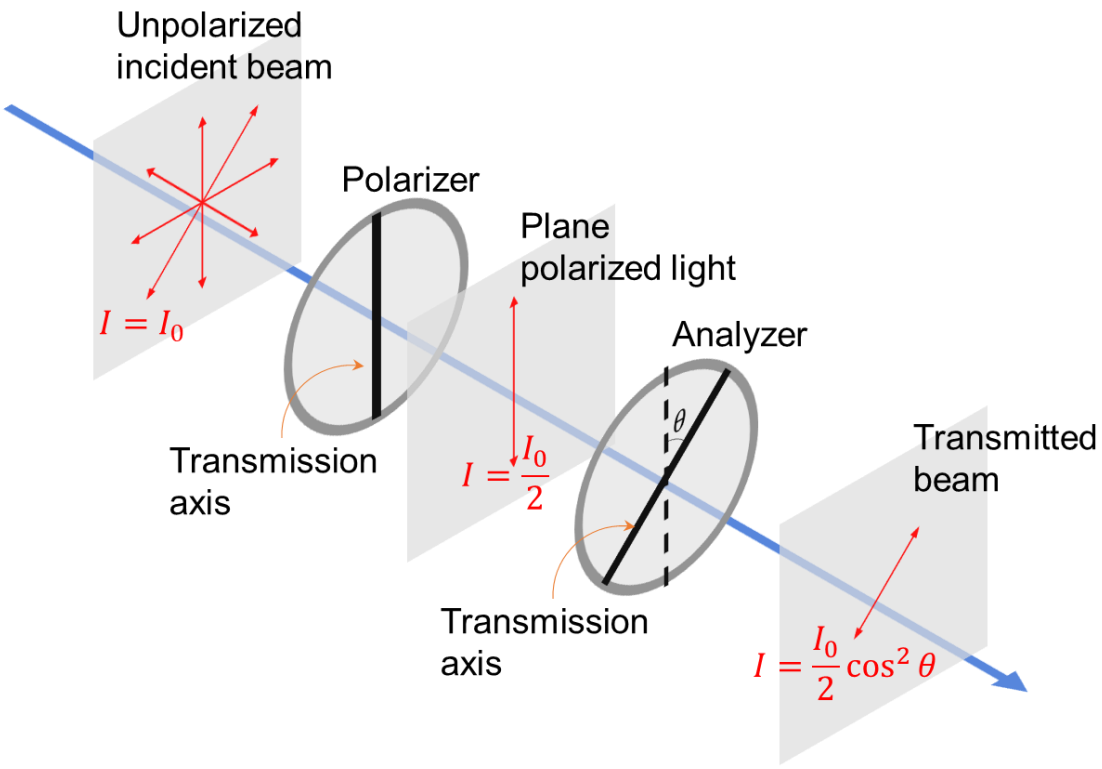


Figure 3-14: Polarizer-analyzer light filtering principle with a pair of linear polarizing filters. The first filter is called a polarizer, and the second filter is called an analyzer.

The polarizer-analyzer principle is illustrated in Figure 3-14. A light emitting source emits unpolarized light which gets polarized by a linear polarizing filer. This has two effects on the transmitted light, first, it gets linearly polarized in an axis parallel to the filter, second its intensity is halved. This light is called plane polarized. When the light next encounters another linear polarizing filter in its path, depending on the angle of this filter, the light intensity reduces according to the equation:

$$I = \frac{1}{2} I_0 \cos^2 \theta \qquad \text{Equation 3-1}$$

where, I_0 is the incident light intensity, and θ is the angle between the optical axes of the polarizer and analyzer. If the polarizer and analyzer are placed parallel to each other, the light passes through unchanged, and if they are placed perpendicular or crossed then all light is blocked, as given by Equation 2-2. These filters are made up of specific polymer sheets which polarize light by absorbing any incident light polarized parallel to their polymer chain arrangement and transmits light polarized perpendicular to the chains. Ice crystals, similar to quite a lot of mineral crystals, have a specific property called birefringence, which essentially means it has different refractive indices based on the incident polarization angle of light. These materials can effectively rotate the angle when polarized light passes through on them. If a birefringent material is placed between a set of polarizer-analyzers, it can in essence change the expected intensity of light passing through the two filters compared to when there is no material between them. This principle has been used previously to detect ice nucleation[182,183] in flow tubes and crystallization in polymer laden droplets[184].

In our setup, the polarizer-analyzer set with rotatable filters (Edmund Optics) was installed in the optical path between the light source and the camera as shown in Figure 3-15a. The light source is polarized through a plane polarizer before passing through the droplets once. The light is then reflected from the mirror coating underneath the substrate before passing through the droplets a second time. Finally, it passes through the analyzer before reaching the camera. With perpendicular alignment of the analyzer with respect to

the polarizer, almost all light passing through the liquid and PDMS layers can be extinguished leaving a dark image. In this arrangement, only the frozen droplets show up as bright regions in the image due to the birefringence of the ice crystals. As mentioned before, the birefringence changes the polarization angle of the light, which allows light to pass through the analyzer (Figure 3-15b). While use of 90° polarization is a straightforward method for rapid detection of droplets containing ice crystals, it poses several challenges from a coding standpoint. Purely liquid droplets remain completely dark and blend in with the background, making it a challenge to detect the total number of droplets that pass through the channel. Additionally, some frozen droplets appear darker than liquid drops even in polarized light since the crystal plane orientation inside a droplet is random with respect to the polarization angle of the incident light (Figure 3-15e, h). To solve this issue, the analyzer was placed at 60° and 85°, to find the optimum balance between the intensity of the frozen drops and the ability to still identify liquid drops.

To partially automate the detection of the frozen droplets with this setup, a MATLAB image analysis code was created to detect and count frozen and liquid droplets under polarized light conditions. Briefly, the code is used to calculate two intensity threshold parameters based on either type of frozen drops, brighter or darker than liquid drops. The higher threshold was selected to identify droplets that are brighter than a liquid drop and vice versa for the lower threshold. These parameters are used to analyze the remaining video frames and give a final count of liquid and frozen droplets in a video, which are typically on the order of 4 hours when played back at 60 fps and 4000-6000 droplets. This code, which takes minutes to run, saves the user hours of analysis time that

would have to be spent manually counting droplets. The polarized light detection code and usage instructions are given in Appendix D.

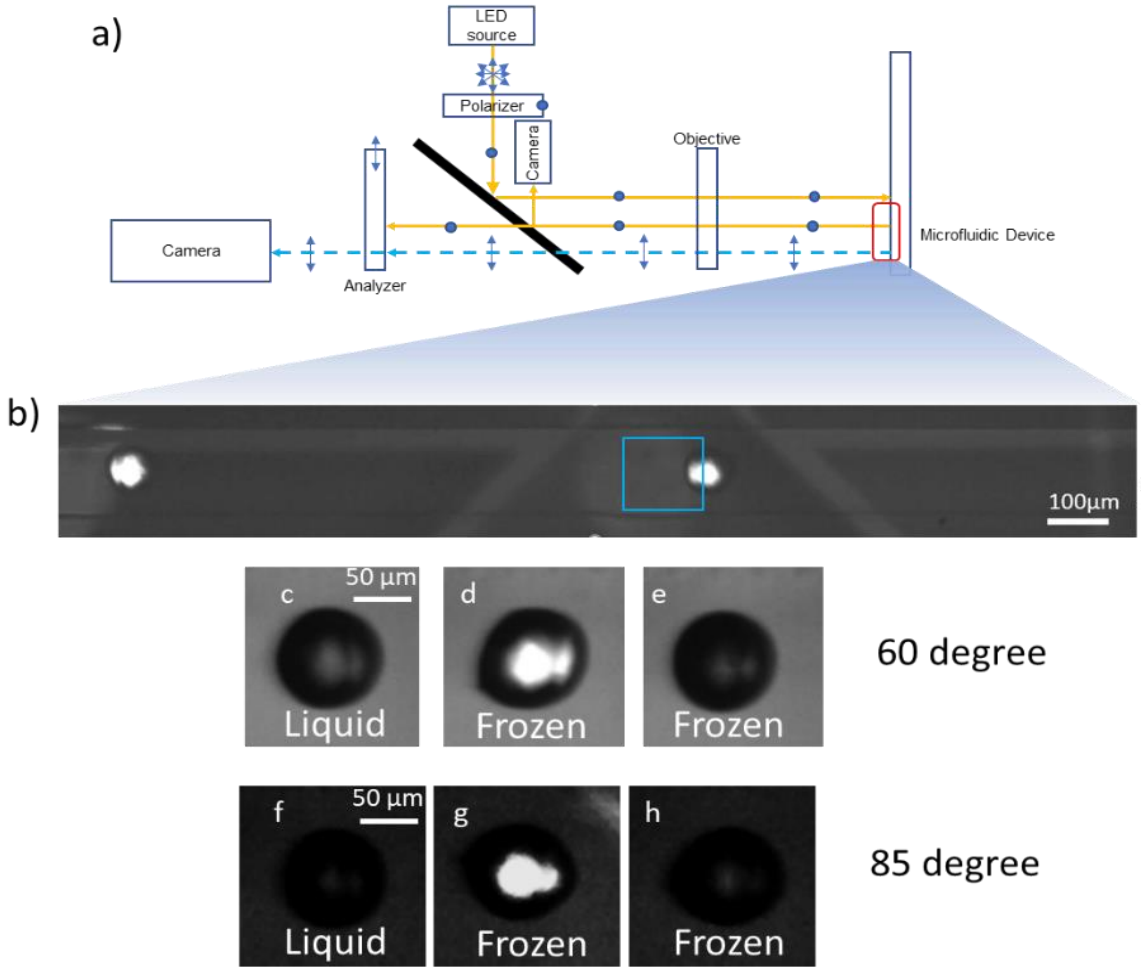


Figure 3-15: Experimental setup and freezing detection algorithm using a polarized intensity threshold method. (a) Polarized imaging optical path. (b) ROI highlighted in blue inside the flow channel. (c-h) Liquid, frozen (bright) and frozen (dark) droplets as they appear inside the ROI for different analyzer angles indicated.

3.3.4 Deep neural network algorithm for freezing detection

3.3.4.1 Brief introduction to neural networks

A neural network is a supervised machine learning algorithm which is modeled loosely on the basic functional structure of a human brain. In machine learning, a single or series of equations are used to calculate or "predict" the value of a desired variable based on values of some other input variables using a specific algorithm. The "learning" part of supervised machine learning comes from the fact that preexisting data with some context or training data is provided to the algorithm, which is learned by the algorithm and used to predict future data that is similar but not quite equal to the training data. The learning is essentially manifested in changes of numerical value of some coefficients or parameters in the equations. A neural network (or artificial neural network) specifically consists of a series of layers each with one or more nodes, i.e. a 2D matrix structure like the one shown in Figure 3-16. In this figure, the green circles are the nodes or the variables, and the black arrows are the "weights" (or parameters) with numerical values. This is a representation of a series of equations with many variables and parameters. The network is initialized with some random weights and training data is fed from the left end. In this particular example, the training data must be represented by the input variables and the values will be entered into the nodes in the 1st or input layer. The values then get multiplied by the weights connecting them to the 2nd layer and so on. In this way, the input data gets transformed and an output comes out in the form of two variable values at the rightmost or output layer. This "predicted" value is compared against what is known about the input data and the all the preceding weights are adjusted until the differences between the prediction and the known value or "ground truth" is minimized. This process is repeated for a set of input data

values, which constitute a training set. Through this training, all the weights in the network end up at a specific value, unique to the training set. Now, when an unknown or test data is fed to the trained network, that data is passed through the nodes and end up at the output which then becomes the predicted value. This type of machine learning algorithm is particularly efficient at classifying objects, rather than predicting the value of a continuous variable, although that is also possible. In classification, the output layer nodes represent different classes input variable can belong to, and the predicted values at these nodes are the probabilities of the input belonging to that class.

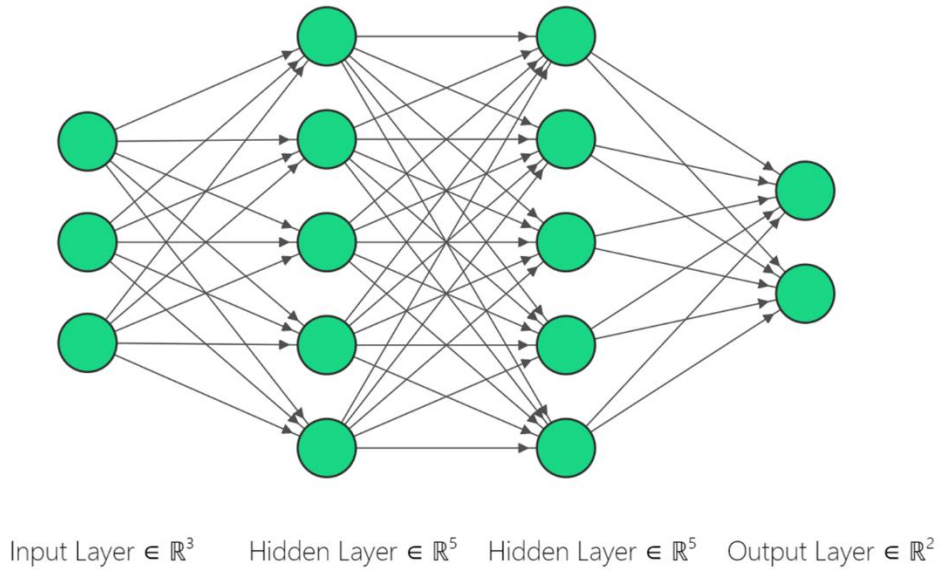


Figure 3-16: A typical Artificial Neural Network. The green circles represent nodes or cell body of a neuron, the black arrows represent weights or the interconnecting axon between neurons.

The network shown in Figure 3-16 is quite shallow, only having two hidden layers between the input and output. Also, the layers themselves have only a maximum of 5 nodes each. Classifying droplets using this kind of network would be difficult since the input can

only have three variables. Hence, any droplet image would have to be broken down into three quantifiable features. This was attempted as shown in Figure 3-17. The three features selected were the circularity or the perimeter divided by a circle of equal area, mean intensity and mean intensity gradient. As can be seen from the figure, the frozen droplets do not easily fall into any separable location based on the variables chosen. i.e. the variability between the liquid droplets is sometimes higher than the variability between a liquid and a frozen droplet.

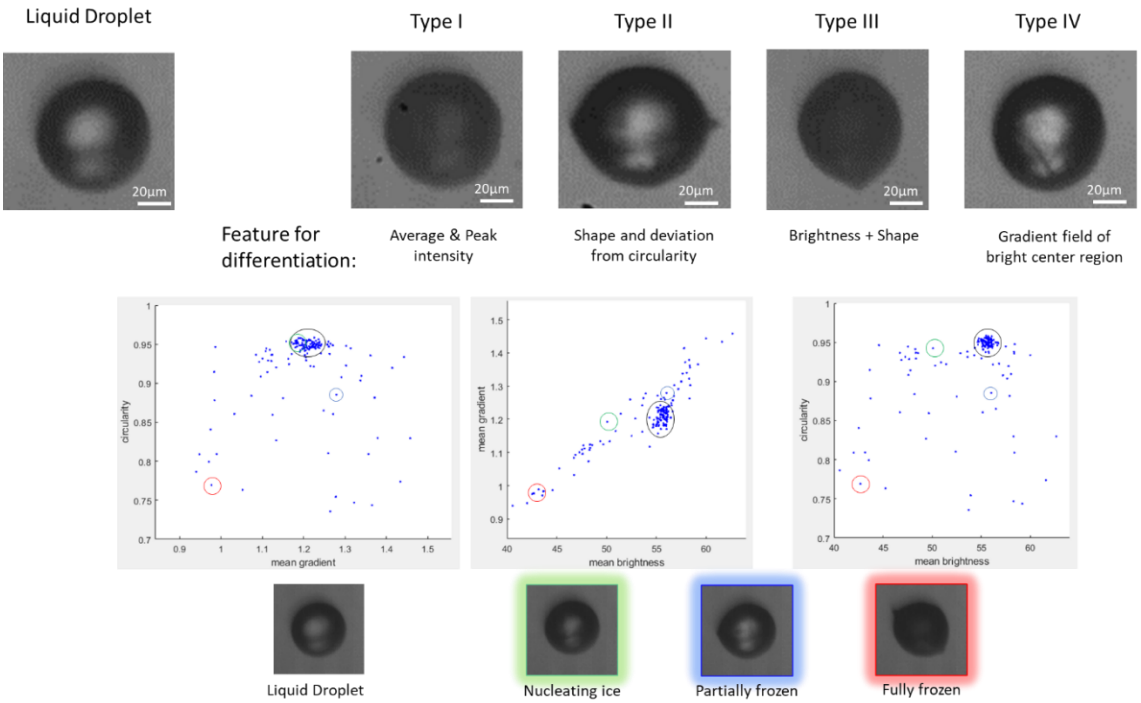


Figure 3-17: Different types of droplets, both liquid and frozen are shown in the top row. The type I-IV label represents droplets which showcase the three features selected. The plots in the middle row shows a set of droplets plotted on axes representing numerical values of these features. Each blue dot represents a single droplet. The same dataset is plotted in all three graphs. The bottom row

represents three different frozen droplets with red, green and blue glowing outlines. Same droplets are circled in the middle graphs. The liquid droplet is circle in black.

As is evident from this feature-based attempt to classify the droplets, the peculiar challenge of the huge variability in droplet images posed too difficult for a simple machine learning approach. However, the interesting fact here is that a human being is still able to identify and distinguish between the droplet. Hence an algorithm which more closely mimics the image recognition and classification abilities of humans was required for solving this problem.

3.3.4.2 Image recognition with deep neural networks

Artificial neural networks that can more closely mimic human brains are comprised of many layers with many nodes in each layer, in much higher numbers than the one shown in Figure 3-16. These networks are called deep neural networks (DNN). This has become feasible in the past decade due to order of magnitude increases in computational power available to single computers. Usually, neural networks require a lot of data preprocessing, like the attempts described previously. However, with DNNs it has become easier to train computers to recognize images with minimal user input and are used with vastly different datasets, finding use in highly diverse application areas such as facial recognition, automated driving, mail sorting, biomedical research, online marketing etc. Most of the DNNs in use for image recognition today use an approach called convolutional neural network. It works on the pixels of an image directly rather than extracted features from the image. The preprocessing required is lower since the network can learn to extract the features by itself.

A DNN is an example of a convolutional neural network. It mainly involves two operations or layers repeated several times to create a DNN like architecture: convolution and pooling. These operations are used to extract features from images. Convolutions are usually performed by creating a small filter, usually 3x3 pixels and moving it over the image to create an output feature map. In a convolution layer there are usually many filters, each filter results in a single output feature map. In this way, a single input image gets filtered into many feature maps. Next comes the pooling layer. It down samples the feature maps, by usually applying some operation like taking the maximum value or average over a defined region and reducing its overall size. In this way, only the most prominent features are down selected into a smaller image. Then a new set of convolution and pooling is applied. As the image moves through the network, it keeps multiplying into more and more feature maps and keeps getting smaller in size, with the features getting more prominent. Finally, the entire image is flattened into a row vector or pixel intensity values and passed to a final hidden layer for classification. This operation is described pictorially in Figure 3-18. FC Layer 1 and 2 are fully connected layers, like the hidden layers in Figure 3-16. The intensity of the cyan color indicates the pixel intensity as a numerical value (ranging from 0 to 255). The convolution filters are highlighting edges of certain orientations in the first layer, and then become more and more abstract and specialized in the other layers.

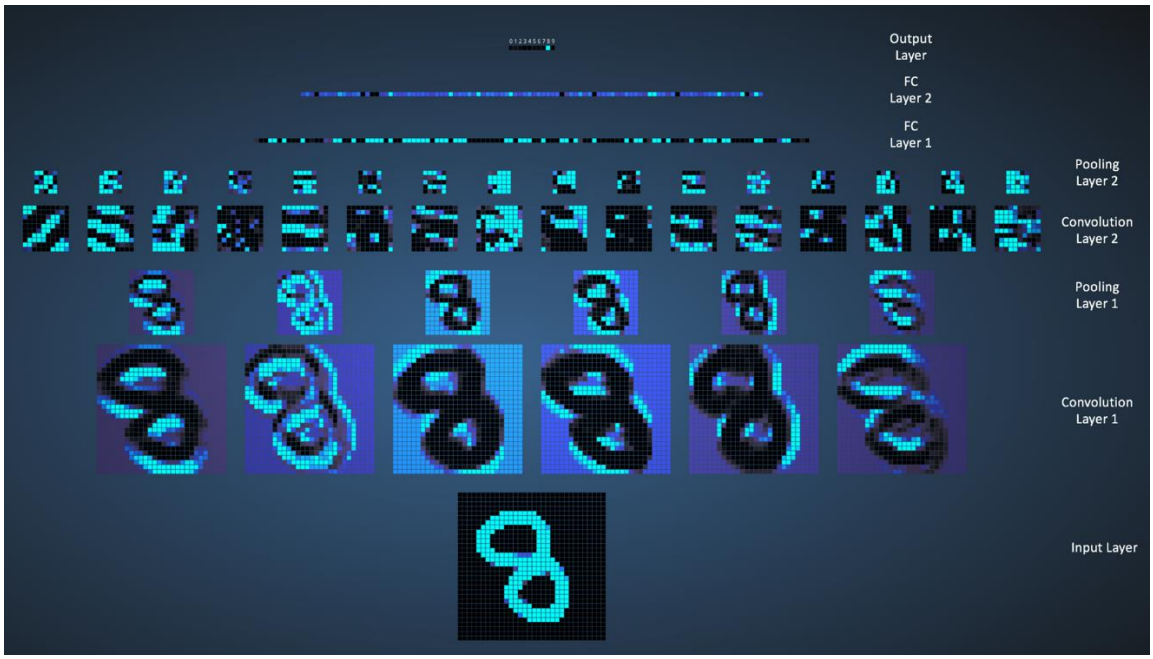


Figure 3-18: Visualizing the convolution and pooling operations on detecting a handwritten digit. Original illustration created using 2D convolutional network visualization tool (URL https://www.cs.ryerson.ca/~aharley/vis/fc/flat.html) from Harley, A. W. An interactive node-link visualization of convolutional neural networks', in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2015. Springer Verlag, pp. 867–877. doi: 10.1007/978-3-319-27857-5_77. Copyright 2015, Springer Nature.

Training a DNN to recognize images of known or labeled liquid and frozen droplets and using the trained network to detect images of test sample droplets was chosen as the simplest and most elegant method of performing frozen droplet detection. This avoids the need to extract features using separate methods and having to go through many different machine learning algorithms to try to fit the extracted features to find the best model suitable for this application.

3.3.4.3 Implementation of transfer learning in freezing detection

As mentioned before, the weights in the neural network are initialized randomly, and approach an optimum value through learning from training data. However, in deep learning, a very useful concept to speed up this learning is to initialize the weights by taking them from a pretrained network, even if it was trained on images completely different from the current application [185]. This is possible because the convolutional layers do not look at high level features of images, but focus on low level details like local arrangement of pixels leading to curves, edges etc.

For the transfer learning, the well-known AlexNet [186] was chosen. This network, available in the MATLAB Deep Learning Toolbox, has 23 layers, 1000 object classes and has been pretrained with 1.3 million high-resolution images in the LSVRC-2010 ImageNet training set. We modified this network by keeping the convolution layers intact and replacing the final three layers with 1) a new fully connected layer, 2) a SoftMax layer, and 3) a classification layer with two classes: frozen and liquid. The modified network is called dropletnet.m and is trained with liquid and frozen drop images extracted from videos taken at the highest and lowest temperatures during the experiment. The training set temperatures were carefully chosen far away from the expected temperature range where the frozen fraction was expected to be between 0 and 1. Training was performed separately for each sample tested as we found this gave more accurate results. This was probably due to changes in the background light intensity and tiny shifts in position of the camera with respect to the channel between experiments. A custom MATLAB app was developed based on the algorithm.

The optical path for bright field imaging used the same setup as the polarized imaging, with the analyzer turned to 0° with respect to the polarizer (Figure 3-19a). The microfluidic channel in this mode is shown in Figure 3-19b in a frame of video captured during an experiment. The orange box indicates the ROI which is cropped from the video and fed to the DNN for classification. Details about the algorithm, MATLAB app and usage instructions are given in Appendix E.

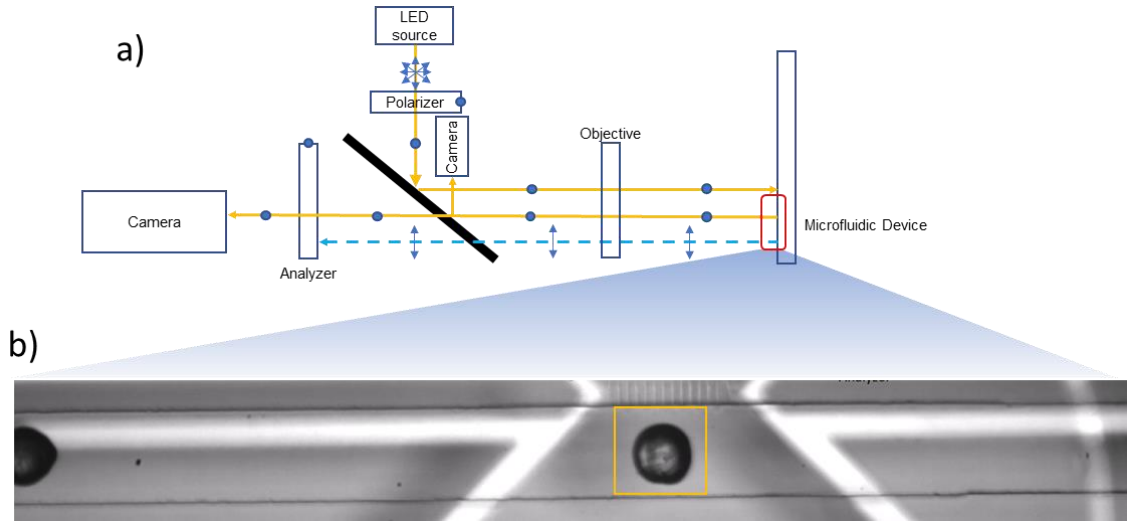


Figure 3-19: Experimental setup and freezing detection algorithm using a deep neural network: (a) Bright-field imaging optical path. (b) ROI highlighted in yellow inside the flow channel

3.4 Algorithm performance intercomparison

The polarized method and the DNN methods for automated detection are compared to results obtained from simple manual counting of frozen droplets from optical images in Figure 3-20 at four different temperatures with a model INP, Snomax. The machine learning DNN method yields results within 99.1 ± 0.7% accuracy compared to the manual

method. In contrast, it is apparent from initial trials at a few distinct temperatures that the polarized method is the less accurate of the two. The polarized light code underestimates the frozen droplet count at both 60° and 85° analyzer angles compared to manual and DNN count, discussed in subsection 3.5. As a result, the DNN was used for all Snomax results presented in Chapter 4.

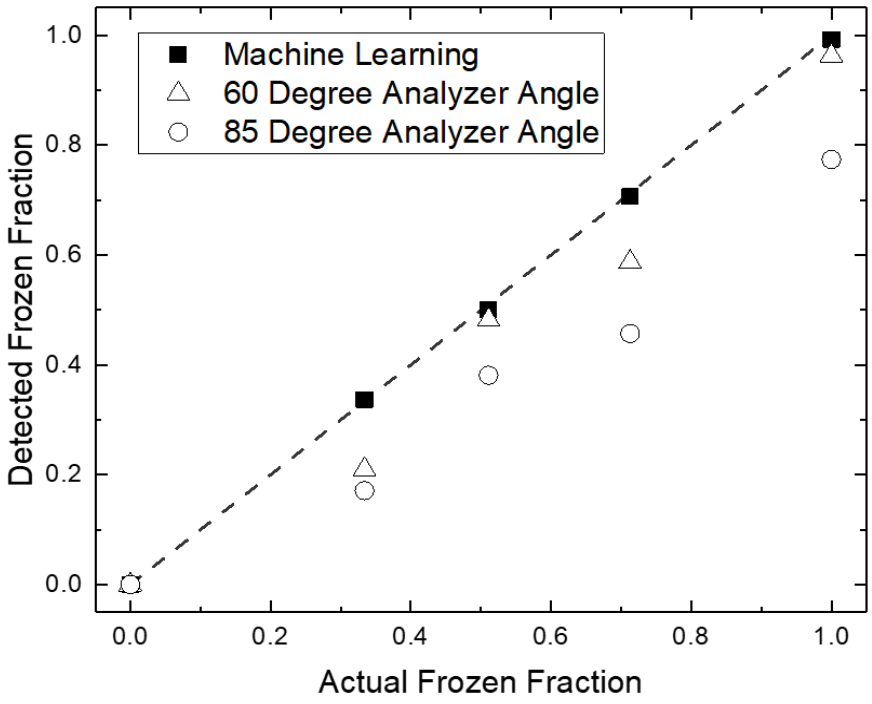


Figure 3-20: Comparison of the different methods for detecting frozen droplets. The dotted line denotes an ideal detection method i.e. where the method performs identical to a human operator.

3.5 Discussion

As mentioned in the results section, the polarized detection method undercounts the frozen fraction compared to both human and DNN count while postprocessing a video. This error occurs when the birefringence inside the frozen droplet is not pronounced. It is

especially apparent in recalescing droplets [187–189], where the droplet becomes an ice-water slush due to the presence of both multiple kinetic ice crystals and supercooled water immediately after ice nucleation. These drops do not change the polarization angle of light appreciably due to their inhomogeneity and hence are almost indistinguishable from liquid drops in the polarized lighting setup. When observed under bright-field light, it is immediately apparent that these drops have nucleated ice. Additionally, due to the random nature of the ice crystal orientation inside the droplet and the bulk rotations in the droplets as a whole as they flow through the channel, it is possible that even some completely crystallized droplets do not show up as bright spots in the polarized light. This kind of crystal orientation rearrangement could be caused by either progressive crystal formation inside the already ice nucleated droplets or droplet bulk rotation inside the channel.

Because of these shortcomings, the polarized method was not used to generate accurate datasets in our present device and working fluid combination. But we see other uses for this method, such as in a real-time active sorting device design where the frozen droplet must be detected at run-time for downstream actuation or sorting to take place. The droplet birefringence might be more pronounced in other combinations of carrier and droplet phases [74,80] which would aid the polarized method in becoming more accurate compared to manual count. This kind of approach would be economic, eschewing the need for a high-speed camera by using simpler 60 fps area scan cameras, or even using line scan cameras where a single line of pixels is captured at high framerates.

A potential disadvantage of flow-through microfluidic devices, when compared to static devices, is the high cooling rates experienced by the droplets as they cool while

flowing through a fixed temperature gradient at high velocities. For example, in our setup the droplet spends around 3.3 seconds in total inside the device, out of which 1 second is spent inside the isothermal freezing region. This imparts a cooling rate on the droplet which is dependent on the cold region temperature, ranging from 140°C/min at 0°C to 720°C/min at -20°C. A previous flow-through microfluidic INP study by Tarn et al. [74] with similar droplet sizes, velocities and channel dimensions demonstrated that droplets with relatively short residence times (~0.2 seconds) inside a cold isothermal channel could still freeze in similar temperature ranges reported by instruments with orders of magnitude higher residence times. While ice nucleation by atmospheric particles is a stochastic process, it has been reported that highly efficient INPs including Snomax [178] can be approximated with the deterministic, time independent approach we used for this study. Nevertheless, in order to adapt our device for studying atmospherically relevant particles, it would be valuable to study the effect of different cooling rates and residence times by modifying the cold stage and flow device to impart highly variable cooling rates.

We did not observe homogeneous freezing of ultrapure water down to -20°C. Attempts to reduce the temperature further were unsuccessful due to the carrier fluid viscosity; the syringe pumps stalled trying to push the liquid through the microfluidic device. Notably, studies on flow-through microfluidic INP counters [74,80] with similar droplet sizes, velocities and microfluidic channel dimensions have measured homogeneous freezing of water between -35°C and -37°C. A different carrier fluid and redesigned channels with lower pressure drop could enable homogeneous freezing measurements in future INP studies. In the absence of the background water freezing curve, this present

work is intended to highlight our platform, freezing detection algorithms and the relative effects of modifying the IN behavior due to aging and heat deactivation compared to untreated Snomax.

3.6 Conclusions

In this chapter, we developed a flow-through, temperature-controlled microfluidic platform with automated algorithms for ice nucleation detection. The custom MATLAB apps for the detection algorithms enable new users without specific knowledge of the codes or the algorithms to use the device for making measurements. The DNN app enables us to classify millions of droplets without human intervention. Future studies will include measurements of heterogeneous ice nucleation activity of less efficient INPs as well as homogeneous ice nucleation temperature of pure water, which could be made possible with modifications in the carrier fluid and pumping setup. Ultimately, our device with automated detection could be modified to allow for active sorting of droplets containing INPs with the help of actuation pumps in future for aiding chemical and biological analysis of INPs downstream of our device.

Chapter 4

Ice nucleation case study with a biological particle: Snomax

Portions of this chapter are adapted with permission from Roy, P., House, M. L. and Dutcher, C. S. (2021) 'A Microfluidic Device for Automated High Throughput Detection of Ice Nucleation of Snomax®', Micromachines, 12(3), p. 296. doi: 10.3390/mi12030296.

4.1 Introduction

Biological particles in aerosols have been studied extensively over the years [190–196] for their relevance in atmospheric processes, including IN. These particles are either microbes, pollen, algae, lichens, viruses or cellular fragments, as well as biological macromolecules such as proteins [197], fats [198], polysaccharides [199] and cellulose [200]. Studies have shown that these biological particles dominate ice nucleation in a mixed population of aerosols and only a small proportion is enough to accelerate IN in cloud droplets at warmer temperatures [201,202]. While there are many variations and mechanisms through which a biological particle nucleates ice, it is generally accepted that at temperatures above -20°C, they are the dominant sources, while below -20°C, inorganic sources such as mineral dust are the primary sources of INPs [12].

More recently, there has been focused research into correlating abundances of biological INPs in the troposphere layer with local ecotypes [168,203,204] and characterizing their emission by relative humidity [205], rainfall [206–208] and anthropogenic activity like harvests [48,209,210]. Efforts have also been made to measure abundance of existing biological INPs in clouds, snow and rainfall [211–214].

Among the specific biological INPs, tree pollen have the lowest IN temperatures, typically freezing around -20 to -15°C [215–217]. However, there is variation between different species as well. Fungal particles have the second lowest IN temperature range, around -15 to -10°C [218,219]. Finally, bacteria have by far the warmest IN range, specifically, a range of bacteria from the genera *Pseudomonous, Pantoea* and *Xanthomonus* can nucleate ice in supercooled at >-5°C [220,221].

4.1.1 Structure and mechanism of action of ice nucleating proteins and Snomax

The primary instrument for nucleating ice employed by these bacteria are ice nucleating proteins attached to their outer cell membrane. Recent field studies have shown the presence of these bacteria at large numbers in high altitudes [222] and has been identified as an important factor in cloud microphysics [208,218]. Specifically, the bacterium *Pseudomonas syringae* which has been known to nucleate ice at -2°C [223] and as a result it has been fragmented and used as a commercial product called Snomax for making snow in ski slopes at warmer temperatures. The transmembrane protein responsible for ice nucleation in *P. Syringae* is InaZ. It consists of ~1200 residues and has a β-helix dominant structure [224,225] similar to antifreeze proteins in insects [226].

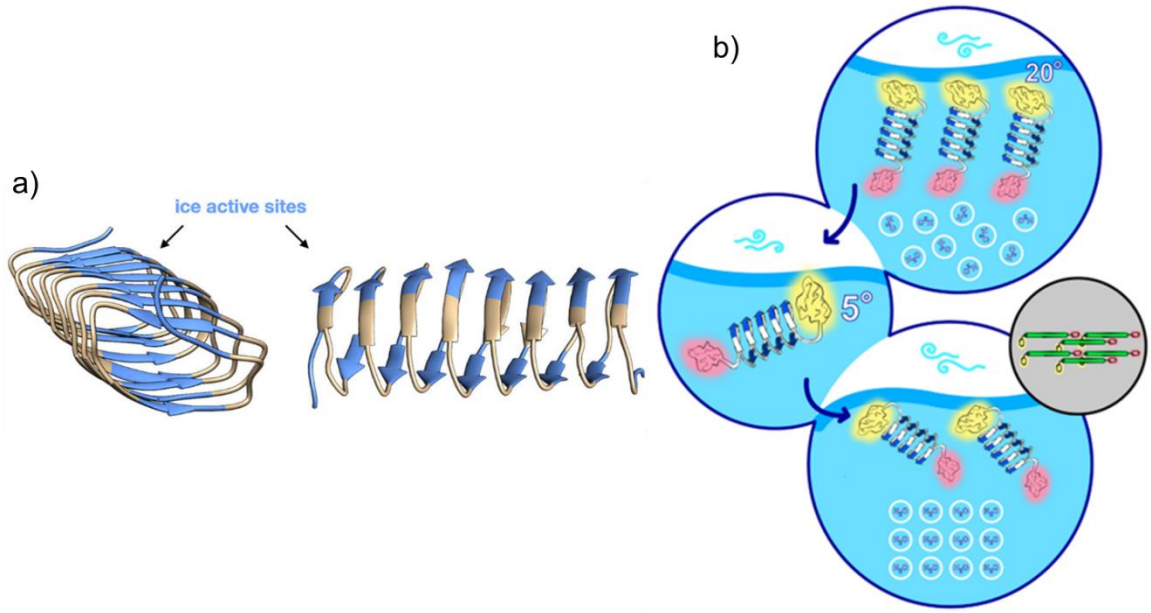


Figure 4-1: (a) Cartoon representation of the inaZ membrane protein showing the repeated β-helix structure. This figure shows the repeated blue segments in the backbone which are ice nucleation active due to the repeated nature and the precise spacing which matches that of a hexagonal ice lattice. (b) Temperature dependence of the orientation of the membrane proteins. The top bubble shows the proteins buried in the membrane and disordered water molecules at 20°C. As the temperature is lowered to 5°C, the molecules orient themselves more parallel to the membrane and exposes the ice nucleating sites to the water, thus ordering the molecules. The small gray inset shows a zoomed-out view of the individual staggered protein strands lying parallel to the membrane. Adapted with permission from Roeters, S. J.; Golbek, T. W.; Bregnhøj, M.; Drace, T.; Alamdari, S.; Roseboom, W.; Kramer, G.; Šantl-Temkiv, T.; Finster, K.; Pfaendtner, J.; Woutersen, S.; Boesen, T.; Weidner, T. Ice-Nucleating Proteins Are Activated by Low Temperatures to Control the Structure of Interfacial Water. Nature Communications. 2021, 12 (1), 1–9. https://doi.org/10.1038/s41467-021-21349-3. Copyright © 2021, The Author(s) (CC BY 4.0).

The β-helix structure of InaZ and its temperature dependent water templating is shown in Figure 4-1a. Here "templating" refers to the ability of the protein's repeated helical structure to create hydrogen bonds with water molecules in solution, and due to the specific spatial orientation of the residues, mimicking a hexagonal ice crystal like structure on a molecular level. Ironically, this ice nucleating protein and several anti-freeze proteins work on this same basic templating principle, i.e. controlling the order and arrangement of water molecules through hydrogen bond interactions. In the case of ice nucleating proteins, templating leads to an increase in the ice seed size by addition of a large number of these proteins in close proximity, thus lowering the nucleation barrier for supercooled water to phase change into ice. This effect gets more pronounced as the temperature is lowered, with the proteins orientation changing in the membrane, as evident from Figure 4-1b.

Comprehensive knowledge about the nature of biological INPs and their mechanism of action is required in order to explore the aerosol-cloud-climate interaction and the distribution patterns of these particles in the atmosphere. In this effort, instruments which can detect INPs in samples with high-throughput and high level of accuracy can be indispensable. In this chapter, Snomax, was used as a proxy for biological INPs in a case study in order to prove the viability of the flow-through droplet freezing counter described in Chapter 3.

4.2 Materials and methods

4.2.1 Chemicals used

Light mineral oil (CAS 8042-47-5, Sigma Aldrich) was used as the continuous phase in all the experiments and HPLC grade water (CAS 7732-18-5, Thermo Fisher Scientific) was used as the dispersed phase. Snomax dissolved in the water was used as a model INP for our case study. Snomax was received in dry pellet form and stored in a -20°C freezer between uses. On the day of the experiment, pellets were dissolved in water at a concentration of 1mg/ml in 50ml sterile centrifuge tubes (Nunc 50mL, Thermo Fisher Scientific) by initially vortexing and then shaking manually until the solution became visibly homogeneous.

The Snomax solution and mineral oil were loaded into 10ml gas tight glass syringes (SGE Inc.). Between each run, the syringes were emptied and cleaned with methanol, distilled water and isopropanol and dried with N_2 gas. Syringe filters (Millex 0.2µm, Millipore Sigma) were added to filter out any large undissolved Snomax particles and contaminants such as fibers in the syringes. Polyethylene tubing with ID 0.047", OD 0.067" (BD Intramedic, BD Medical) was used to connect the filter outlets to the microfluidic device. Syringe pumps (Harvard Apparatus) were used to pump the liquids inside the microfluidic device at a steady rate.

4.2.2 Aging and heat treatment of Snomax

A temperature-controlled platform based on Stan et al. [80,180] was constructed with multiple independent temperature zones, to control the droplet temperatures inside the microfluidic channel. The design allows a temperature gradient along the flow channel. Figure 1a shows our platform design with seven upper copper blocks (called cold zones)

with the microfluidic device placed on top. The cold zones are in thermal contact with separate heat exchangers (called cooling blocks) with cooled liquid flow going through them to maintain a fixed temperature.

4.2.3 Fourier Transform Infrared Spectroscopy (FTIR)

For the FTIR study, samples were prepared by dissolving Snomax in D_2O (Sigma-Aldrich, MO) at 1g/ml which resulted in a highly viscous paste-like consistency. This process was developed to eliminate the H_2O scissoring band overlap with the protein amide-I band as described in Liu et al [227]. Additionally, the smaller amount of solvent ensures a stronger Snomax signal above the background spectrum. The samples were measured in an attenuated total reflectance (ATR) accessory (MVP-Pro, Harrick Scientific). The ATR is equipped with a diamond sampling crystal and is internally purged with dry air. A mercury cadmium telluride detector cooled by liquid nitrogen was used for the study. Each spectrum is averaged over 64 consecutive readings measured with resolution 4 cm^{-1} in the wavenumber range 4000-650 cm^{-1}. Omnic 9.2 software, supplied with the instrument, was used to fit the amide-I region (1600-1700 cm^{-1}) of the spectra [228] with Gaussian peaks to isolate relative abundance of secondary structures of the *inaZ* protein in the samples following the treatments. While the effects of heat treatment on proteins are more straightforward, aging requires the proteins to interact with the solvent in a more direct way than heat treatment. Additionally, literature suggests that D_2O has a different effect on the stability of certain proteins than H_2O [229]. As a result, to avoid introducing another variable into the experiment, the aged Snomax samples were not measured with FTIR.

4.3 Results

Figure 4-2 shows curves of frozen fraction from Snomax samples from our microfluidic set-up compared to other methods. The microfluidic experiment is performed by first setting the cold zones to a desired temperature and reading the resulting temperature inside the microfluidic channel. Adjustments to compensate for the thermal connection are made by either changing the individual set point temperatures for the cold zones and/or the temperature of the underlying liquid nitrogen cooling block until a uniform temperature is reached over most of the microfluidic channel. Once the thermal test conditions are reached, a video of the droplets is recorded, and the frozen fraction is calculated by the DNN using the formula:

$$f(T) = \frac{N_f(T)}{N_{total}} \qquad \text{Equation 4-1}$$

Where, $f(T)$ is the frozen fraction at temperature T, $N_f(T)$ is the number of frozen droplets, and N_{total} is the total number of droplets.

This process was repeated over multiple temperatures to generate the curve shown in Figure 4-2. Each point in the curve has data from three independent samples or around 18,000 droplets. The horizontal error bars indicate uncertainty in freezing temperature measurement and the vertical error bars indicate standard deviation of the frozen fraction measured. Compared to [74,79,230], our sample starts to nucleate ice at lower temperatures, at around -4°C and does not reach a frozen fraction of 1 until -8°C. However, this curve is in an overall warmer zone compared to [73,231,232]. The differences in the temperature ranges are

due to the differences in the quality (or age) of the Snomax samples. Specifically, it was found that different batches of Snomax create significant variation in the ice nucleation onset temperature [231]. The measurement method used [232] can also introduce variability in the IN temperature measured.

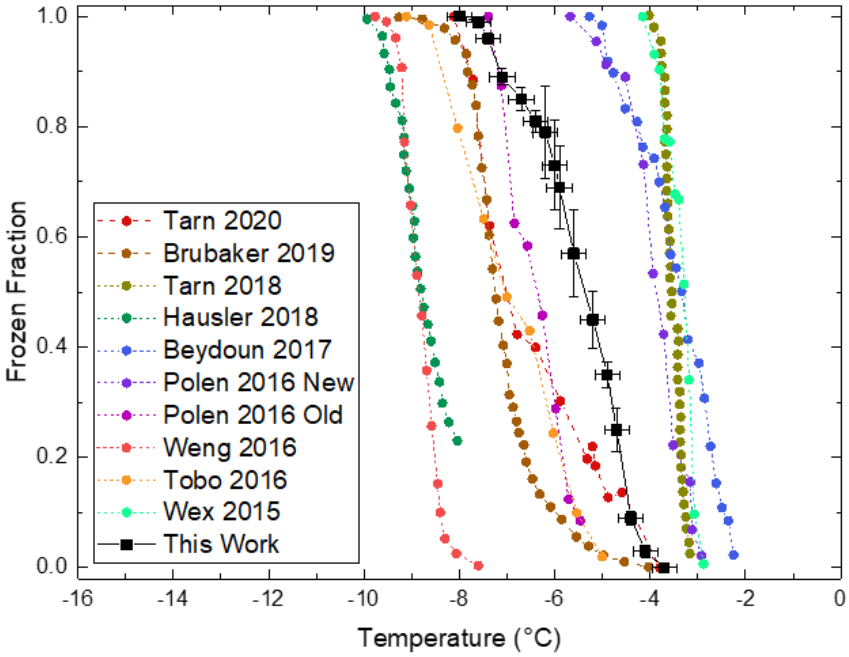


Figure 4-2: Plots showing the frozen fraction of Snomax in our microfluidic setup vs. experimental data from literature[73,74,76,77,79,230–233]. The horizontal error bars in the data represent the combined temperature uncertainty of the droplets due to the temperature variation along and across the isothermal channel during experiments. The vertical error bars represent the standard deviation from three independent samples.

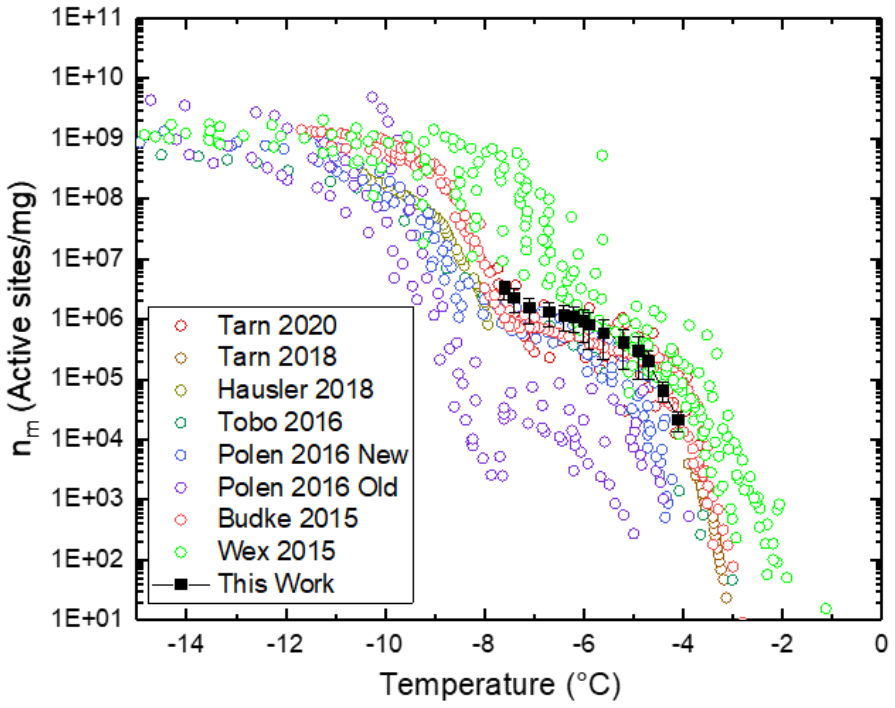


Figure 4-3: Plots showing the ice nucleation site density per unit mass of Snomax and comparison with literature data [73,74,79,178,230–232].

Heterogeneous IN, such as the nucleation resulting from using Snomax as an INP here, can be modeled using both a deterministic time-independent approach or a classical nucleation theory based stochastic time-dependent approach [12]. Following methods employed by [73,74,231,232] for Snomax and for simplicity, we adopt the deterministic approach and report temperature dependent IN active site density $n_m(T)$ using the equation [234] put forth by Vali [234]:

$$INP(mL^{-1}) = \frac{-\ln(1 - f(T))}{V_{droplet} \times C_m} \qquad \text{Equation 4-2}$$

where, $V_{droplet}$ is the droplet volume and C_m is the Snomax concentration in the droplet. A comparative plot is shown in Figure 4-3. It shows that our device captures the trend in Snomax samples quite well when compared to other devices. Snomax has been shown to be an aggregate of three different classes (A, B, C) of INPs which are responsible for nucleating ice at different temperature ranges [235]. Our Snomax sample freezes between -4.1° to -7.9°C which indicates the presence of class A (freezes above -4.5°C) and class B (freezes between -6.5°C and -4.5°C) INPs with some class C (freezes below -7.5°C) INPs. Recent studies [232,236] postulated that class A and class C INPs dominate the population and there are very few class B INPs present in Snomax which explains the steep rise at the warmest and coldest regions of the curve with the intermediate plateau.

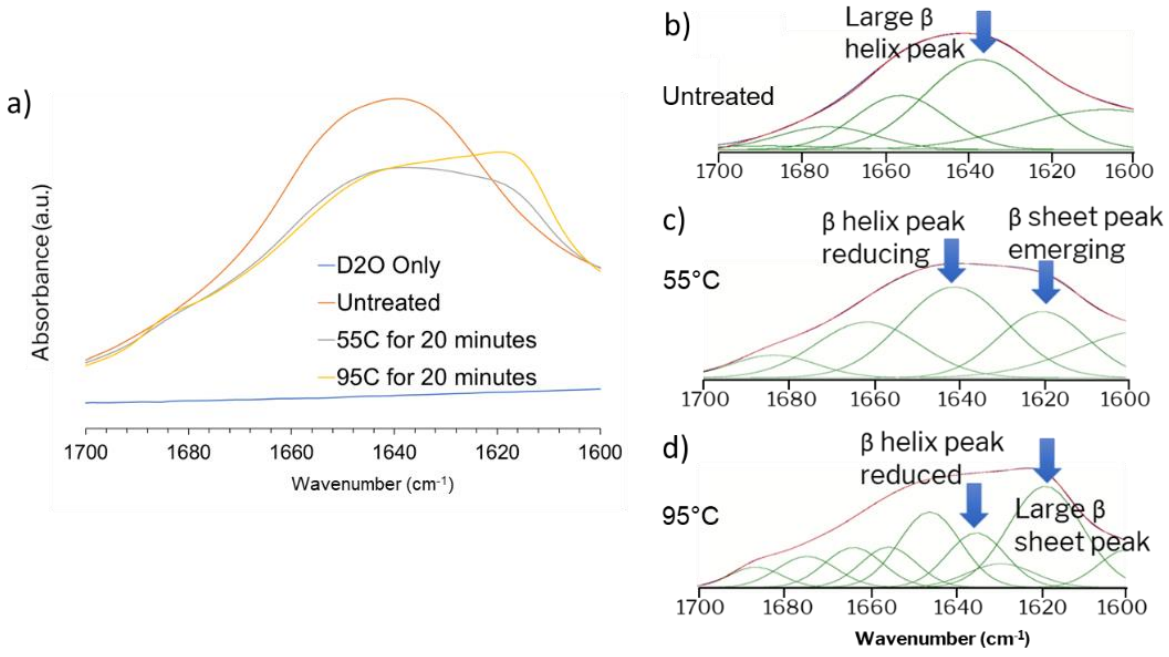


Figure 4-4: a) FTIR spectra of D_2O, untreated, 55°C and 95°C treated Snomax samples in the Amide-I region; (b, c, d) Peak resolve analysis of Amide-I region for distinguishing secondary

structure of the ice nucleating protein *inaZ* in Snomax with the heat treatment conditions for the samples indicated in the image.

A previous study on Snomax by Polen et al. [231] investigated the detrimental effects of long term storage, which also partially explains some of the variability in the freezing temperatures between results seen in Figure 4-2. Another study investigated antifreeze protein induced inhibition[236] of IN in Snomax. We chose to study the effect of heat and aging at room temperature by heat treating Snomax samples at two different temperatures as shown in Figure 4-4a. FTIR spectra of the treated samples (Figure 4-4b) show that there is a significant difference between the positive control of untreated Snomax, the negative control of D_2O and the heat treated samples. This difference is significant in the Amide - I region which reveals the secondary structures of proteins (Figure 4-4c). A Gaussian peak fitting method was used to resolve the peaks in this region. The peaks are shown in Figure 4-4(b-d) and the corresponding assignments are given in Table 4-1.

Table 4-1: Peak assignments for the Snomax FTIR samples using Gaussian fitting

Sample	**Center (cm^{-1})**	**Height (cm^{-1})**	**Full width at half maximum (cm^{-1})**	**Area**	**Secondary structure**
Untreated	1606	0.0711	46.7042	3.5361	β-sheet
	1636	0.1304	31.5232	4.3768	β-helix
	1655	0.0829	25.2748	2.2297	α-helix
	1672	0.0314	27.1489	0.9061	Turns

55°C heat treated	1594	0.0502	40.0102	2.1385	β-sheet
	1619	0.0698	24.1509	1.7948	β-sheet
	1639	0.0958	28.819	2.9389	β-helix
	1660	0.0588	28.0399	1.7561	α-helix
	1681	0.0238	23.0015	0.5832	Turns
95°C heat treated	1596	0.0402	16.0521	0.686	β-sheet
	1616	0.1076	20.552	2.3542	β-sheet
	1626	0.0252	18.5359	0.4967	β-sheet
	1632	0.058	16.2145	1.0007	β-sheet
	1643	0.0803	17.162	1.4671	β-helix
	1653	0.0429	15.3516	0.7013	α-helix
	1661	0.0424	16.1014	0.7274	Turn
	1672	0.0331	16.3272	0.575	Turn
	1684	0.0217	14.2168	0.3285	Turn

Across all the samples, the area, and the full width at half maximum (FWHM) β-helix structure significantly decreases in the heat-treated samples, thus indicating a shrinkage in the amount of β-helix in the Snomax sample. Concurrently, the same

parameters for the β-sheet structure increases, indicating an overall transition in the secondary structure of the protein. This structure change is irreversible as it does not revert back to the native untreated state after the sample is cooled to room temperature. Interestingly, the α-helix structure also reduces with temperature. The changes on the whole correlate with the heat treatment temperature, i.e. 95°C shows the largest change compared to the untreated sample. These changes indicate thermal denaturation of the ice nucleating protein in Snomax.

Figure 4-5 shows the frozen fraction of heat-treated samples as a function of temperature in H_2O measured with our microfluidic platform. The frozen fraction vs. temperature curve of the 55°C treated sample moves about 1.3°C towards the colder temperatures indicating partial deactivation of the active ice nucleating protein InaZ in Snomax due to initiation of heat-induced damage. The 95°C treated sample shifts dramatically to the colder region, with a >7°C decrease in onset and a flattened frozen fraction curve, indicating a drastic decrease in the IN ability, likely due to intensive thermal degradation of the IN sites.

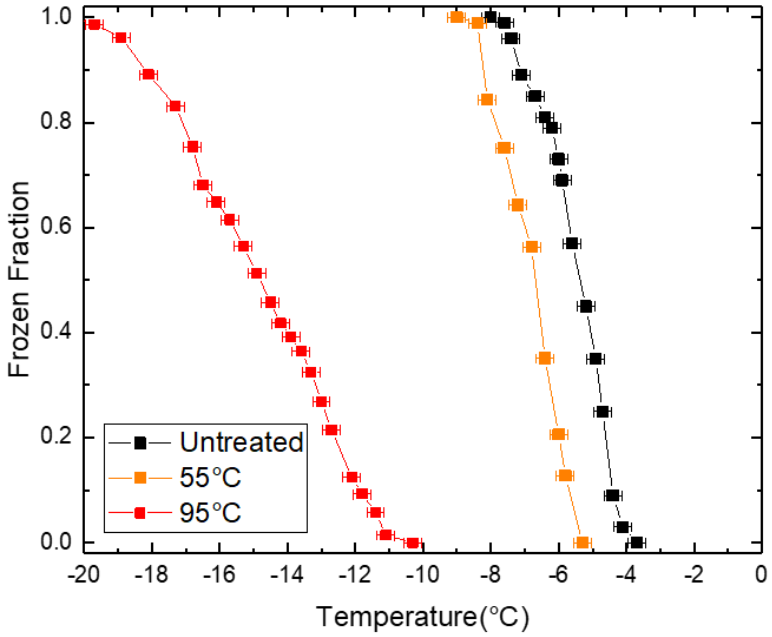


Figure 4-5: Frozen fraction of heat treated Snomax samples as a function of freezing temperature.

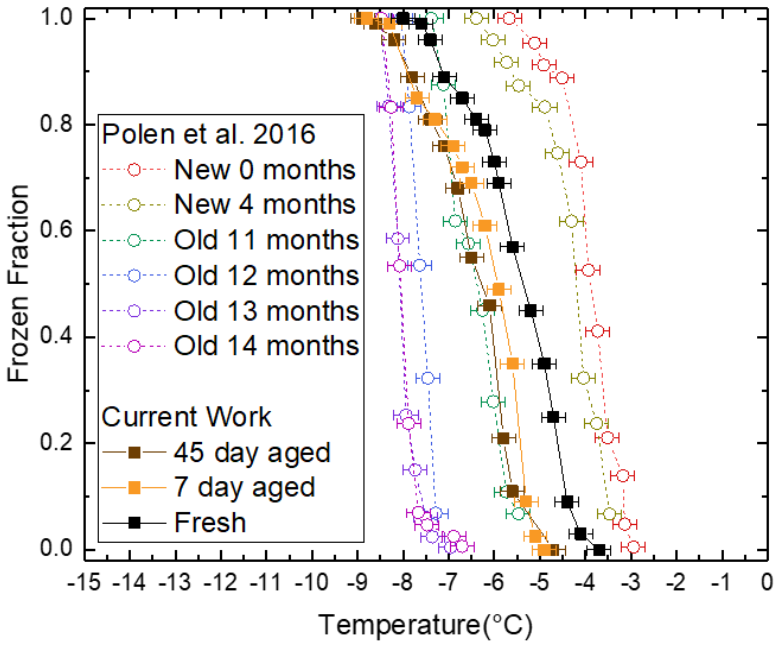


Figure 4-6: Frozen fraction of room temperature aged Snomax samples as a function of freezing temperature.

Results of Snomax aging at room temperature are shown in Figure 4-6. The ice nucleation temperature range from our study is comparable to Polen et al. [231]. Our samples were obtained more than a year before the experiments and stored in a freezer at -20°C. Storage at such low temperatures appears to have reduced the negative impact of aging, compared to the results of Polen et al. at 12 months of age. Also, once taken out of storage, subsequent aging had some impact on the IN profiles, though there was no significant difference between the aged samples themselves. Our untreated or "freshly thawed samples" freeze at temperatures colder than the "New" samples from Polen et al. The 7-day and 45-day aged samples show about 1°C lower freezing onset temperatures when compared to our untreated sample. Each aged and untreated data point on the curve in Figure 4-6 represents the mean of three repeated measurements on the same experimental sample. The error bars on the untreated sample have been omitted for consistency.

4.4 Discussion

A potential disadvantage of flow-through microfluidic devices, when compared to static devices, is the high cooling rates experienced by the droplets as they cool while flowing through a fixed temperature gradient at high velocities. For example, in our setup the droplet spends around 3.3 seconds in total inside the device, out of which 1 second is spent inside the isothermal freezing region. This imparts a cooling rate on the droplet which is dependent on the cold region temperature, ranging from 140°C/min at 0°C to 720°C/min at -20°C. A previous flow-through microfluidic INP study by Tarn et al. [74] with similar droplet sizes, velocities and channel dimensions demonstrated that droplets with relatively short residence times (~0.2 seconds) inside a cold isothermal channel could still freeze in

similar temperature ranges reported by instruments with orders of magnitude higher residence times. While ice nucleation by atmospheric particles is a stochastic process, it has been reported that highly efficient INPs including Snomax [178] can be approximated with the deterministic, time independent approach we used for this study. Nevertheless, in order to adapt our device for studying atmospherically relevant particles, it would be valuable to study the effect of different cooling rates and residence times by modifying the cold stage and flow device to impart highly variable cooling rates.

We did not observe homogeneous freezing of ultrapure water down to -20°C. Attempts to reduce the temperature further were unsuccessful due to the carrier fluid viscosity; the syringe pumps stalled trying to push the liquid through the microfluidic device. Notably, studies on flow-through microfluidic INP counters [74,80] with similar droplet sizes, velocities and microfluidic channel dimensions have measured homogeneous freezing of water between -35°C and -37°C. A different carrier fluid and redesigned channels with lower pressure drop could enable homogeneous freezing measurements in future INP studies. In the absence of the background water freezing curve, this present work is intended to highlight our platform, freezing detection algorithms and the relative effects of modifying the IN behavior due to aging and heat deactivation compared to untreated Snomax.

Our case study shows that our current setup is capable of detecting ice nucleation with a very high throughput without manual intervention. The frozen fraction curves for the untreated and aged Snomax are comparable to literature data compiled using a multitude of other methods by Wex et al. [232] such as an acoustic levitator, a wind tunnel

and droplet arrays on cold plate methods. Microfluidic approaches, both with surfactants to prevent droplet coalescence and stabilize droplets [73,74], and without surfactants [76], have been compared as well and show similar trends in terms of both the frozen fractions and the active ice nucleation site densities. Our setup currently has the highest throughput (1500 droplets/minute) among the microfluidic methods reviewed here primarily due to the use of the DNN to aid droplet classification.

Aging Snomax samples at room temperature reduced the IN activity by a small amount which resulted in the reduction of the IN temperature curve when compared to the fresh samples. However, changing the duration of aging from 7 to 45 days did not show appreciable difference, indicating that the reduction in IN activity had already taken place before the 7 day measurement had been made. Perhaps a much longer aging of the sample at room temperature is required before it can degrade enough to cause a large shift in the frozen fraction data. As a result, speeding up and amplifying the degradation was attempted using heat treatment. We also measured the secondary structure of the ice nucleating protein *inaZ* which is primarily responsible for templating water molecules to form ice and lowering the nucleation barrier [237,238]. It has been reported that the β-helix region of this protein is the primary IN site [224,239] and its presence can be detected using FTIR techniques [240] from the Amide – I region of its IR spectra. A study into the Amide – I region of purified IN proteins from Snomax was published recently [241], where the authors concluded that heating above 55°C caused irreversible changes to the protein structure and which resulted in reduced IN activity. We demonstrated for the first time that heat treatment causes most of the β-helix secondary structure in untreated Snomax to convert to a β-sheet or strand

like structure using FTIR, and our microfluidic setup was able to show an associated decrease in the IN temperature. Furthermore, we showed that the extent of heat treatment correlated with the β-helix conversion and IN temperature reduction. Aged Snomax samples were not studied in the FTIR-ATR. While the effects of heat treatment on proteins are straightforward, aging in solution requires the proteins to interact with the solvent in a more direct way than heat treatment. Additionally, literature suggests that D_2O has a different effect on the stability of certain proteins than H_2O [229]. As a result, studying aged Snomax would require creating a dilute solution in H_2O and studying the sample in transmission mode. This would be an interesting measurement in future work with biological INPs.

4.5 Conclusions

In this chapter, we performed a case study with Snomax INPs with our automated high-throughput microfluidic droplet freezing counter. We also performed an investigation into the molecular nature of Snomax IN activity with FTIR and showed the effect of heat on the secondary structure of the ice nucleating protein, and measured the corresponding IN activity reduction in our microfluidic platform. It was found that aging at room temperature and mild heating at 55°C had a minor detrimental impact on the ice nucleation ability of the Snomax. However, heating at 95°C reduced the ice nucleating ability of Snomax drastically. We generated millions of droplets and showed congruence with literature data on Snomax.

Chapter 5

Ice nucleation and phase study in marine samples

Portions of this chapter are adapted with permission from Roy, P., Hill, T. C. J., Mael, L. E., Mehndiratta, L., Peiker, G., House, M. L., Grassian, V. H., Demott, P. J., Dutcher, C. S. (2021) "Ice nucleating activity and particle morphology of bulk seawater and sea surface microlayer", ACS Earth and Space Chemistry, submitted for publication.

5.1 Introduction

Oceans cover 71% of the Earth's surface and hence sea spray aerosol (SSA) is potentially a major source of INPs, as well as cloud condensation nuclei (CCN), to the atmosphere [242]. SSA is created by wave breaking, bubble bursting, or through gas phase emissions and secondary organic aerosol formation in the atmosphere [243]. Characterization of marine INP sources is an active area of research for both laboratory and field studies [12,244], and INPs have been measured in bulk seawater as well as the sea surface microlayer (SSML) [245–255]. These studies found that the SSML is more enriched with INPs compared to the bulk overall [246–248] and that the IN activity of SSA can be explained by the INP type and concentration in the underlying bulk seawater and SSML [256]. The exact INP

concentration, and the variation of this concentration between the bulk and SSML, depends on multiple factors such as the chemistry of the seawater [256], composition of the SSML [257] and the abundance, size and characteristics of the biogenic INP sources [246,257,258].

Despite a growing consensus that the ocean is a significant source of INPs, global climate models currently underestimate the outgoing shortwave radiation in ocean dominated regions [259]. This is due primarily to a gap in the understanding of the spatial and temporal distribution of INPs and their connection to the bulk seawater and SSML around the world. In addition to more field studies, there is a demand for laboratory studies of collected samples to further our understanding of the connection between marine INP sources and the chemical and physical properties of the emitted INPs.

In general, surface properties as well as the physical and chemical phase state of an aerosol play an important role in aerosol processes such as IN, CCN, light scattering and adsorption, and gas-particle partitioning[260]. In IN, the particle size [261,262] and morphology [263,264] have been correlated with IN activity. Specifically, heterogeneous IN at lower temperatures has been shown to be influenced by morphology [101,265–267] and organic surface coatings [268] of effloresced salt particles. However, these studies have been performed on model or simplified salt solutions. There are very few studies that attempt to correlate the IN property of marine samples with the morphology of the SSA. In studies measuring IN properties of SSA collected from a waveflume or from smaller tanks, McCluskey et al. [258] and Lee et al. [269] examined the morphology of SSA particles impacted onto substrates. The INP morphology was correlated to biomarkers such as chlorophyll concentration, and heterotrophic bacteria counts.

Detailed discussions on microfluidic phase and IN measurement instrumentation are provided in Chapter 2 and Chapter 3 and briefly restated here. A number of microfluidic ice nuclei counters have been developed very recently, using either static [73,76,270] or flow-based [74,80,271] approaches. PDMS, being permeable to water, has also led to the construction of trap-based devices where droplet dehydration and phase transitions can be observed over a long duration in a quasi-equilibrium manner to map out phase states of aerosols at different RH [14,30,32]. A recent review by Roy et al. [272] lists different microfluidic devices for droplet-based studies.

For the current work, IN spectra and chemical properties of bulk seawater and SSML are studied with three complementary sets of instrumentation to connect IN spectra with dry-particle morphology as well as biological and chemical contributions. Samples were collected during a large multi-center experiment called SeaSCAPE (Sea Spray Chemistry and Particle Evolution) at the Scripps Institute of Oceanography, La Jolla, CA[273]. A sealed 30m long channel ("waveflume") was filled with coastal Pacific Ocean seawater collected at a depth of 2m. A piston was used to generate waves that broke on simulated beach within the waveflume to perform studies on nascent SSA. Both online and offline measurements of SSA and the bulk seawater were performed using an array of physical, chemical and biological characterization equipment during SeaSCAPE (described in Sauer et al. [273]).

A custom microfluidic INP counter and phase transition device was designed based on a recent microwell freezing counter developed by Brubaker et al. [76] and specific IN spectra for a given effloresced particle morphology after drying of the droplets were first

measured. This provided a way to investigate correlations between the IN temperature of a droplet and the effloresced residual particle morphology, possibly caused by any potential differences in sample constituents. Micro-Raman spectroscopy was performed on the samples to identify chemical signatures of effloresced bulk and SSML samples. Finally, the INP population in the samples was also measured using an ice spectrometer that tests large (50 μL) aliquots of solutions, as validation for the microfluidic IN spectra results. In addition, 95°C heat treatment and peroxide digestion was applied to the SSML samples to highlight the contribution of biogenic and organic INPs respectively.

5.2 Materials and methods

5.2.1 Bulk seawater and SSML sample collection

Samples were collected at various stages of an algae bloom from the Scripps Institute of Oceanography waveflume at La Jolla, CA in the summer of 2019 during the SeaSCAPE experiment [273]. The waveflume was filled with coastal seawater from the Pacific Ocean near the Scripps Pier at the start of SeaSCAPE. The bulk seawater samples were collected using a 2m long siphon made of Teflon tubing and stored in Nalgene carboys precleaned with methanol, ethanol, 0.1 M HCl and ultrapure water. SSML samples were collected using a glass plate, and Teflon scraper. The equipment was rinsed in methanol, ethanol and 10% HCl and baked in a 500°C furnace for 5 hours to remove any organic contaminants prior to sampling. The plate was immersed into the water at 5-6 cm/s and retracted, then the scraper was used to scrape off the adhered liquid film into a collection vessel, resulting in sampling thickness of about 50 μm [274]. A subset of the

collected liquids from different days during the phytoplankton bloom was shipped frozen to different labs in 50 mL sterile centrifuge tubes for downstream analyses. Samples shown here were primarily collected on 07-26-2019, 08-02-2019 and 08-06-2019. The samples were frozen within 1 hour of collection. Figure 5-1 shows a photograph of the 30-meter-long waveflume where the samples were collected from, lit up to simulate a diel cycle.

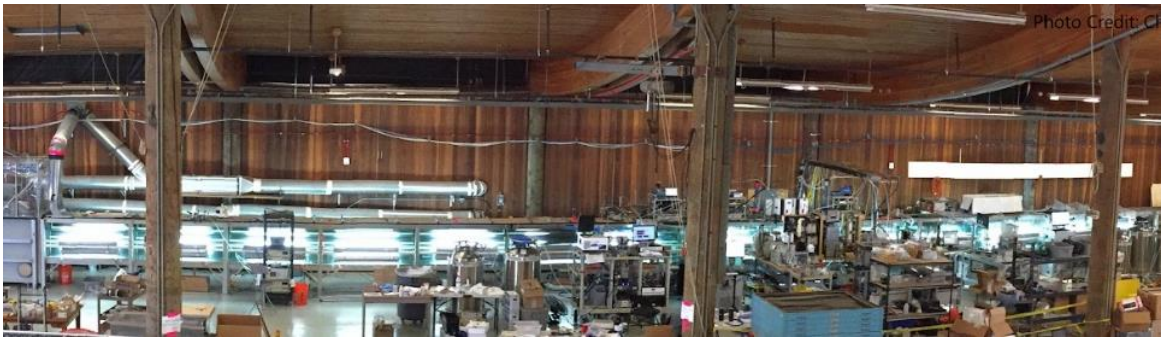


Figure 5-1: Scripps Institute of Oceanography wave flume. Bulk and SSML samples were collected from the water filling the flume during different days of a phytoplankton bloom.

5.2.2 Microfluidic device design and fabrication

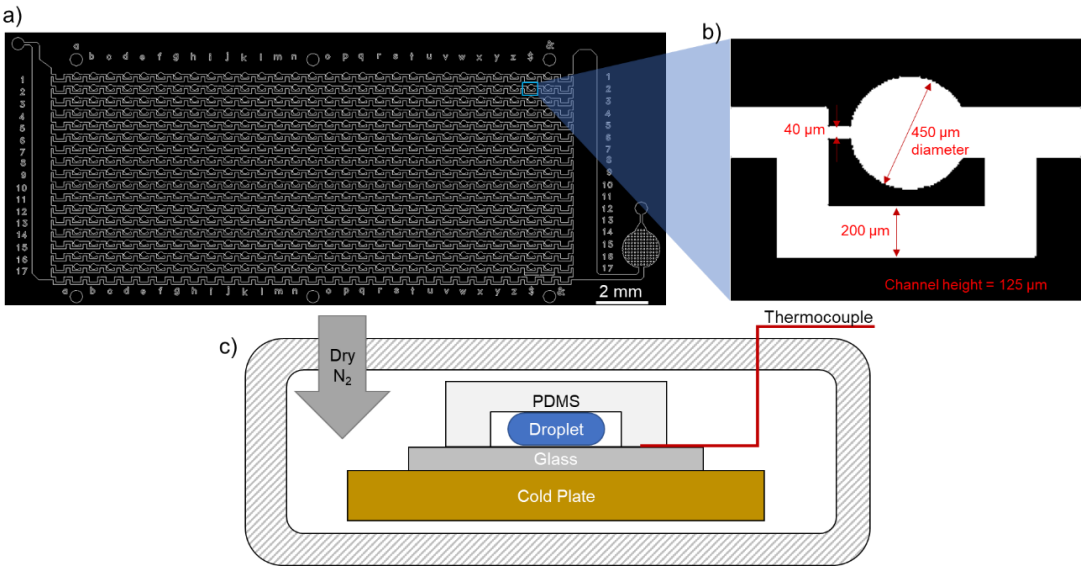


Figure 5-2: (a) Schematic of the static trap-based microfluidic phase and ice nucleation counter device, showing an array of individual wells. (b) A close up view of a single well showing the dimensions. (c) A schematic of the microfluidic device on the cold plate with dry N_2 gas purging the experimental chamber during a run.

The microfluidic device used in this study was designed based on the combination of two static well devices adapted from Sun et al. [133] and Brubaker et al. [76]. The single well shown in Figure 5-2b was replicated in rows and columns to create a 17x29 grid with 493 separate wells. The wells are connected by an inlet and an outlet to common rails as shown in Figure 5-2a. The design was prepared in AutoCAD (AutoDesk) and printed on a vinyl transparency (CAD/Art Services) to create a photomask. This photomask was used in a standard photolithography process in the Minnesota Nano Center cleanroom to create a negative mold on a 4" silicon wafer. For this process, a 150μm thick SUEX sheet (DJ Microlaminates) was hot laminated onto the wafer before exposing it to UV light using the photomask and curing the sheet before development. The microfluidic device itself was created by pouring a mixture of PDMS (Sylgard 184, silicone elastomer, Dow Corning) and the corresponding curing agent in a 10:1 ratio by weight into the silicon wafer mold. PDMS was cured overnight in a 70°C oven before the device was cut out with a razor and holes were punched for the inlet and outlet with a 1.5mm biopsy punch. Additionally, secondary channels were cut into the device to provide a place for thermocouples during the IN experiment. For the substrate, a 25x75x1 mm glass slide was coated with PDMS in a spin coater (Laurell) at 3500 rpm for 30 seconds and cured in an oven alongside the mold. The device and the substrate were plasma etched for 60 seconds before pressing them together to form a seal.

Samples were loaded into sterile disposable 1mL syringes (Norm-Ject) fitted with either a 100 nm pore-size polytetrafluoroethylene (PTFE) syringe filter (Whatman) for pure water background IN experiments or unfiltered in case of SeaSCAPE samples. Following this, silicone oil (Sigma Aldrich) was injected to scavenge out the excess sample from the channels while leaving droplets trapped in the circular wells surrounded by silicone oil. The device also allowed quasi-equilibrium dehydration or rehydration of the droplets after IN measurements due to the permeability of PDMS to water [14,30,32].

5.2.3 Temperature and RH controlled platform

A temperature-controlled cold plate (LTS 420, Linkam Scientific) was used for the experiments. A glass slide with a mirror layer of 500nm aluminum was placed between the microfluidic device and the cold plate. Thermal paste (Ceramique 2, Arctic Silver) was applied between the mirror slide and the cold plate, and a drop of silicone oil was placed between the device and the mirror slide to ensure good thermal contact while providing the best possible visuals. To provide a wide field of view, the lid supplied with the stage was removed and a custom laser-cut acrylic lid with a rubber gasket on the bottom was used to seal the device from the ambient air. A schematic of the setup is shown in Figure 5-2c. Dry N_2 gas was introduced into the experiment chamber through Teflon tubing and a port in the acrylic lid as shown in Figure 5-2c. For the IN experiments, this prevented condensation of water vapor on the outside of the device. For the phase experiment, during dehydration, this expedited water evaporation from the droplet as the droplet dehydration rate is directly influenced by the RH of the surrounding air.

5.2.4 Temperature measurement process

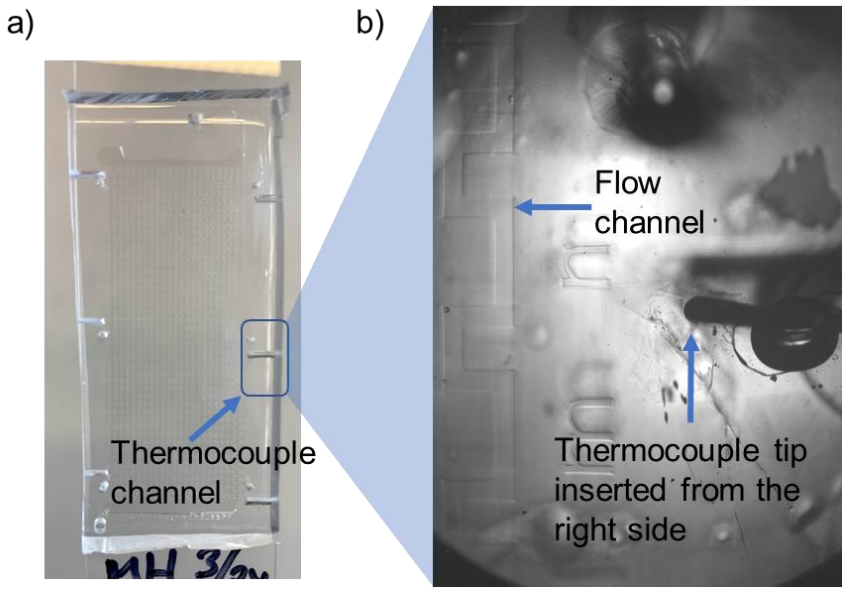


Figure 5-3: (a) Microfluidic device, showing the slits for thermocouple insertion. (b) A zoomed in view of a thermocouple inserted into the device next to the right-hand side row of flow channels from panel a.

Figure 5-3a shows the microfluidic device and the thermocouple slits surrounding the wells and flow channels. There are 6 slits in the device to map out the temperature differences across the device. However, in practice only about half of the wells are in the view of the camera during an experiment. So, only one half of the device was measured with four thermocouples. In Figure 5-3a, thermocouples were plugged in to the top four slits. The thermocouple beads were about the same size as the channels and wells as can be seen from Figure 5-3b.

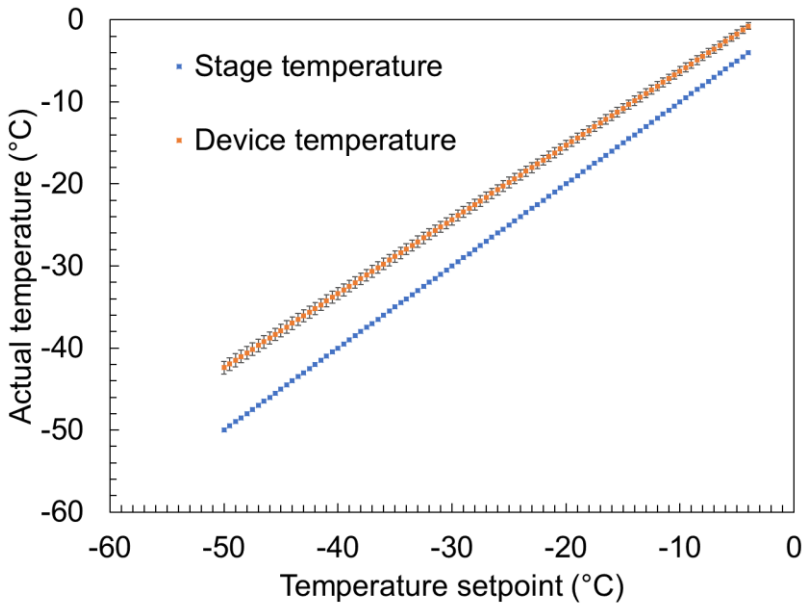


Figure 5-4: Temperature of the cold stage reported in the control software vs the temperature of the microfluidic device, measured with thermocouples.

To measure the temperature of the device, a ramp of 0.5°C/minute was performed using the Link software supplied with the LTS 420 stage. A mirror glass slide of 1mm thickness was placed on the cold plate of the stage and stuck to the stage with thermal grease (Ceramique 2, Arctic Silver). The microfluidic device was placed on top of the mirror with drop of silicone oil between the mirror and the device. This introduced some thermal resistance between the cold plate and the device, which manifested in a temperature difference between the two during the ramp. The cold plate temperature and the device temperature as measured by the thermocouples are shown in Figure 5-4. Also shown in a linear fit of the temperature of the device as a function of the stage temperature. However, this relationship is for illustrative purpose only, and was not used to measure IN temperatures since the actual thermocouples were used while measuring each sample. The

error bars on the device temperature graph indicates the maximum and minimum variation of temperature between the different thermocouples and is an indicator of the uncertainty in the final temperature measurement. Some values from this graph are given in Table 5-1. The variation from mean of ±0.69°C at -35°C stage and -28.85°C mean device temperature was used as the temperature uncertainty in all microfluidic IN figures.

Table 5-1: Temperature of the cold plate, mean thermocouple temperature and variation of the temperature at different temperatures

Stage temperature (°C)	**Mean thermocouple reading (°C)**	**Variation from mean (°C)**
-5.0	-1.72	±0.49
-10.0	-6.26	±0.54
-15.0	-10.78	±0.57
-20.0	-15.30	±0.60
-25.0	-19.81	±0.63
-30.0	-24.33	±0.66
-35.0	-28.85	±0.69
-40.0	-33.37	±0.72

The temperature of the device was also verified with the melting points of pure hydrocarbons of different molecular weights, spanning the range of working temperatures in the device. These measurements were performed by filling the device completely with these chemicals instead of using silicone oil to create droplets. This was done to avoid some of these compounds dissolving into the silicone oil. Table 5-2 shows the results of these measurements.

Table 5-2: Melting points of pure hydrocarbons measured with thermocouples in the microfluidic device

Material	Melting Point (°C)	Measured (°C)
N-dodecane	-9.65	-9.6
1-dodecyne	-19.0	-18.7
N-decane	-29.85	-29.2

5.2.5 Microfluidic experimental procedure

During a typical experiment, as explained before, pre-calibrated thermocouples (T type, 5SRTC-TT-T-36-36-ROHS, Omega Engineering) were installed inside a precut slit cut near the center of the device between the top thick PDMS layer and the thin PDMS coated substrate. Temperature measurement of the device was performed during each sample measurement. The device was placed on the mirror slide, the top acrylic cover was set in place and the dry airflow was turned on. Imaging was performed on a reflective microscope (SZX10, Olympus) at different magnification levels and images/videos were recorded with a 1200x1600px monochromatic camera (acA1600-60gm, Basler)

A flowchart of a typical experiment is shown in Figure 5-5. IN measurement was done immediately after loading the device with the sample droplets. The microscope was zoomed out to the widest view possible, at which point, ~185 droplets were in view simultaneously. Droplets were held at 0°C for 10 minutes for equilibration and then the temperature of the cold plate was lowered at 0.5°C/min using the supplied Link software and device thermocouple temperatures were monitored. Frozen droplets turned darker than

liquid droplets. A video was recorded during the ramp to help track the number of droplets frozen at each temperature.

Following this step, the temperature was brought back to 20°C rapidly and held there for the dehydration experiment. The camera was zoomed in to keep a 3x3 grid of droplets in view. A time lapse video was recorded as the dehydration process happened in a quasi-equilibrium manner over the course of a day. Finally, the camera was zoomed in to single wells and all the particles in the device that were observed for the IN experiment were scanned one at a time under the microscope to categorize each into separate morphologies. The bulk and SSML samples from 07-26-2019, 08-02-2019 and 08-06-2019 were measured using the microfluidic device. However, there were some contamination issues with the 08-02 and 08-06 samples as discussed in section 5.3.5.

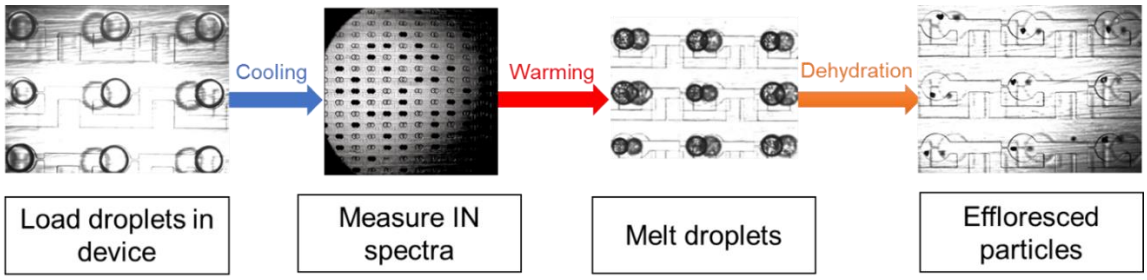


Figure 5-5: Flowchart of experimental steps performed. The cooling and warming process takes about ~1 hour while the dehydration process takes >24 hours.

For the IN experiment, frozen fraction of droplets $f(T)$ as a function of temperature T was recorded from the captured video and calculated as given by Equation 4-1 and reproduced here:

$$f(T) = \frac{N_f(T)}{N_{total}} \qquad \text{Equation 5-1}$$

$N_f(T)$ is the number of wells that are frozen at T and N_{total} is the total number of wells in the view of the camera. The INP concentration in the sample can be calculated using the volume concentration correlation put forth by Vali [234]:

$$INP(mL^{-1}) = \frac{-\ln(1 - f(T))}{V(mL)} \qquad \text{Equation 5-2}$$

here, V refers to the volume of the droplet at the time of observation and was calculated with the following pancake shape assumption [32], given previously in Equation 2-2 and reproduced here:

$$V = \frac{\pi h^3}{6} + \frac{\pi h}{4}(D - h)\left(\frac{\pi h}{2} + D - h\right) \qquad \text{Equation 5-3}$$

Where, D is the observed diameter of the droplet and h is the height of the microfluidic channels (150 μm). Droplets that filled the wells completely had a volume of 21nL. Only droplets within 90% of this maximum were considered for the IN spectra. A freezing point depression correction of +2°C was applied to all spectra.

5.2.6 Environmental cell micro-Raman spectroscopy

Frozen bulk and SSML samples corresponding to the identical days analyzed with the IS were melted in a warm water bath at 30°C. Following this, approximately 1μL droplets of aqueous solutions of the SSML and the bulk sea water were pipetted onto a hydrophobic substrate. The substrate consists of a quartz disc (Ted Pella, 16001-01) coated

with Rain-X. These hydrophobic discs containing the microdroplets were then placed into an environmental cell (Linkam, LTS 120) with temperature control. The cell is coupled to a micro-Raman spectrometer (Horiba, LabRAM HR Evolution) for spectral analysis. The spectrometer is fitted to an optical reflective microscope (Olympus BX41), a 100X super long working distance objective and a 532 nm laser. The environmental cell is fitted with a gas inlet and an outlet which in turn in connected to a CR-4 hygrometer (Buck instruments) which allows for RH control through a feedback loop. The RH control is done by sending N_2 through a bubbler of milli-Q water and modulating the ratio of wet to dry N_2 flowing into the environmental cell. Once in the environmental cell, the droplets were dried for 12-48 hours under a flow of N_2 at 0.05 LMP at 25°C prior to data collection, and then held under the same conditions throughout the experiment. Once dried, Raman spectra of different particles of varying morphologies (identified with the Olympus optical microscope fitted to the confocal Raman system) were collected using 10X objective with 3 exposures of 3 seconds averaged for each scan, from 400 to 4,000 cm^{-1}.

5.2.7 Ice spectrometer, heat and peroxide treatment

INP temperature spectra were additionally obtained using Colorado State University's Ice Spectrometer (IS). The IS measures immersion freezing INPs in aliquots of water arrayed in four 96-well polymerase chain reaction (PCR) plates. These are fitted into aluminum blocks with machined wells and encased by cold plates through which coolant is circulated. The IS produces immersion freezing spectra reaching to -27°C, with a detection limit of 0.6 INPs mL^{-1} and is supported by optimized experimental protocols [169,175,275,276].

To measure INPs in each sample, and in each serial dilution (several 11-fold dilutions made in 0.1 μm-pore-filtered deionized water), 32 aliquots of 50 μL were dispensed into PCR trays (OPTIMUM ULTRA, Life Science Products) in a laminar flow hood and placed into the cooling blocks. These and a headspace N_2 purge flow were cooled at 0.33 °C/min using a recirculating low temperature bath, and the freezing of wells recorded. Freezing fraction results were corrected for INPs in the deionized water used for dilutions, and INP temperature spectra were obtained by converting the number of frozen wells at each temperature to the number of INPs mL^{-1} using Eq. 13 in Vali (1971) [234]. Freezing temperatures of undiluted bulk seawater and SSML were adjusted by +2°C to correct for freezing point depression. Ninety-five percent confidence intervals were obtained from Eq. 2 in Agresti and Coull (1998) [277].

Tests were performed to estimate heat-labile (e.g., proteins) and total organic INPs. For the former, 2 mL of SSML were heated to 95°C for 20 min and the sample re-analyzed to measure the reduction in INPs. To quantify organic INPs, 1 mL of 30% H_2O_2 (Sigma) was mixed with 2 mL of sample and digested at 95°C for 20 min under UV-B. This and the decomposition of residual H_2O_2 are detailed in Suski et al. (2018)[210]. The sample was then re-analyzed in the IS to measure the reduction after the removal of organic INPs.

5.3 Results

5.3.1 Microfluidic IN spectra

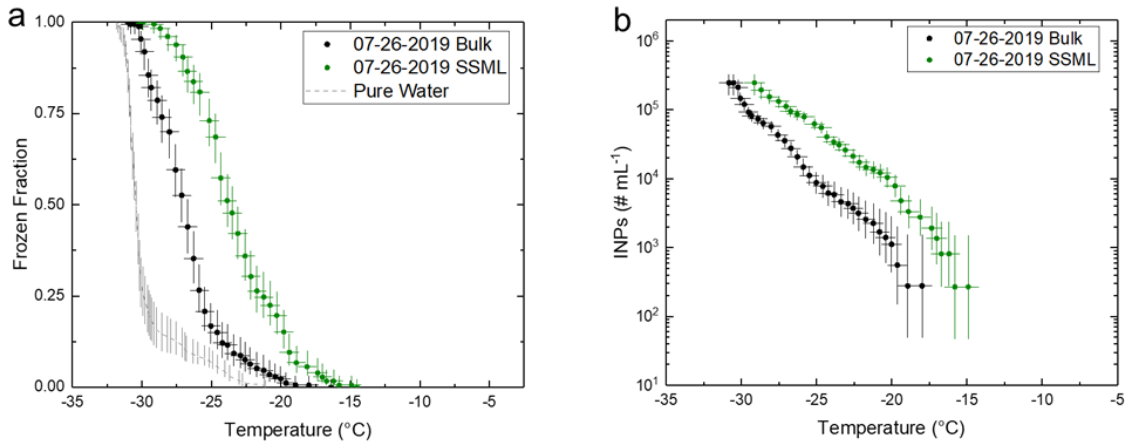


Figure 5-6: IN spectra measured using the microfluidic device from bulk and SSML of 07-26-2019. (a) Frozen fraction as a function of temperature, calculated from the fraction of wells frozen out of all wells in the view of the camera at any given temperature. The gray line represents the background pure water spectra. The background was collected after loading the device with a syringe of deionized water fitted with a 0.1um PTFE syringe filter. (b) The bulk and SSML samples were loaded with syringes without any filters attached. Vertical error bars indicate 95% confidence interval. Horizontal error bars indicate uncertainty of 0.69°C in temperature measurement from thermocouples.

IN data of the bulk and SSML from 07-26-2019 collected with the microfluidic device are shown in Figure 5-6. The horizontal error bars represent uncertainty in temperature measurement from the thermocouples. The pure water ice nucleation background spectra is plotted in Figure 5-6a and shows a freezing curve spanning around 10°C (between -21 and -31°C) with a steep rise below -30°C. The median freezing temperature for ultrapure water in this device is -30.3°C. This curve has a similar shape and freezing temperature range to other droplet based IN counters [73,76,278]. While the wide range of freezing observed at temperatures warmer than -30°C indicates heterogeneous

nucleation, either from contaminants in the water or from contact with particles and/or the surfaces already present inside the device, this background freezing is still sufficiently less IN active than the sample spectra.

A prominent feature of these graphs is that the gap in INP concentrations (Figure 5-6b) between the bulk and the SSML is about an order of magnitude at temperatures >-26°C. This enrichment in INPs in the SSML may also indicate a compositional difference, and to explore this, the droplets were dried after the IN spectra measurement to investigate any overall trends in the efflorescent particle morphologies. The microfluidic device allows droplet dehydration directly after IN spectra measurement, without having to transport the droplets to a new device. The connection between the freezing temperatures of the individual droplets and the effloresced particle morphologies was investigated as explained in the following section.

5.3.2 Effloresced particle morphology

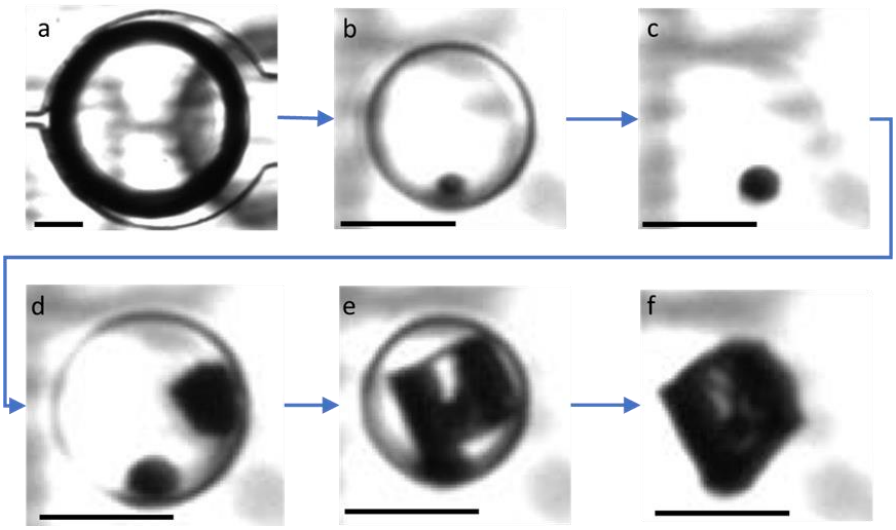


Figure 5-7: (a-f) Droplet morphologies observed in a bulk seawater droplet from 07-26-2019 dehydrating inside a microfluidic well. The arrow indicates the progression of the experiment. A particle nucleates in panel b and its size remains relatively constant throughout the rest of the dehydration process. The refractive index of the droplet matches the surrounding silicone oil in panel c. A new salt crystal, possibly NaCl, nucleates in panel d and keeps growing into a larger cubic crystal until panel f where a translucent coating forms over the whole dry particle. Scale bars in the images represents 100μm.

The droplet dehydration process is shown in Figure 5-7. Note that the reflection of the droplet from the mirror surface below can be seen as a faint image in the background of Figure 5-7(a-f). The fresh, or just-loaded, droplet in Figure 5-7a is relatively large and slowly dehydrates until it reaches the state shown in Figure 5-7b. At this point, a small darker particle emerges from the liquid, and the whole droplet keeps shrinking as it evaporates until it reaches the state in Figure 5-7c. Here, the droplet disappears from view due to the refractive index of the liquid in the droplet becoming equal to the refractive index of the surrounding oil. Following this, the droplet continues to shrink and reappears into view as a second particle emerges from a different spot in Figure 5-7d. The second particle grows and forms a transparent cubic shape as seen in Figure 5-7e. This is an NaCl crystal and has a cubic crystalline structure. Near the very end of the dehydration process, a translucent coating appears to form and cover the whole dry particle as shown in Figure 5-7f. The main part of the dehydration process from Figure 5-7a to b takes about 24 hours, and the remaining part from b to f takes around an hour.

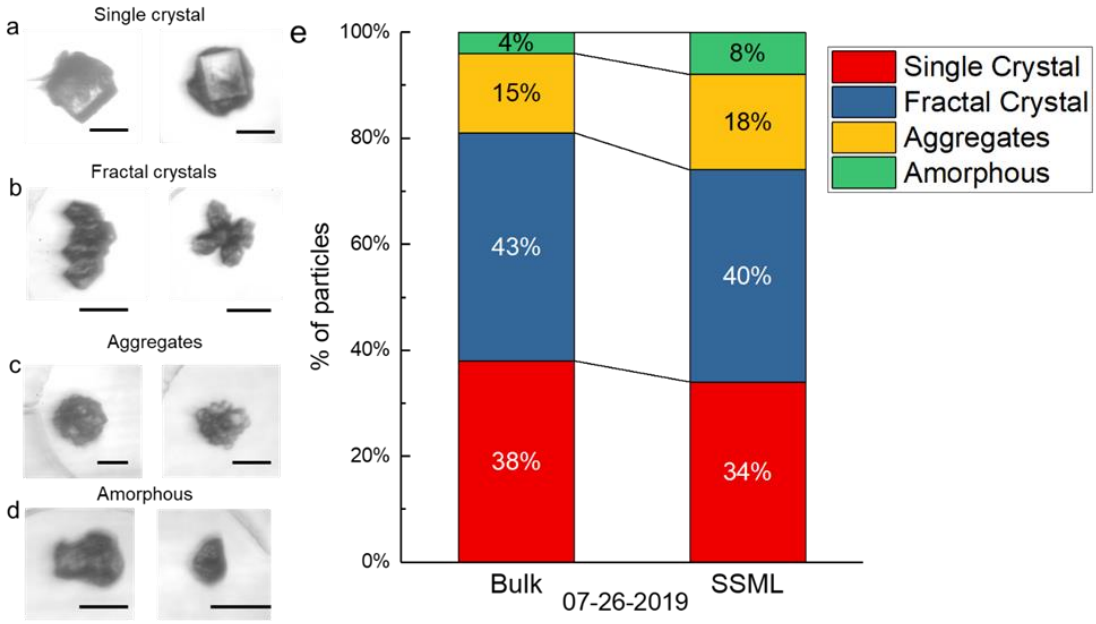


Figure 5-8: (a-d) Typical effloresced droplet morphologies observed in microfluidic wells after complete droplet dehydration. Scale bars in the images represent 50 μm. (e) Droplet morphologies represented as a fraction of total droplet count observed in each sample. The samples contained 173 and 178 total droplets respectively.

Figure 5-8(a-d) shows the types of particle morphologies observed after an efflorescence experiment. The single crystal morphology has a single large cubic crystal of NaCl surrounded by a more amorphous coating or other much smaller crystals, as evident from the transitions shown in Figure 5-7. The fractal crystal morphology has multiple medium sized crystals growing in different orientations in a jagged geometric repeating pattern, possibly indicating nucleation from multiple sources through smaller particles present in the droplets. The aggregate particle morphology has almost a circular shape with many small crystals, forming the outer surface of the particle. The amorphous morphology is assigned to particles that have no discernible crystalline shape on the

outside, which possibly indicates a larger amount of detritus or organics in the initial droplet. These form a thicker or more irregular coating on the outside of salt crystals inside the particle, thus concealing them. The distribution of different particle types formed after efflorescence is shown in Figure 5-8e. The bulk sample has a slightly larger fraction of single and fractal crystal particle types when compared to the aggregate and amorphous particles.

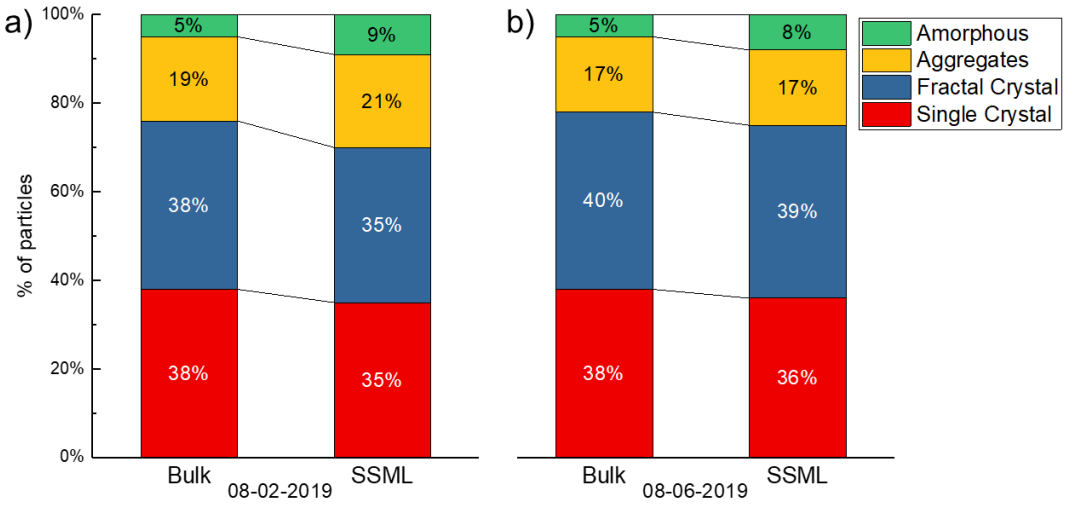


Figure 5-9: Effloresced particle morphology data collected in the microfluidic device (a) during chlorophyll peak during a bloom and (b) post the chlorophyll peak.

Figure 5-9a,b shows particle morphology data of bulk and SSML collected for two additional days during the phytoplankton bloom. Out of these, 08-02-2019 was during and 08-06-2019 was after a peak in the chlorophyll count during the bloom. Both these datasets show a similar trend to the samples reported in Figure 5-8, i.e. the SSML sample has higher amounts of aggregates and amorphous particles than the bulk samples regardless of

collection date. In samples from all days, SSML contains more aggregate and amorphous samples, while the bulk contains more crystal structures.

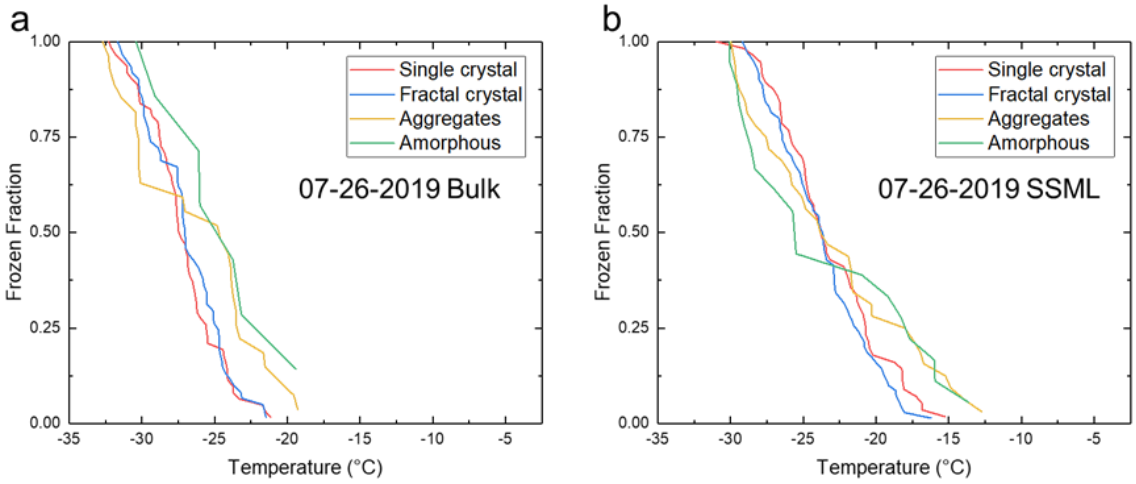


Figure 5-10: Frozen fraction curves of each droplet morphology for a given sample. a) 07-26-2019 bulk, b) 07-26-2019 SSML.

The freezing temperatures of the droplets in each of the wells were connected to the effloresced particle morphology and the resulting frozen fraction curves for each particle type are plotted in Figure 5-10a,b. From the subpanels it is apparent that there is no significant difference in the IN temperatures of the single crystal vs. the fractal crystal types in both bulk and SSML. However, taking a closer look at the aggregates and the amorphous category curves reveals that these droplets start freezing at a warmer temperature than the single or fractal crystal droplets across the two samples. They also generally have different freezing curves with more gradual slopes, leading to freezing over a broader temperature range compared to the single and fractal crystal particles. This perhaps indicates that the liquid droplets in these wells contained a more heterogeneous

mix of contents contributing a diverse freezing range of INPs, requiring further chemical identification as described in the next section.

5.3.3 Chemical signatures of particles

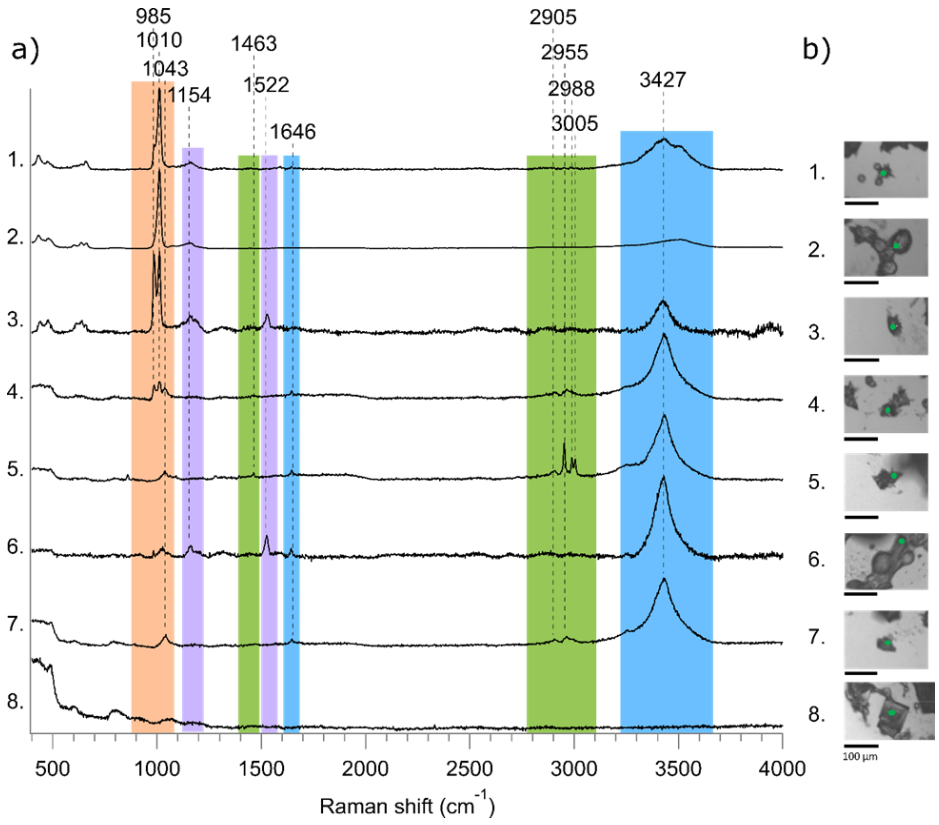


Figure 5-11: Results from representative (a) spectra and corresponding (b) optical images from dried SSML and bulk single particles. Different color bands highlight the significant spectral regions and peaks that correspond with literature values of sulfate (orange), carotenoids (purple), other organic compounds (green), possibly sialic acid (5), and water (blue). Green dots on the optical images indicate the location spectra were collected[1].

[1] Data collected by Liora Mael from the Grassian research group, University of California San Diego

Representative Raman spectral and optical data for dried droplets from SSML and bulk seawater were collected. This was done for samples collected throughout the phytoplankton bloom (07-26-19, 08-02-19, and 08-06-19). These data are shown in Figure 5-11. All of the representative micro-Raman spectra shown in Figure 5-11 were found throughout the bloom in both the bulk and SSML samples. Many particles (spectra 1-7) contain sulfate. These are identified by the sulfate symmetric stretch ranging from 950 to 1100 cm^{-1} (highlighted in orange). The exact frequency of this vibrational mode depends on several factors including the coordinating cation (Na^{+}, Ca^{2+} and Mg^{2+}) and hydration state[279,280]. In addition to sulfate, particles corresponding to spectra 3 and 5 showed the presence of carotenoids as indicated by the two peaks at 1522 and 1154 cm^{-1} (highlighted in purple)[281]. Several particles exhibit vibrational peaks indicative of organic compounds (shown in spectra labeled 4, 5, 7) in addition to sulfate (highlighted in green). The amount of organics present in the spectrum labeled "5" is much higher than in other spectra. This has been identified as sialic acid as discussed in previous studies [280,282]. Many particles found across the bloom were identified as sodium chloride (spectrum 8). These particles were identified because they have no Raman bands in the region investigated, and in many cases, clear salt crystal habits were observed in the optical image. All particle spectra identified (1-8) most likely contain various amounts of NaCl, but due to the lack of spectral signature this cannot be quantified.

Table 5-3: Raman spectra assignment for the bulk and SSML samples

Wavenumber (cm^{-1})	**Peak assignment**	**Reference**

985, 1010, 1043	*Sulfate* n(S-O)	279,280
1154, 1522	*Carotenoids* n(C-C) n(C=C):	281
1463 2905, 2955, 2988, 3005	*Organics* δ(C-H) ν(C-H)	280,282
1646, 3427	Water δ(O-H) ν(O-H):	280

5.3.4 Heat treatment and H_2O_2 oxidation results

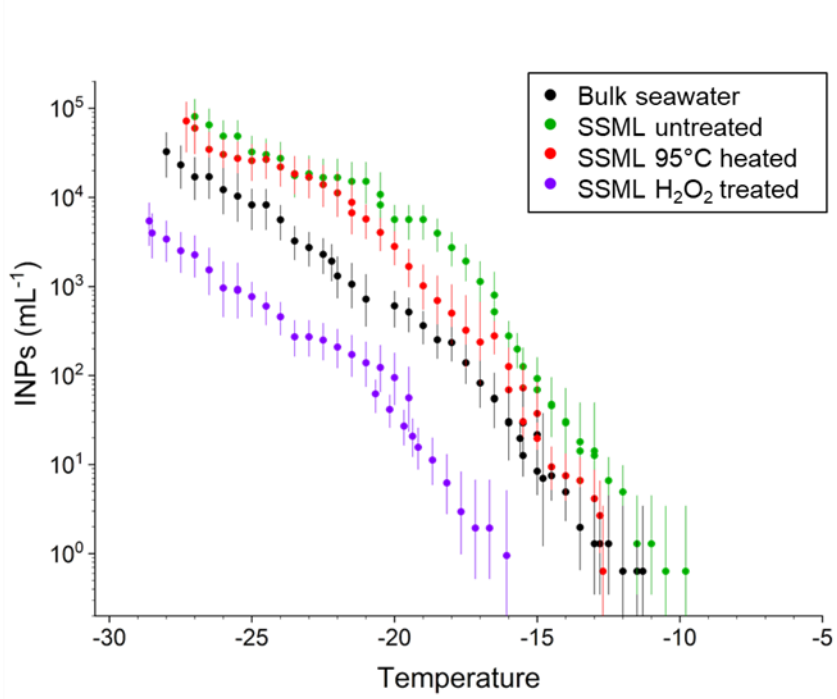


Figure 5-12: INP counts in SSML and bulk seawater for nominal, 95°C heat-treated and H_2O_2 digested sample from 07-26-2019. The data was collected with the automated ice spectrometer

using a 96 well plate. The curves have been corrected for freezing point depression. The error bars represent 95% confidence intervals[2].

Figure 5-12 shows the INP counts in bulk and SSML samples from 07-26-2019 measured with the IS using 50 μL aliquots. In the regions of overlap between the microfluidic and IS spectra (from around -15 to -28°C) there was very close agreement in both INP concentrations and spectral curvature for both bulk and SSML samples. As seen with the microfluidics measurements, there is a significant enrichment of INPs in the SSML. Also shown are the INP counts after treatment of the SSML samples by heating at 95°C and after peroxide digestion. Specifically, INPs that were active above -22°C in the SSML were moderately heat sensitive, while >98% of INPs at all temperatures were eliminated by H_2O_2 oxidation (i.e, the population was almost entirely organic).

[2] Data collected by Tom Hill from the Demott research group, Colorado State University

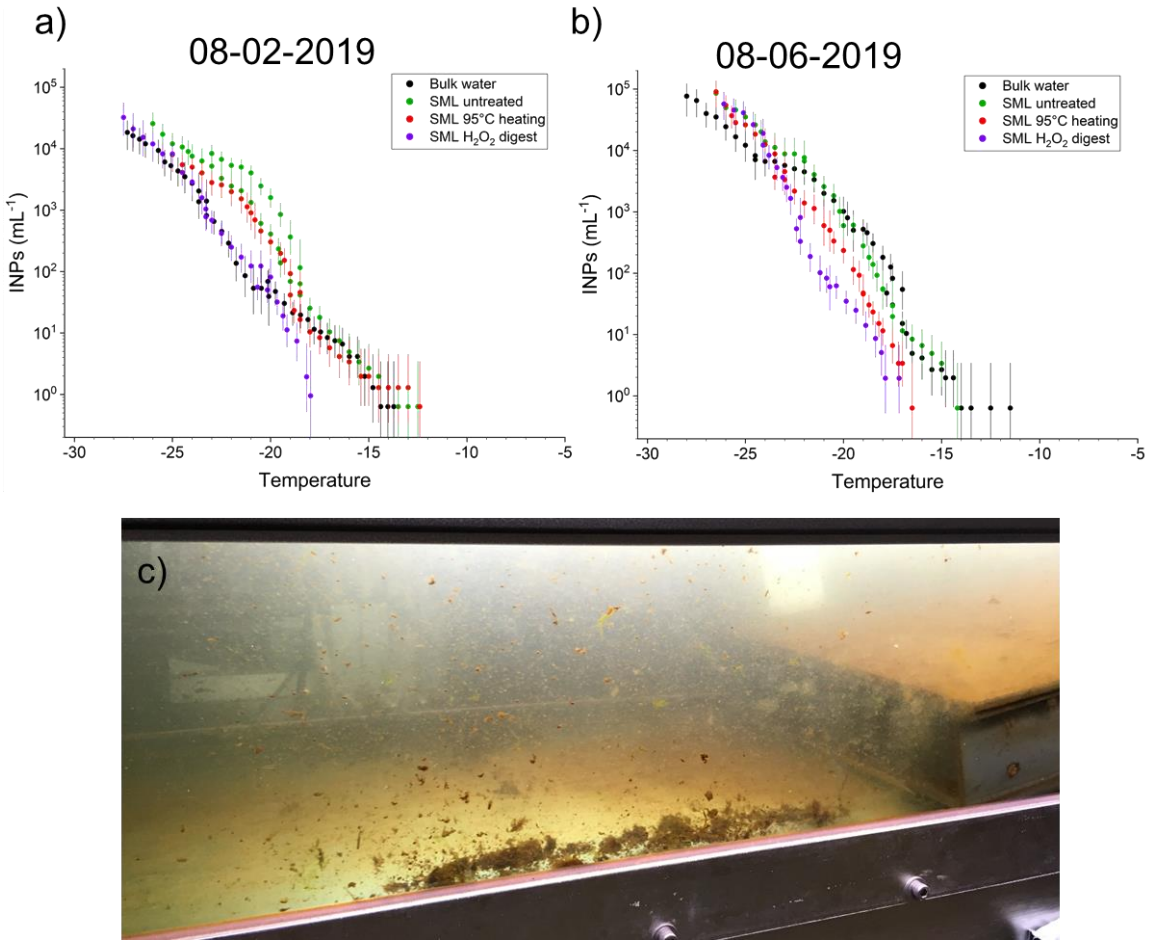


Figure 5-13: (a, b) INP counts in SSML and bulk seawater for nominal, 95°C heat-treated and H_2O_2 digested samples. Sample days are indicated on the graphs. The curves have been corrected for freezing point depression. The error bars represent 95% confidence intervals. (c) Waveflume at 11 am on 08-06-2019, after cleaning of inner walls with wipers. This picture was taken before collecting the sample of 08-06-2019. The sample from 08-02-2019 was collected at peak-bloom after addition of nutrients; and 08-06-2019 was collected at post-bloom as indicated by the peak of the chlorophyll count of sampled water[3].

[3] Data collected by Tom Hill from the Demott research group, Colorado State University

Shown in Figure 5-13a,b are IS data from two additional days. In the 08-02-2019 SSML sample, collected during increased chlorophyll concentration in the water during a bloom, the concentration of INPs in both the SSML and bulk water had significantly decreased compared with the sample taken seven days previously, especially for those activating above -25°C. Because the reduction was more pronounced in the SSML, the surface enrichment of INPs also declined, with none evident for INPs active >-19°C and only a minor level of enrichment below -24°C. The SSML INPs were also now heat insensitive. The effect of H_2O_2 digestion varied by temperature range and was generally was much less pronounced. Compared with 07-26-2019, the concentration of inorganic INPs in the SSML was the same above -22°C, but had increased for those active at colder temperatures, matching that of the underlying water, and perhaps indicating that the SSML was now so diminished that it was more akin to the bulk water in its composition. In this sample, we also observed an unusual effect of dilution upon IN activity: INPs in samples diluted 121-fold in DI water were more active (upper series of green dots spanning from -18 to -23°C) than those in either the 11- and 1331-fold dilutions, which corresponded with each other. The test was repeated and identical results were obtained. The sample from 08-06-2019, taken after biofilms on the glass walls and sediments on the base had been dislodged using scrapers the previous day (Figure 5-13c), simulates the resuspension of material in coastal waters following a storm. Hence, there was an increase in bulk water INP concentrations in this sample compared with 08-02-2019. Interestingly, this pulsed release of biological material into the water column did not lead to a pronounced INP enrichment in the SSML. The INPs were, however, heat sensitive and, above -23°C,

organic. Given the lack of SSML enrichment, the biological and organic characteristics of this sample may, like with the previous sample, reflect those of the INPs in the bulk water.

5.3.5 Contamination of samples through use of a filter

Droplet loading in the microfluidic device must be done exceptionally carefully so as to not introduce any extra sources of contamination in the sample. An instance of contamination was observed in the initial IN measurements of the SeaSCAPE samples. The following protocol was the culprit: frozen samples in 50 mL centrifuge tubes were thawed and loaded onto a sterile disposable 3mL syringe. Then a 10 μm syringe filter (Kinesis, Cole Parmer) was used to filter out any large particulate matter that might be present in the sample. Ice spectra recorded with this loading protocol are shown in Figure 5-14.

Some of the trends observed in Figure 5-7 can be seen here as well. For instance, the order of magnitude difference in bulk and SSML INPs, reflected in both the frozen fraction as well as the INP concentration figures are consistent between the data. Also, when compared with Figure 5-6, the actual INP count data lines up with the IS measurements below -20°C in all samples. However, there is a specific trend of about 10% of the wells freezing before -15°C in all the samples. This translates to more than 10^3 INPs above -15°C. This early freezing behavior stops and picks up again around -20°C. This "two region" freezing behavior indicates the presence of highly efficient biological INPs in the samples such as bacteria. But also, from Figure 5-6, no such INP population existed in the samples. And indeed, it would be quite impossible for all the samples to show similar

kind of biological INP presence given that the samples were collected during different days of a phytoplankton bloom.

It was suspected that the microfluidic samples were contaminated during handling, possibly due to the additional filtering with the 10 μm filter. To test this hypothesis, samples of synthetic seawater (SSW) and ultrapure water was run in the device, with and without the 10 μm filter being used during loading. Results are presented in Figure 5-15.

When either frozen fraction or the INP concentration count of the SSW sample with and without the filter is compared, it becomes apparent that there was some contamination that was present in the filter that made its way into the droplets inside the microfluidic device. Curiously, the same behavior was not observed with ultrapure water. i.e. the filtering step did not introduce early freezing behavior in the ultrapure water sample. This puzzling behavior can perhaps be attributed to the differences in the ability of the SSW and the fresh water to dislodge the biomaterial contaminant from the filter membrane. Salt is well known to induce osmotic pressure related injury to biomaterials[283]. It is possible that the salt in SSW destroys and dislodges the IN active bacteria or biomaterial much more effectively than is possible by the ultrapure water. A more detailed study is necessary to quantify this specific behavior for future filter usage.

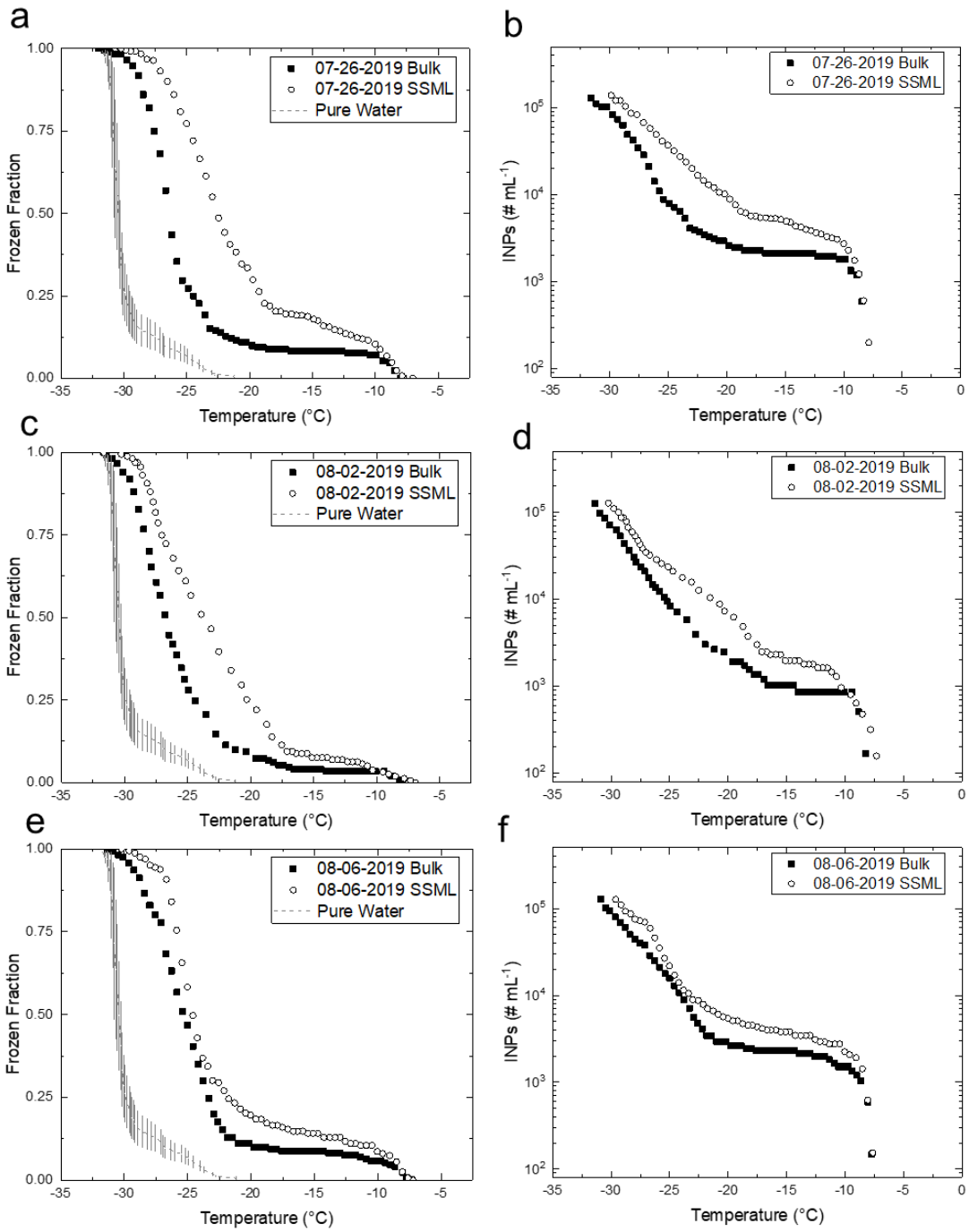


Figure 5-14: Contaminated ice spectra of SeaSCAPE samples recorded after loading the device with a 10 micron filter. (a,c,e) Frozen fraction data of three different days. (b,d,f) INP per mL sample from the three days. The collection dates are indicated in each legend.

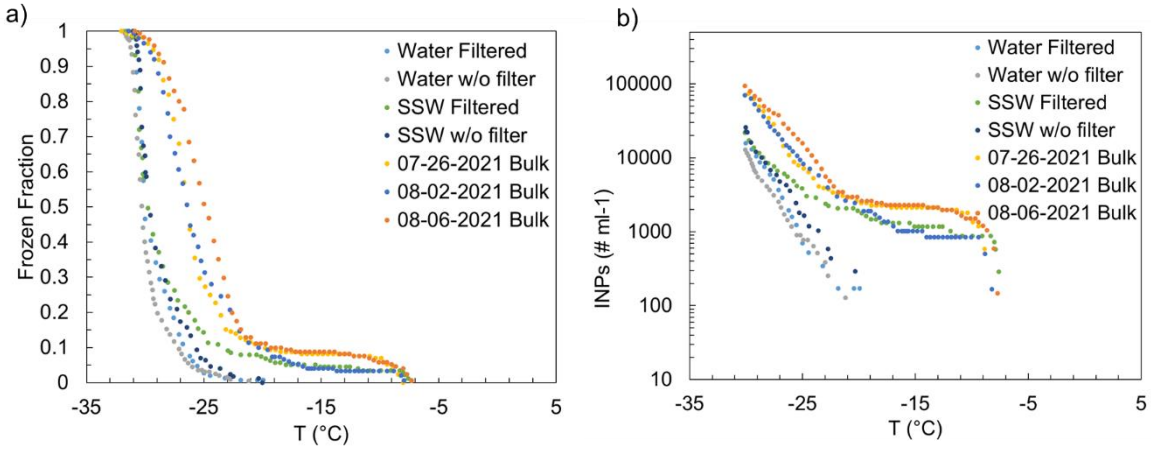


Figure 5-15: Results comparing contaminated samples. (a) Frozen fraction of all SeaSCAPE samples and SSW and pure water. The SSW and pure water samples were used both with and without the 10 μm filter. (b) The corresponding INP counts per mL of sample.

5.4 Discussion

Previous studies on the ice nucleating properties of SSML and bulk seawater on field or lab-generated samples show that there is usually large variability in the ice nucleation temperature even in a single study [246,248,255–257,284]. The variations arise from different conditions at the sampling locations such as the water chemistry, local biological activity, SSML composition, and the sampling method employed. In particular, results in SSA and seawater (both bulk and SSML) collected from waveflumes[257,258] have shown significant variability in the IN properties before, during and after a phytoplankton bloom. The bulk and SSML INP counts reported with both the IS and the microfluidic device reported here are within the range reported in previous studies on bulk seawater [246,248,250,251,255,256,258] as well as SSML [247,255–258,284].

The microfluidic ice nucleation results show clear differences in the INP concentration between the bulk and SSML samples, indicating significant enrichment compared to the bulk. Similar results have been reported previously [284]. Marine biological entities such as heterotrophic bacteria [285], phytoplankton [254,286], surface active organics [247] and gels are enriched in the SSML.

The morphologies of effloresced particles observed in the microfluidic wells are broadly equivalent to the particle types observed in previous studies on SSA impacted onto substrates and imaged with atomic force microscopy [269] or scanning electron microscopy [258], i.e. crystalline, aggregates and amorphous entities. Due to the high salt content of sea water, the microfluidic results show aggregate and amorphous morphology particle counts that are much smaller in number compared to the single and fractal crystal morphologies, the latter making up 81% and 74% of the particles in the bulk and SSML samples, respectively. IN temperatures of the aggregate and amorphous morphologies are spread over a wider range of temperatures than the single and fractal crystals. Similarly, both the amorphous and aggregates morphologies have a several degree warmer onset temperatures of freezing than the other types.

The differences in morphology signify an excess amount of organic material present in the droplets that effloresced into aggregate and amorphous particles, perhaps related to an inhomogeneity present in the specific well or departure from the overall average composition of the sample as a whole. This may indicate the presence of larger gels. The single and fractal crystal particle freezing curves in both bulk and SSML samples have similar slopes and overall nucleate at colder temperatures and with a smaller spread,

indicating more homogeneous and perhaps inorganic salt contents in the droplets. Further investigation of the chemical composition was determined with environmental micro-Raman spectroscopy on effloresced residual particles formed from microliter droplets. The Raman spectra show NaCl, sulfate, carotenoids, and other organic compounds in particles from both the bulk and the SSML collected from the same day as those used in the microfluidic experiments. These samples also included residual particles containing some unique signatures of sulfate hydrate salts with organics with peaks in the region 2905-3005 cm^{-1}. Some of these particles included sialic acid which was not present in the bulk. Sialic acid has been identified as an algae synthesized product [287], and hence its presence in the samples is not surprising for these seawater-based samples.

Finally, heat and peroxide treatments further differentiated the heat-labile (i.e., biological) and total organic components of the INP populations. There was a moderate reduction of INPs after heat treatment but a further very large decrease after peroxide digestion. This indicated that the INP population in the SSML consisted a mix of heat-labile biological entities and heat-stable organic INPs (only 0.5-2% of INPs were inorganic). Interestingly, the peroxide treatment also reduced the INP count below that of the bulk seawater, suggesting that the bulk also contained a significant amount of organic INPs. This generally agrees with previously observed trends of larger amounts of biological and organic INPs present in SSML samples compared to bulk samples [246,247,258], which also deactivated with heat [246–248,258]. The sample analyzed here (07-26-2019) is from relatively fresh seawater, sampled three days after the filling of the waveflume. A significant heat sensitive INP population in the was found in the SSML at >-20°C, which is also the

temperature range where the amorphous and aggregate residual particles start freezing earlier than the single and fractal crystal particles in the microfluidic device. This result is indicative of higher amounts of the warmer INPs in the former morphologies.

5.5 Conclusions

In this chapter, the IN characteristics were connected to particle morphology and composition of bulk seawater and SSML from the SeaSCAPE campaign, using a microfluidic method complemented by ice spectrometry and micro-Raman spectroscopy measurements.

For the microfluidic method, a grid of microfluidic wells was used in a PDMS-based device to create nanoliter-sized droplets and study their ice nucleation temperature. The grid allowed the correlation of the temperature at which ice nucleated with the morphology of the particle. Upon drying, bulk and SSML samples from the same day showed broadly four different morphologies: single crystal, fractal crystal, aggregates and amorphous. The single crystal and fractal crystal types were present in a larger fraction of the particles in the bulk sample compared to the SSML sample. When the ice spectra of the samples were analyzed according to particle type, it was found that droplets which effloresced into aggregate and amorphous particles started freezing at a warmer temperature compared to the other two types. This is an indication perhaps of some unique organic material or contaminants present in the droplets or some other sources of inhomogeneity in the drops forming the aggregate and amorphous particles compared to

the largest fraction of drops, which effloresced into the single and fractal crystal morphologies.

Micro-Raman spectroscopy of dried microliter droplets of SSML and bulk seawater showed the presence of sulfates, organic compounds identified in some cases as carotenoids and sialic acid, and sodium chloride. Pure sodium chloride crystals were identified as well. These different chemical signatures were present in the bulk and SSML. Ice nucleation was also measured with a 96 well plate-based ice spectrometer for both bulk and SSML. The nominal INP counts had good correspondence with the microfluidic IN results, serving as a validation of the microfluidic method. The SSML sample was additionally measured after heat treatment at 95°C and H_2O_2 digestion to isolate any heat-labile biological INPs and organic matter respectively. The samples showed vastly different amounts of enrichment of heat sensitive and heat insensitive biological and organic species present in the bulk and SSML. This indicates the immense diversity and spatio-temporal variability in the INP concentration present in the samples.

This study, for the first time, utilizes a microfluidic device to investigate both phase transitions and ice nucleation characteristics of aqueous droplets, and provides a platform to correlate effloresced particle morphology to droplet ice nucleation temperature. The microfluidic device results were complemented with ice spectrometer results, with heat treatment and oxidation of the bulk and SSML samples, to isolate contribution of heat-labile and insensitive organic INPs to the samples. Environmental cell Raman microscopy was performed to identify the chemical makeup of the effloresced particles. In future studies, the microfluidic device can be modified with only minor changes to integrate both

heating and Raman spectroscopy in a combined single setup. This will increase ease of use and reduce variability among samples and measurement methods. The microfluidic platform shown here has the potential to perform these studies with an order of magnitude reduction in sample volume and experimental time compared to microliter sized droplet-based methods. Ultimately, it can contribute significantly to the field of atmospheric chemistry by providing a simpler and cost-effective alternative to traditional aerosol instrumentation.

Chapter 6

Future directions and conclusions

6.1 Future directions

6.1.1 Droplet efflorescence and deliquescence study with ice nucleation

Real-world samples can be made up of many different components and do not always have a single efflorescence relative humidity i.e. not all components solidify at a single point in time during dehydration, if at all. This effect was seen in the marine samples in Chapter 5. Efflorescence typical implies that a solute is going into supersaturation and hence can happen at different water activity depending on the amount of supersaturation and the presence of other solutes. This provides an opportunity to perform an efflorescence-deliquescence cycle on droplets comprised of many solutes and use the order of appearance of these transitions as a kind of indicator of the solid particle's chemical identity. In addition to identifying compounds, being able to measure the ERH and DRH of complex solutions in our device is useful for more fundamental studies where parameters for thermodynamic models are calculated.

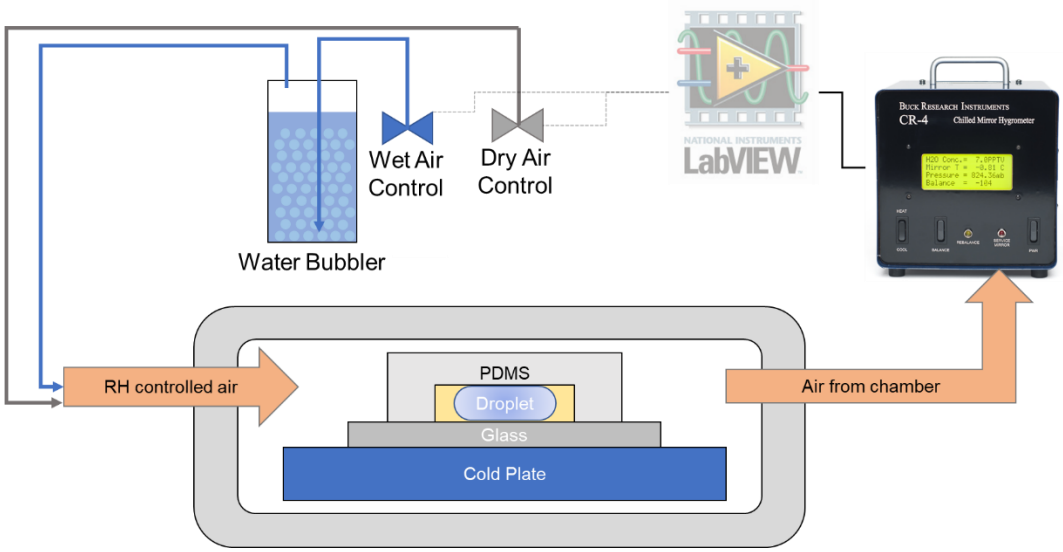


Figure 6-1: Initial stages of relative humidity control. Currently the dry and wet air lines are operated manually, and the relative humidity is read using the hygrometer. Future plan includes making a LabVIEW code for feedback control of the RH inside the chamber by automating the wet and dry air mass flow controllers.

Building of a relative humidity control setup was started with the help of colleague Margaret House. A preliminary study was performed by manually using dry air for efflorescence, followed by wet air flow by flowing air through a water bubbler for deliquescence. The schematic of the experimental setup is shown in Figure 6-1. A preliminary study was performed with this setup on a SeaSCAPE SSML sample (details on sample collection given in Chapter 5). Figure 6-2a shows, during efflorescence, two distinct crystallization events were observed. First, a small dark crystal appeared early during dehydration which did not grow much in size as the droplet kept reducing in size, then a second crystal appeared which grew much larger in size before the droplet completely dehydrated. During efflorescence, the reverse happened, i.e. the larger crystal

started to grow smaller and disappeared first before the smaller crystal dissolved. It was possible to distinguish between them because the crystals did not move around between efflorescence and deliquescence, highlighting another advantage of the static trap method.

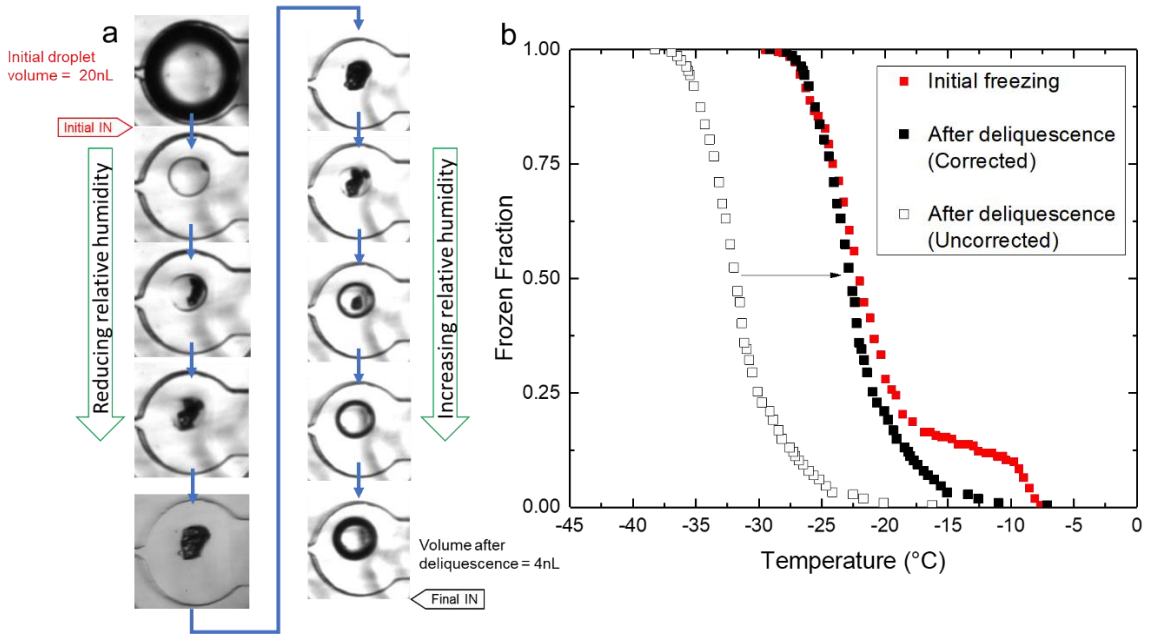


Figure 6-2: SSML efflorescence and deliquescence experiment. (a) A single droplet efflorescing and deliquescing during the experiment. (b) The frozen fraction curves obtained with the droplets initially loaded into the device, and after deliquescence. Curves shown with filled squares have been corrected for freezing point depression due to different amounts of solutes present in the drops at the points where the IN experiment was performed. Also shown with open squares is the uncorrected frozen fraction curve for the droplets after deliquescence. The arrow shows the direction of the correction.

IN measurement was also performed on the droplets, the results are shown in Figure 6-2b. Please note that this sample was loaded with the contaminated 10µm filter as discussed in Chapter 5. In the initial spectra shown in red, we note a fraction of droplets

freeze before -15°C which is attributable to the contamination. However, after efflorescence and deliquescence, this fraction disappears. The second IN curve has been corrected for freezing point depression due to reduced water activity and shifted for ease of comparison with the initial curve. This poses an interesting question about the freeze-thaw behavior of biological INPs in salty water, and future studies will be performed to investigate this peculiar behavior.

6.1.2 Marine organic dissolved matter phase and chemical characterization

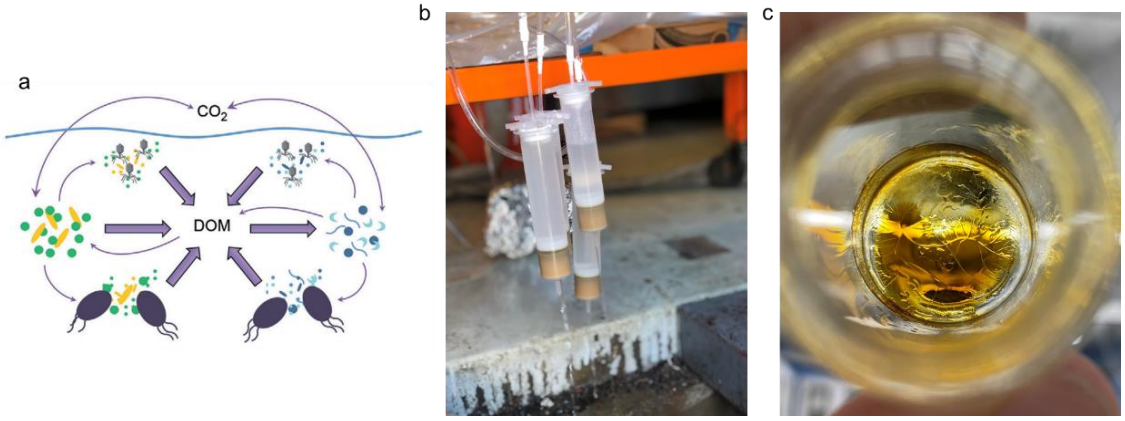


Figure 6-3: (a) Marine dissolved organic matter (mDOM) generation pathways. (b) Solid phase extraction of mDOM samples from SeaSCAPE water. (c) Eluted and dried mDOM. Images courtesy of Michael Alves, University of California San Diego.

Dissolved organic matter (DOM) is a common component of marine ecosystem structure and function and consists of many(millions) organic carbon, nitrogen and phosphorus rich compounds created due to the interaction of hydrologic cycle with the biological life forms in oceans, rivers, groundwater and rain (Figure 6-3a). Samples of marine DOM (mDOM) was collected from the water in the waveflume from SeaSCAPE

using solid phase extraction cartridges (Figure 6-3b) and stored dried. There is an increased interest in complete characterization of these samples from many different instruments and methods. Our group has focused on visible phase transitions during dehydration, and chemical characterization in situ.

Preliminary studies with Raman spectroscope by our collaborators at UC San Diego showed considerable fluorescence which means that Raman cannot be used to identify the components in these drops. As an alternative, FTIR can be used to identify the compounds. However, FTIR *in situ* is not simple to perform in a microfluidic well. The reason is that the usual glass and PDMS materials used are not transparent to IR in the spectra range of interest. There has been a few attempts at using PDMS devices for FTIR [288–290]. These devices still place a thin layer of PDMS in the optical path and hence has high background noise. IR transparent materials like sapphire or CaF_2 can be used instead. However, fabricating devices entirely out of these devices is not a trivial task. Fabrication has to be done either through some kind of dry or wet etching process, machining or by clever design of soft layers which can still be molded but does not block the IR light [288]. A preliminary design and machined mockup of a phase design for this experiment was done and shown in Figure 6-4. The mockup was machined in poly-methylmethacrylate (PMMA). The two halves of the device can be sealed with heat and pressure or with PMMA glue.

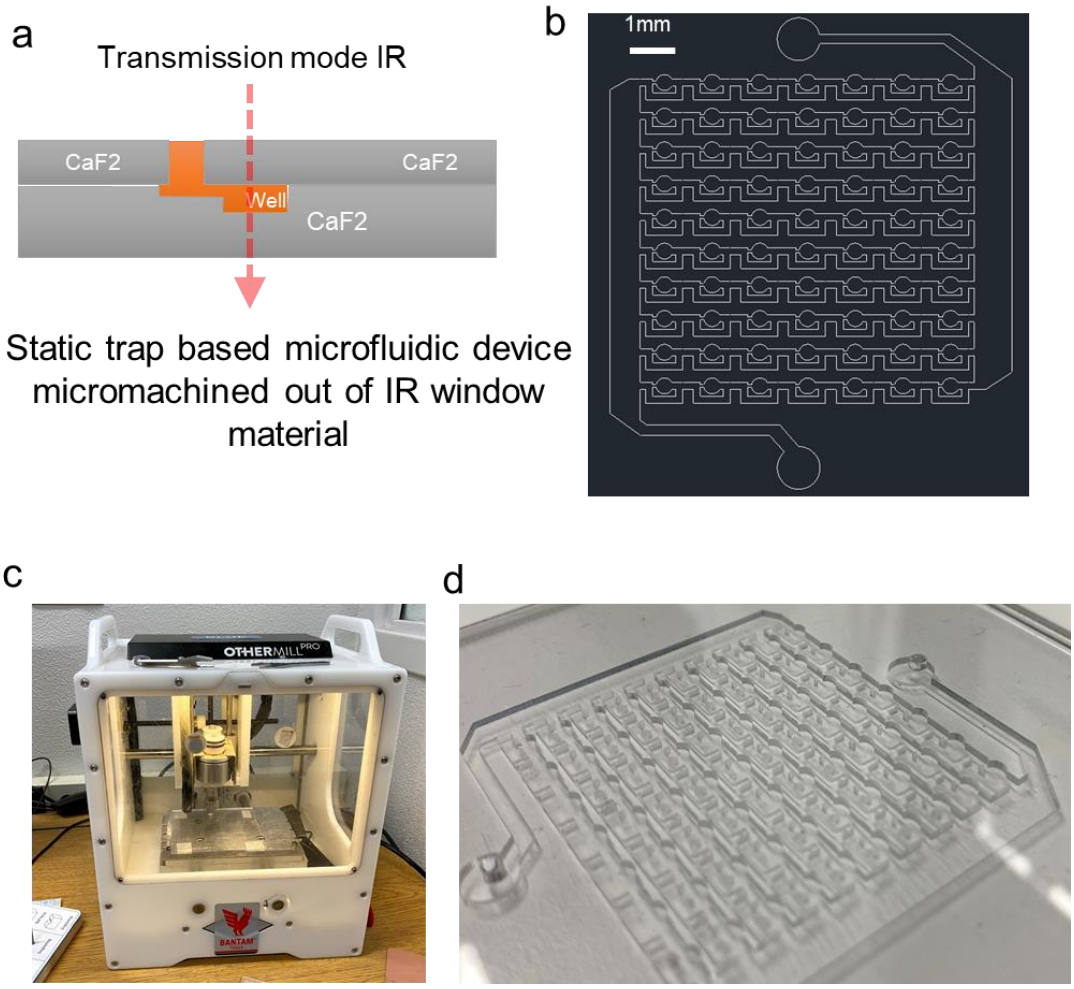


Figure 6-4: (a) Cross-section of the proposed microfluidic device. (b) Design of the microfluidic device. (c) Desktop milling machine used for making a mockup device. (d) The mockup device machined in PMMA.

6.1.3 Thermocouple-based flow-through device

An alternative design for droplet temperature measurement in the flow-through droplet freezing device described in Chapter 3 was designed. The design and temperature measurement method are shown in Figure 6-5. The design is based on a recent research by Stan et al. [74,75] where a similar microfluidic device was used with thermocouples to measure

droplet temperature for counting INPs. In this design, there are two parallel channels on the same microfluidic chip fabricated close to each other. The main flow channel is where the sample droplets are carried in a flow of oil. The secondary flow channel is only meant for oil flow and is a reference or proxy for measuring the temperature in the main channel. The idea being that thermocouple(s) can be inserted into this reference channel and measure the temperature of the flowing liquid with the assumption that the temperatures in the same axial location in both channels are identical or very close. Any differences can be characterized through calibration. The reason the reference channel is required is because to measure the fluid temperature, the thermocouple tip must be immersed in the flow and that would cause issues with droplet flow through the actual sample channel.

In the design shown here, provision for multiple thermocouples was created since the underlying temperature controlled platform is able to create temperature gradients and hence there is an opportunity to measure a range of IN temperatures at the same time using different thermocouples, thus reducing the sampling requirements and experimental time.

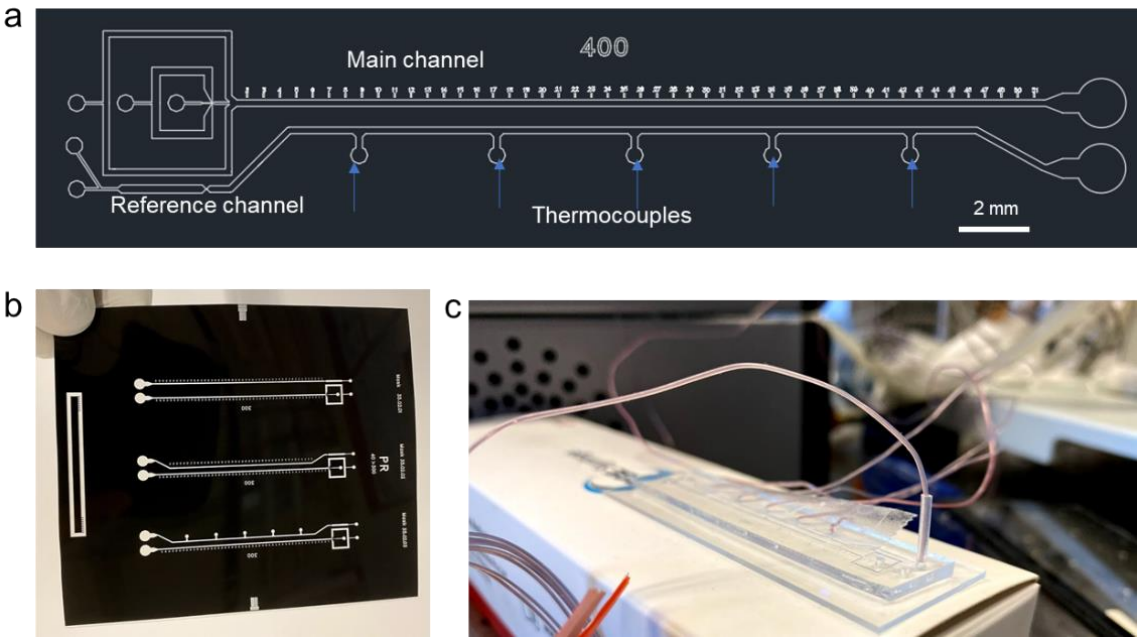


Figure 6-5: Flow-through device design with thermocouples replacing the platinum thin film temperature sensors shown in Chapter 5. (a) The design of the device. There are two flow channels designed to provide similar flow rates and pressure drops. (b) The vinyl photomask with a few additional designs. (c) Thermocouples plugged into the device for calibration.

6.2 Final conclusions

In this thesis, three distinct droplet microfluidic setups for studying different aspects of atmospheric aerosols have been developed, and temperature dependent phase transitions such as LLPS, efflorescence and ice nucleation in droplets were studied. In Chapter 2, the smaller static trap-based system was used for studying the temperature dependence of phase separation and efflorescence in ternary aqueous droplet mimic systems comprising of ammonium sulfate or sodium chloride in solution with various organics including 3-methylglutaric acid (3-MGA), a common constituent in aerosols. This

study showed that the microfluidic setup could be used for studying the relative humidity of phase separation and it was a function of temperature. In Chapter 3, the flow-through setup was constructed and automation algorithms using machine learning were created to detect high-throughput freezing. In Chapter 4, the flow-through setup and detection algorithms were used to study IN characteristics of Snomax, a highly efficient biological ice nucleating particle. Its molecular structure responsible for ice nucleating ability was also investigated using an FTIR. This study showed the feasibility for an automated high throughput ice nucleation counter and is the foundation for future microfluidic devices on the same platform augmenting the abilities of the current device. In Chapter 5, the large static trap array was used to investigate IN property and residual dry particle morphology of samples collected from a marine source. This device performed on par with an established ice spectrometer and had the additional advantage of allowing phase transition studies on the same droplets. It showed that sea surface microlayer samples have different dry particle morphology compared to bulk sweater and that the former has a warmer ice nucleation spectrum.

These devices and the initial studies performed show great potential to be used with more real-world samples and can be easily modified to meet the need for custom sampling and experimental protocols. The advantage of good temperature and relative humidity control combined with high throughput and low sample volume requirements make the methods described in this thesis ideal for aerosol science and the benchmarking against existing instrumentation shows the viability of these methods. There is a huge need in the atmospheric chemistry literature for data on temperature dependence of atmospheric

aerosols in terms of their phase state and heterogeneous chemistry. These methods will allow for creation of large datasets on highly relevant chemical systems relatively quickly and inexpensively. Eventually, the data could be used to enrich climate modeling and hence our understanding of our planet.

Bibliography

1. Wallace, J. & Hobbs, P. *Atmospheric Science - 2nd Edition*. (Academic Press, 2006).

2. Seinfeld, J. H. & Pandis, S. N. *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. (Wiley, 2016).

3. Chin, M., Diehl, T., Ginoux, P. & Malm, W. Intercontinental transport of pollution and dust aerosols: Implications for regional air quality. *Atmos. Chem. Phys.* **7**, 5501–5517 (2007).

4. Ervens, B., Turpin, B. J. & Weber, R. J. Secondary organic aerosol formation in cloud droplets and aqueous particles (aqSOA): A review of laboratory, field and model studies. *Atmospheric Chemistry and Physics* **11**, 11069–11102 (2011).

5. Hoyle, C. R. *et al.* A review of the anthropogenic influence on biogenic secondary organic aerosol. *Atmospheric Chemistry and Physics* **11**, 321–343 (2011).

6. Farmer, D. K., Cappa, C. D. & Kreidenweis, S. M. Atmospheric Processes and Their Controlling Influence on Cloud Condensation Nuclei Activity. *Chem. Rev.* **115**, 4199–4217 (2015).

7. Kerminen, V. M. *et al.* Cloud condensation nuclei production associated with atmospheric nucleation: A synthesis based on existing literature and new results. *Atmos. Chem. Phys.* **12**, 12037–12059 (2012).

8. Wark, K. & Richards, D. E. *Thermodynamics*. (WCB/McGraw-Hill, 1999).

9. Booth, A. M., Topping, D. O., McFiggans, G. & Percival, C. J. Surface tension of mixed inorganic and dicarboxylic acid aqueous solutions at 298.15 K and their importance for cloud activation predictions. *Phys. Chem. Chem. Phys.* **11**, 8021 (2009).

10. Frosch, M., Prisle, N. L., Bilde, M., Varga, Z. & Kiss, G. Joint effect of organic acids and inorganic salts on cloud droplet activation. *Atmos. Chem. Phys* **11**, 3895–3911 (2011).

11. Cruz, C. N. & Pandis, S. N. The effect of organic coatings on the cloud condensation nuclei activation of inorganic atmospheric aerosol. *J. Geophys. Res. Atmos.* **103**, 13111–13123 (1998).

12. Kanji, Z. A. *et al.* Overview of Ice Nucleating Particles. *Meteorol. Monogr.* **58**, 1.1-1.33 (2017).

13. Freedman, M. A. Phase separation in organic aerosol. *Chem. Soc. Rev.* **46**, 7694–7705 (2017).

14. Nandy, L. & Dutcher, C. S. Phase Behavior of Ammonium Sulfate with Organic Acid Solutions in Aqueous Aerosol Mimics Using Microfluidic Traps. *J. Phys. Chem. B* **122**, 3480–3490 (2018).

15. You, Y. *et al.* Images reveal that atmospheric particles can undergo liquid-liquid phase separations. *Proc. Natl. Acad. Sci.* **109**, 13188–13193 (2012).

16. Obrien, R. E. *et al.* Liquid-liquid phase separation in aerosol particles: Imaging at the nanometer scale. *Environ. Sci. Technol.* **49**, 4995–5002 (2015).

17. Twomey, S. Pollution and the planetary albedo. *Atmos. Environ.* **8**, 1251–1256 (1974).

18. O'Dowd, C. D. *et al.* Biogenically driven organic contribution to marine aerosol. *Nature* **431**, 676–680 (2004).

19. Meskhidze, N. & Nenes, A. Phytoplankton and cloudiness in the Southern Ocean. *Science (80-.).* **314**, 1419–1423 (2006).

20. Houghton, J. T., Genkins, G. J. & Ephraums, J. J. *Climate Change: The IPCC Scientific Assessment*. (1990).

21. Stocker, T. F. *et al. IPCC, 2013: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. (2013).

22. Stocker, T. F. *et al. IPCC, 2013: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. (2013). doi:10.1017/CBO9781107415324

23. Xie, S.-P. *et al.* Towards predictive understanding of regional climate change. *Nat. Clim. Chang.* **5**, 921–930 (2015).

24. Rich, D. Q. *et al.* The triggering of myocardial infarction by fine particles is enhanced when particles are enriched in secondary species. *Environ. Sci. Technol.* **47**, 9414–9423 (2013).

25. Raaschou-Nielsen, O. *et al.* Air pollution and lung cancer incidence in 17 European cohorts: Prospective analyses from the European Study of Cohorts for Air Pollution Effects (ESCAPE). *Lancet Oncol.* **14**, 813–822 (2013).

26. Miranda, M. L., Edwards, S. E., Chang, H. H. & Auten, R. L. Proximity to roadways and pregnancy outcomes. *J. Expo. Sci. Environ. Epidemiol.* **23**, 32–38 (2013).

27. van Doremalen, N. *et al.* Aerosol and Surface Stability of SARS-CoV-2 as Compared with SARS-CoV-1. *N. Engl. J. Med.* **382**, 1564–1567 (2020).

28. Martin, S. T. Phase transitions of aqueous atmospheric particles. *Chem. Rev.* **100**, 3403–3453 (2000).

29. Stewart, D. J. *et al.* Liquid-liquid phase separation in mixed organic/inorganic single aqueous aerosol droplets. *J. Phys. Chem. A* **119**, 4177–4190 (2015).

30. Nandy, L. *et al.* Multistep Phase Transitions in Sea Surface Microlayer Droplets and Aerosol Mimics using Microfluidic Wells. *ACS Earth Sp. Chem.* **3**, 1260–1267 (2019).

31. Bertram, A. K. *et al.* Predicting the relative humidities of liquid-liquid phase separation, efflorescence, and deliquescence of mixed particles of ammonium sulfate, organic material, and water using the organic-to-sulfate mass ratio of the

particle and the oxygen-to-carbon ele. *Atmos. Chem. Phys.* **11**, 10995–11006 (2011).

32. Roy, P. *et al.* Temperature-Dependent Phase Transitions of Aqueous Aerosol Droplet Systems in Microfluidic Traps. *ACS Earth Sp. Chem.* **4**, 1527–1539 (2020).

33. Song, M., Marcolli, C., Krieger, U. K., Lienhard, D. M. & Peter, T. Morphologies of mixed organic/inorganic/aqueous aerosol droplets. *Faraday Discuss.* **165**, 289–316 (2013).

34. Freedman, M. A., Hasenkopf, C. A., Beaver, M. R. & Tolbert, M. A. Optical properties of internally mixed aerosol particles composed of dicarboxylic acids and ammonium sulfate. *J. Phys. Chem. A* **113**, 13584–13592 (2009).

35. Cooke, W. F., Liousse, C., Cachier, H. & Feichter, J. Construction of a 1° × 1° fossil fuel emission data set for carbonaceous aerosol and implementation and radiative impact in the ECHAM4 model. *J. Geophys. Res. Atmos.* **104**, 22137–22162 (1999).

36. Vignati, E., Wilson, J. & Stier, P. M7: An efficient size-resolved aerosol microphysics module for large-scale aerosol transport models. *J. Geophys. Res. D Atmos.* **109**, 1–17 (2004).

37. Mie, G. Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen. *Ann. Phys.* **330**, 377–445 (1908).

38. Riemer, N., West, M., Zaveri, R. A. & Easter, R. C. Simulating the evolution of soot mixing state with a particle-resolved aerosol model. *J. Geophys. Res.* **114**, D09202 (2009).

39. Curtis, J. H., Riemer, N. & West, M. A single-column particle-resolved model for simulating the vertical distribution of aerosol mixing state: WRF-PartMC-MOSAIC-SCM v1.0. *Geosci. Model Dev.* **10**, 4057–4079 (2017).

40. Ching, J., Zaveri, R. A., Easter, R. C., Riemer, N. & Fast, J. D. A three-dimensional sectional representation of aerosol mixing state for simulating optical properties and cloud condensation nuclei. *J. Geophys. Res. Atmos.* **121**, 5912–5929 (2016).

41. Matsui, H. *et al.* Development and validation of a black carbon mixing state resolved three-dimensional model: Aging processes and radiative impact. *J. Geophys. Res. Atmos.* **118**, 2304–2326 (2013).

42. Ching, J., Riemer, N. & West, M. Impacts of black carbon mixing state on black carbon nucleation scavenging: Insights from a particle-resolved model. *J. Geophys. Res. Atmos.* **117**, 1–21 (2012).

43. Pruppacher, H. R. A New Look at Homogeneous Ice Nucleation in Supercooled Water Drops. *J. Atmos. Sci.* **52**, 1924–1933 (2002).

44. DeMott, P. J. *et al.* Predicting global atmospheric ice nuclei distributions and their impacts on climate. *Proc. Natl. Acad. Sci.* **107**, 11217–11222 (2010).

45. Murray, B. J., Broadley, S. L., Wilson, T. W., Atkinson, J. D. & Wills, R. H. Heterogeneous freezing of water droplets containing kaolinite particles. *Atmos. Chem. Phys.* **11**, 4191–4207 (2011).

46. Hill, T. C. J. *et al.* Measurement of ice nucleation-active bacteria on plants and in

precipitation by quantitative PCR. *Appl. Environ. Microbiol.* **80**, 1256–1267 (2014).

47. Knopf, D. A., Alpert, P. A. & Wang, B. The Role of Organic Aerosol in Atmospheric Ice Nucleation: A Review. *ACS Earth and Space Chemistry* **2**, 168–202 (2018).

48. Hill, T. C. J. *et al.* Sources of organic ice nucleating particles in soils. *Atmos. Chem. Phys.* **16**, 7195–7211 (2016).

49. Fletcher, N. H. *The Physics of Rainclouds*. (Cambridge University Press, 1962).

50. DeMott, P. J. *et al.* Predicting global atmospheric ice nuclei distributions and their impacts on climate. *Proc. Natl. Acad. Sci. U. S. A.* **107**, 11217–11222 (2010).

51. Pruppacher, H. R. & Klett, J. D. *Microphysics of Clouds and Precipitation*. *Microphysics of Clouds and Precipitation* (Springer Netherlands, 1997). doi:10.1007/978-94-009-9905-3

52. Hoose, C. & Möhler, O. Heterogeneous ice nucleation on atmospheric aerosols: a review of results from laboratory experiments. *Atmos. Chem. Phys. Discuss.* **12**, 12531–12621 (2012).

53. Lohmann, U. & Diehl, K. Sensitivity studies of the importance of dust ice nuclei for the indirect aerosol effect on stratiform mixed-phase clouds. *J. Atmos. Sci.* **63**, 968–982 (2006).

54. *Cirrus*. (Oxford University Press, 2002).

55. Seifert, P. *et al.* Ice formation in ash-influenced clouds after the eruption of the Eyjafjallajkull volcano in April 2010. *J. Geophys. Res. Atmos.* **116**, 1–14 (2011).

56. Vali, G., DeMott, P. J., Möhler, O. & Whale, T. F. Technical Note: A proposal for ice nucleation terminology. *Atmos. Chem. Phys.* **15**, 10263–10270 (2015).

57. Hoose, C., Kristjánsson, J. E., Chen, J. P. & Hazra, A. A classical-theory-based parameterization of heterogeneous ice nucleation by mineral dust, soot, and biological particles in a global climate model. *J. Atmos. Sci.* **67**, 2483–2503 (2010).

58. Meyers, M. P., DeMott, P. J. & Cotton, W. R. New Primary Ice-Nucleation Parameterizations in an Explicit Cloud Model. *Journal of Applied Meteorology* **31**, 708–721 (2002).

59. DeMott, P. J. *et al.* Sea spray aerosol as a unique source of ice nucleating particles. *Proc. Natl. Acad. Sci. U. S. A.* **113**, 5797–803 (2016).

60. Boyer, H. C. & Dutcher, C. S. Atmospheric Aqueous Aerosol Surface Tensions: Isotherm-Based Modeling and Biphasic Microfluidic Measurements. *J. Phys. Chem. A* **121**, 4733–4742 (2017).

61. Reid, J. P. *et al.* The viscosity of atmospherically relevant organic particles. *Nat. Commun.* **9**, 1–14 (2018).

62. Sullivan, R. C., Gorkowski, K. & Jahn, L. *Characterization of Individual Aerosol Particles*. *Physical Chemistry of Gas-Liquid Interfaces* (Elsevier Inc., 2018). doi:10.1016/B978-0-12-813641-6.00012-1

63. Lee, H. D. & Tivanski, A. V. Atomic force microscopy: an emerging tool in measuring the phase state and surface tension of individual aerosol particles. *Annu. Rev. Phys. Chem.* (2020).

64. Pipe, C. J. & McKinley, G. H. Microfluidic rheometry. *Mech. Res. Commun.* **36**, 110–120 (2009).

65. Mukhopadhyay, A. & Granick, S. Micro- and nanorheology. *Curr. Opin. Colloid Interface Sci.* **6**, 423–429 (2001).

66. Seinfeld, J. H. & Pandis, S. N. *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. *Atmospheric Chemistry and Physics* (2006). doi:10.1063/1.882420

67. Sackmann, E. K., Fulton, A. L. & Beebe, D. J. The present and future role of microfluidics in biomedical research. *Nature* **507**, 181–189 (2014).

68. Lifton, V. A. Microfluidics: An enabling screening technology for enhanced oil recovery (EOR). *Lab Chip* **16**, 1777–1796 (2016).

69. Jokerst, J. C., Emory, J. M. & Henry, C. S. Advances in microfluidics for environmental analysis. *Analyst* **137**, 24–34 (2012).

70. Narayan, S., Moravec, D. B., Hauser, B. G., Dallas, A. J. & Dutcher, C. S. Removing water from diesel fuel : Understanding the impact of droplet size on dynamic interfacial tension of water-in-fuel emulsions. *Energy & Fuels* **32**, 7326–7337 (2018).

71. Chen, Y. & Dutcher, C. S. Size dependent droplet interfacial tension and surfactant transport in liquid-liquid systems, with applications in shipboard oily bilgewater emulsions. *Soft Matter* **16**, 2994–3004 (2020).

72. Metcalf, A. R., Narayan, S. & Dutcher, C. S. A review of microfluidic concepts and applications for atmospheric aerosol science. *Aerosol Sci. Technol.* **52**, 310–329 (2018).

73. Tarn, M. D. *et al.* The study of atmospheric ice-nucleating particles via microfluidically generated droplets. *Microfluid. Nanofluidics* **22**, 1–25 (2018).

74. Tarn, M. D. *et al.* On-chip analysis of atmospheric ice-nucleating particles in continuous flow. *Lab Chip* **20**, 2889–2910 (2020).

75. Porter, G. C. E., Sikora, S. N. F., Shim, J. U., Murray, B. J. & Tarn, M. D. On-chip density-based sorting of supercooled droplets and frozen droplets in continuous flow. *Lab Chip* **20**, 3876–3887 (2020).

76. Brubaker, T. *et al.* Development and characterization of a "store and create" microfluidic device to determine the heterogeneous freezing properties of ice nucleating particles. *Aerosol Sci. Technol.* **54**, 79–93 (2020).

77. Weng, L. *et al.* Bacterial Ice Nucleation in Monodisperse D2O and H2O-in-Oil Emulsions. *Langmuir* **32**, 9229–9236 (2016).

78. Brubaker, T. *et al.* Development and characterization of a "store and create" microfluidic device to determine the heterogeneous freezing properties of ice

nucleating particles. *Aerosol Sci. Technol.* **54**, 79–93 (2020).

79. Häusler, T., Witek, L., Felgitsch, L., Hitzenberger, R. & Grothe, H. Freezing on a Chip-A new approach to determine heterogeneous ice nucleation of micrometer-sized water droplets. *Atmosphere (Basel).* **9**, (2018).

80. Stan, C. A. *et al.* A microfluidic apparatus for the study of ice nucleation in supercooled water drops. *Lab Chip* **9**, 2293 (2009).

81. You, Y. *et al.* Images reveal that atmospheric particles can undergo liquid-liquid phase separations. *Proc. Natl. Acad. Sci.* **109**, 13188–13193 (2012).

82. McMurry, P. H. *et al.* Elemental composition and morphology of individual particles separated by size and hygroscopicity with the TDMA. *Atmos. Environ.* **30**, 101–108 (1996).

83. Zardini, A. A. *et al.* A combined particle trap/HTDMA hygroscopicity study of mixed inorganic/organic aerosol particles. *Atmos. Chem. Phys.* **8**, 5589–5601 (2008).

84. Cheng, Y., Su, H., Koop, T., Mikhailov, E. & Pöschl, U. Size dependence of phase transitions in aerosol nanoparticles. *Nat. Commun.* **6**, 5923 (2015).

85. Bzdek, B. R., Power, R. M., Simpson, S. H., Reid, J. P. & Royall, C. P. Precise, contactless measurements of the surface tension of picolitre aerosol droplets. *Chem. Sci.* **7**, 274–285 (2016).

86. Möhler, O. *et al.* The effect of organic coating on the heterogeneous ice nucleation efficiency of mineral dust aerosols. *Environ. Res. Lett.* **3**, 025007 (2008).

87. Mason, B. J. The cloud chamber as a tool in cloud physics. *Contemp. Phys.* **4**, 27–48 (1962).

88. Schnaiter, M. *et al.* Cloud chamber experiments on the origin of ice crystal complexity in cirrus clouds. *Atmos. Chem. Phys.* **16**, 5091–5110 (2016).

89. Lee, A. K. Y., Ling, T. Y. & Chan, C. K. Understanding hygroscopic growth and phase transformation of aerosols using single particle Raman spectroscopy in an electrodynamic balance. *Faraday Discuss.* **137**, 245–263 (2008).

90. Nenes, A., Chuang, P. Y., Flagan, R. C. & Seinfeld, J. H. A theoretical analysis of cloud condensation nucleus (CCN) instruments. *J. Geophys. Res. Atmos.* **106**, 3449–3474 (2001).

91. Roberts, G. C. & Nenes, A. A continuous-flow streamwise thermal-gradient CCN chamber for atmospheric measurements. *Aerosol Sci. Technol.* **39**, 206–221 (2005).

92. Metcalf, A. R., Boyer, H. C. & Dutcher, C. S. Interfacial Tensions of Aged Organic Aerosol Particle Mimics Using a Biphasic Microfluidic Platform. *Environ. Sci. Technol.* **50**, 1251–1259 (2016).

93. Marcolli, C. & Krieger, U. K. Phase changes during hygroscopic cycles of mixed organic/inorganic model systems of tropospheric aerosols. *J. Phys. Chem. A* **110**, 1881–1893 (2006).

94. Pandis, S. N., Wexler, A. S. & Seinfeld, J. H. Dynamics of tropospheric aerosols. *J. Phys. Chem.* **99**, 9646–9659 (1995).

95. Lesins, G., Chylek, P. & Lohmann, U. A study of internal and external mixing scenarios and its effect on aerosol optical properties and direct radiative forcing. *J. Geophys. Res. Atmos.* **107**, AAC 5-1-AAC 5-12 (2002).

96. Fard, M. M., Krieger, U. K. & Peter, T. Shortwave radiative impact of liquid-liquid phase separation in brown carbon aerosols. *Atmos. Chem. Phys.* **18**, 13511–13530 (2018).

97. Hodas, N., Zuend, A., Mui, W., Flagan, R. C. & Seinfeld, J. H. Influence of particle-phase state on the hygroscopic behavior of mixed organic-inorganic aerosols. *Atmos. Chem. Phys.* **15**, 5027–5045 (2015).

98. Cosman, L. M. & Bertram, A. K. Reactive uptake of N2O5 on aqueous H 2so4 solutions coated with 1-component and 2-component monolayers. *J. Phys. Chem. A* **112**, 4625–4635 (2008).

99. Cosman, L. M., Knopf, D. A. & Bertram, A. K. N2O5 reactive uptake on aqueous sulfuric acid solutions coated with branched and straight-chain insoluble organic surfactants. *J. Phys. Chem. A* **112**, 2386–2396 (2008).

100. Altaf, M. B., Dutcher, D. D., Raymond, T. M. & Freedman, M. A. Effect of Particle Morphology on Cloud Condensation Nuclei Activity. *ACS Earth Sp. Chem.* **2**, 634–639 (2018).

101. Peckhaus, A., Kiselev, A., Wagner, R., Duft, D. & Leisner, T. Temperature-dependent formation of NaCl dihydrate in levitated NaCl and sea salt aerosol particles. *J. Chem. Phys.* **145**, 244503 (2016).

102. Tang, I. N. & Munkelwitz, H. R. Composition and temperature dependence of the deliquescence properties of hygroscopic aerosols. *Atmos. Environ. Part A, Gen. Top.* **27**, 467–473 (1993).

103. Gorkowski, K. *et al.* Advanced aerosol optical tweezers chamber design to facilitate phase-separation and equilibration timescale experiments on complex droplets. *Aerosol Sci. Technol.* **50**, 1327–1341 (2016).

104. Sobanska, S., Coeur, C., Maenhaut, W. & Adams, F. SEM-EDX Characterisation of Tropospheric Aerosols in the Negev Desert (Israel). *J. Atmos. Chem.* **44**, 299–322 (2003).

105. Veghte, D. P., Bittner, D. R. & Freedman, M. A. Cryo-transmission electron microscopy imaging of the morphology of submicrometer aerosol containing organic acids and ammonium sulfate. *Anal. Chem.* **86**, 2436–2442 (2014).

106. Takahama, S., Gilardoni, S. & Russell, L. M. Single-particle oxidation state and morphology of atmospheric iron aerosols. *J. Geophys. Res.* **113**, D22202 (2008).

107. Zielinski, A. T. *et al.* Measuring Aerosol Phase Changes and Hygroscopicity with a Microresonator Mass Sensor. *Anal. Chem.* **90**, 9716–9724 (2018).

108. Schindelholz, E., Tsui, L. K. & Kelly, R. G. Hygroscopic particle behavior studied

by interdigitated array microelectrode impedance sensors. *J. Phys. Chem. A* **118**, 167–177 (2014).

109. Arenas, K. J. L., Schill, S. R., Malla, A. & Hudson, P. K. Deliquescence phase transition measurements by quartz crystal microbalance frequency shifts. *J. Phys. Chem. A* **116**, 7658–7667 (2012).

110. Bateman, A. P., Belassein, H. & Martin, S. T. Impactor apparatus for the study of particle rebound: Relative humidity and capillary forces. *Aerosol Sci. Technol.* **48**, 42–52 (2014).

111. Bateman, A. P. *et al.* Sub-micrometre particulate matter is primarily in liquid form over Amazon rainforest. *Nat. Geosci.* **9**, 34–37 (2016).

112. Bateman, A. P. *et al.* Anthropogenic influences on the physical state of submicron particulate matter over a tropical forest. *Atmos. Chem. Phys.* **17**, 1759–1773 (2017).

113. Jain, S. & Petrucci, G. A. A new method to measure aerosol particle bounce using a cascade electrical low pressure impactor. *Aerosol Sci. Technol.* **49**, 390–399 (2015).

114. McMurry, P. H. Chapter 17 A review of atmospheric aerosol measurements. in *Developments in Environmental Science* **1**, 443–517 (2002).

115. Bleier, B. J., Anna, S. L. & Walker, L. M. Microfluidic Droplet-Based Tool to Determine Phase Behavior of a Fluid System with High Composition Resolution. *J. Phys. Chem. B* **122**, 4067–4076 (2018).

116. Shim, J. U. *et al.* Control and measurement of the phase behavior of aqueous solutions using microfluidics. *J. Am. Chem. Soc.* **129**, 8825–8835 (2007).

117. You, Y., Renbaum-Wolff, L. & Bertram, A. K. Liquid–liquid phase separation in particles containing organics mixed with ammonium sulfate, ammonium bisulfate, ammonium nitrate or sodium chloride. *Atmos. Chem. Phys.* **13**, 11723–11734 (2013).

118. Song, M., Marcolli, C., Krieger, U. K., Zuend, A. & Peter, T. Liquid-liquid phase separation in aerosol particles: Dependence on O:C, organic functionalities, and compositional complexity. *Geophys. Res. Lett.* **39**, L19801 (2012).

119. Song, M. *et al.* Liquid–liquid phase separation and viscosity within secondary organic aerosol generated from diesel fuel vapors. *Atmos. Chem. Phys.* **19**, 12515–12529 (2019).

120. You, Y. & Bertram, A. K. Effects of molecular weight and temperature on liquid–liquid phase separation in particles containing organic species and inorganic salts. *Atmos. Chem. Phys.* **15**, 1351–1365 (2015).

121. Song, M., Ham, S., Andrews, R. J., You, Y. & Bertram, A. K. Liquid-liquid phase separation in organic particles containing one and two organic species: Importance of the average O:C. *Atmos. Chem. Phys.* **18**, 12075–12084 (2018).

122. Ham, S., Bin Babar, Z., Bong Lee, J., Lim, H. J. & Song, M. Liquid-liquid phase separation in secondary organic aerosol particles produced from α-pinene

ozonolysis and α-pinene photooxidation with/without ammonia. *Atmos. Chem. Phys.* **19**, 9321–9331 (2019).

123. Ott, E.-J. E., Tackman, E. C. & Freedman, M. A. Effects of Sucrose on Phase Transitions of Organic/Inorganic Aerosols. *ACS Earth Sp. Chem.* **4**, 591–601 (2020).

124. Zobrist, B., Marcolli, C., Pedernera, D. A. & Koop, T. Do atmospheric aerosols form glasses? *Atmos. Chem. Phys.* **8**, 5221–5244 (2008).

125. Chen, Q. *et al.* Mass spectral characterization of submicron biogenic organic particles in the Amazon Basin. *Geophys. Res. Lett.* **36**, L20806 (2009).

126. Sun, Y. *et al.* Characterization of summer organic and inorganic aerosols in Beijing, China with an Aerosol Chemical Speciation Monitor. *Atmos. Environ.* **51**, 250–259 (2012).

127. Jimenez, J. L. *et al.* Evolution of organic aerosols in the atmosphere. *Science (80-.).* **326**, 1525–1529 (2009).

128. Martin, S. T. Phase transitions of aqueous atmospheric particles. *Chem. Rev.* **100**, 3403–3453 (2000).

129. Cziczo, D. J. & Abbatt, J. P. D. Infrared Observations of the Response of NaCl, MgCl2, NH4HSO4, and NH4NO3 Aerosols to Changes in Relative Humidity from 298 to 238 K. *J. Phys. Chem. A* **104**, 2038–2047 (2000).

130. Wise, M. E. *et al.* Depositional ice nucleation onto crystalline hydrated NaCl particles: A new mechanism for ice formation in the troposphere. *Atmos. Chem. Phys.* **12**, 1121–1134 (2012).

131. Tong, H. J., Qian, Z. G., Reid Jonathan, P. & Zhang, Y. H. High temporal and spatial resolution measurements of the rapid efflorescence of sea salt droplets. *Wuli Huaxue Xuebao/ Acta Phys. - Chim. Sin.* **27**, 2521–2527 (2011).

132. Qian, Z. G., Wang, F., Zheng, Y. X., Yu, J. Y. & Zhang, Y. H. Crystallization kinetics of sea-salt aerosols studied by high-speed photography. *Chinese Sci. Bull.* **57**, 591–594 (2012).

133. Sun, M., Bithi, S. S. & Vanapalli, S. A. Microfluidic static droplet arrays with tuneable gradients in material composition. *Lab Chip* **11**, 3949–3952 (2011).

134. Nandy, L. & Dutcher, C. S. Isotherm-Based Thermodynamic Model for Solute Activities of Asymmetric Electrolyte Aqueous Solutions. *J. Phys. Chem. A* **121**, 6957–6965 (2017).

135. Cziczo, D. J., Nowak, J. B., Hu, J. H. & Abbatt, J. P. D. Infrared spectroscopy of model tropospheric aerosols as a function of relative humidity: Observation of deliquescence and crystallization. *J. Geophys. Res. Atmos.* **102**, 18843–18850 (1997).

136. Schlenker, J. C., Malinowski, A., Martin, S. T., Hung, H. M. & Rudich, Y. Crystals formed at 293 K by aqueous sulfate-nitrate-ammonium-proton aerosol particles. *J.*

Phys. Chem. A **108**, 9375–9383 (2004).

137. Clegg, S. L., Ho, S. S., Chan, C. K. & Brimblecombe, P. Thermodynamic Properties of Aqueous (NH4)2SO4 to High Supersaturation as a Function of Temperature. *J. Chem. Eng. Data* **40**, 1079–1090 (1995).

138. Friese, E. & Ebel, A. Temperature dependent thermodynamic model of the system H +-NH4+-Na+-SO42-NO3-Cl--H2O. *J. Phys. Chem. A* **114**, 11595–11631 (2010).

139. Zuend, A., Marcolli, C., Luo, B. P. & Peter, T. A thermodynamic model of mixed organic-inorganic aerosols to predict activity coefficients. *Atmos. Chem. Phys.* **8**, 4559–4593 (2008).

140. Ganbavale, G., Zuend, A., Marcolli, C. & Peter, T. Improved AIOMFAC model parameterisation of the temperature dependence of activity coefficients for aqueous organic mixtures. *Atmos. Chem. Phys.* **15**, 447–493 (2015).

141. Hanford, K. L. *et al.* Comparative thermodynamic studies of aqueous glutaric acid, ammonium sulfate and sodium chloride aerosol at high humidity. *J. Phys. Chem. A* **112**, 9413–9422 (2008).

142. Pope, F. D., Dennis-Smither, B. J., Griffiths, P. T., Clegg, S. L. & Cox, R. A. Studies of single aerosol particles containing malonic acid, glutaric acid, and their mixtures with sodium chloride. I. Hygroscopic growth. *J. Phys. Chem. A* **114**, 5335–5341 (2010).

143. Lei, T., Zuend, A., Wang, W. G., Zhang, Y. H. & Ge, M. F. Hygroscopicity of

organic compounds from biomass burning and their influence on the water uptake of mixed organic ammonium sulfate aerosols. *Atmos. Chem. Phys.* **14**, 11165–11183 (2014).

144. Estillore, A. D. *et al.* Linking hygroscopicity and the surface microstructure of model inorganic salts, simple and complex carbohydrates, and authentic sea spray aerosol particles. *Phys. Chem. Chem. Phys.* **19**, 21101–21111 (2017).

145. Zuend, A., Marcolli, C., Peter, T. & Seinfeld, J. H. Computation of liquid-liquid equilibria and phase stabilities: implications for RH-dependent gas/particle partitioning of organic-inorganic aerosols. *Atmos. Chem. Phys.* **10**, 7795–7820 (2010).

146. Song, M., Marcolli, C., Krieger, U. K., Zuend, A. & Peter, T. Liquid-liquid phase separation and morphology of internally mixed dicarboxylic acids/ammonium sulfate/water particles. *Atmos. Chem. Phys.* **12**, 2691–2712 (2012).

147. Zuend, A. & Seinfeld, J. H. Modeling the gas-particle partitioning of secondary organic aerosol: The importance of liquid-liquid phase separation. *Atmos. Chem. Phys.* **12**, 3857–3882 (2012).

148. Mael, L. E., Busse, H. & Grassian, V. H. Measurements of Immersion Freezing and Heterogeneous Chemistry of Atmospherically Relevant Single Particles with Micro-Raman Spectroscopy. *Anal. Chem.* **91**, 11138–11145 (2019).

149. Ciobanu, V. G., Marcolli, C., Krieger, U. K., Weers, U. & Peter, T. Liquid-liquid

phase separation in mixed organic/inorganic aerosol particles. *J. Phys. Chem. A* **113**, 10966–10978 (2009).

150. Onasch, T. B., McGraw, R. & Imre, D. Temperature-Dependent Heterogeneous Efflorescence of Mixed Ammonium Sulfate/Calcium Carbonate Particles. *J. Phys. Chem. A* **104**, 10797–10806 (2000).

151. Wise, M. E., Baustian, K. J. & Tolbert, M. A. Internally mixed sulfate and organic particles as potential ice nuclei in the tropical tropopause region. *Proc. Natl. Acad. Sci. U. S. A.* **107**, 6693–6698 (2010).

152. Brooks, S. D., Wise, M. E., Cushing, M. & Tolbert, M. A. Deliquescence behavior of organic/ammonium sulfate aerosol. *Geophys. Res. Lett.* **29**, 23-1-23–4 (2002).

153. *Perry's Chemical Engineers' Handbook*. (McGraw-Hill, 2007).

154. Fard, M. M., Krieger, U. K. & Peter, T. Kinetic Limitation to Inorganic Ion Diffusivity and to Coalescence of Inorganic Inclusions in Viscous Liquid–Liquid Phase-Separated Particles. *J. Phys. Chem. A* **121**, 9284–9296 (2017).

155. You, Y., Smith, M. L., Song, M., Martin, S. T. & Bertram, A. K. Liquid-liquid phase separation in atmospherically relevant particles consisting of organic species and inorganic salts. *Int. Rev. Phys. Chem.* **33**, 43–77 (2014).

156. Hofmeister, F. Zur Lehre von der Wirkung der Salze - Zweite Mittheilung. *Arch. für Exp. Pathol. und Pharmakologie* **24**, 247–260 (1888).

157. Davis, R. D., Lance, S., Gordon, J. A., Ushijima, S. B. & Tolbert, M. A. Contact efflorescence as a pathway for crystallization of atmospherically relevant particles. *Proc. Natl. Acad. Sci. U. S. A.* **112**, 15815–15820 (2015).

158. Davis, R. D. & Tolbert, M. A. Crystal nucleation initiated by transient ion-surface interactions at aerosol interfaces. *Sci. Adv.* **3**, e1700425 (2017).

159. Utoft, A., Kinoshita, K., Bitterfield, D. L. & Needham, D. Manipulating Single Microdroplets of NaCl Solutions: Solvent Dissolution, Microcrystallization, and Crystal Morphology. *Langmuir* **34**, 3626–3641 (2018).

160. Wagner, R., Möhler, O. & Martin Schnaiter, M. Infrared optical constants of crystalline sodium chloride dihydrate: Application to study the crystallization of aqueous sodium chloride solution droplets at low temperatures. *J. Phys. Chem. A* **116**, 8557–8571 (2012).

161. Ono, M., Tozuka, Y., Oguchi, T., Yamamura, S. & Yamamoto, K. Effects of dehydration temperature on water vapor adsorption and dissolution behavior of carbamazepine. *Int. J. Pharm.* **239**, 1–12 (2002).

162. Kachrimanis, K. & Griesser, U. J. Dehydration kinetics and crystal water dynamics of carbamazepine dihydrate. *Pharm. Res.* **29**, 902–921 (2012).

163. Stevens, B. & Boucher, O. Climate science: The aerosol effect. *Nature* **490**, 40–41 (2012).

164. Boucher, O. *et al.* Clouds and Aerosols. in *Climate Change 2013 – The Physical*

Science Basis: Working Group I Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change 571–658 (Cambridge University Press, 2013). doi:10.1017/CBO9781107415324.016

165. Twomey, S. The Influence of Pollution on the Shortwave Albedo of Clouds. *J. Atmos. Sci.* **34**, 1149–1152 (1977).

166. Korolev, A. *et al.* Mixed-Phase Clouds: Progress and Challenges. *Meteorol. Monogr.* **58**, 5.1-5.50 (2017).

167. Murray, B. J., O'sullivan, D., Atkinson, J. D. & Webb, M. E. Ice nucleation by particles immersed in supercooled cloud droplets. *Chem. Soc. Rev.* **41**, 6519–6554 (2012).

168. Garcia, E. *et al.* Biogenic ice nuclei in boundary layer air over two U.S. high plains agricultural regions. *J. Geophys. Res. Atmos.* **117**, 1–12 (2012).

169. DeMott, P. J. *et al.* Comparative measurements of ambient atmospheric concentrations of ice nucleating particles using multiple immersion freezing methods and a continuous flow diffusion chamber. *Atmos. Chem. Phys.* **17**, 11227–11245 (2017).

170. Kanji, Z. A. & Abbatt, J. P. D. The University of Toronto Continuous Flow Diffusion Chamber (UT-CFDC): A Simple Design for Ice Nucleation Studies. *Aerosol Sci. Technol.* **43**, 730–738 (2009).

171. Hartmann, S. *et al.* Atmospheric Chemistry and Physics Homogeneous and

heterogeneous ice nucleation at LACIS: operating principle and theoretical studies. *Atmos. Chem. Phys* **11**, 1753–1767 (2011).

172. Baustian, K. J., Wise, M. E. & Tolbert, M. A. Depositional ice nucleation on solid ammonium sulfate and glutaric acid particles. *Atmos. Chem. Phys.* **10**, 2307–2317 (2010).

173. Hoffmann, N., Duft, D., Kiselev, A. & Leisner, T. Contact freezing efficiency of mineral dust aerosols studied in an electrodynamic balance: Quantitative size and temperature dependence for illite particles. *Faraday Discuss.* **165**, 383–390 (2013).

174. Taji, K., Tachikawa, M. & Nagashima, K. Laser trapping of ice crystals. *Appl. Phys. Lett.* **88**, 2004–2007 (2006).

175. Beall, C. M. *et al.* Automation and heat transfer characterization of immersion mode spectroscopy for analysis of ice nucleating particles. *Atmos. Meas. Tech.* **10**, 2613–2626 (2017).

176. Kunert, A. T. *et al.* Twin-plate Ice Nucleation Assay (TINA) with infrared detection for high-throughput droplet freezing experiments with biological ice nuclei in laboratory and field samples. *Atmos. Meas. Tech.* **11**, 6327–6337 (2018).

177. Peckhaus, A., Kiselev, A., Hiron, T., Ebert, M. & Leisner, T. A comparative study of K-rich and Na/Ca-rich feldspar ice-nucleating particles in a nanoliter droplet freezing assay. *Atmos. Chem. Phys.* **16**, 11477–11496 (2016).

178. Budke, C. & Koop, T. BINARY: An optical freezing array for assessing temperature

and time dependence of heterogeneous ice nucleation. *Atmos. Meas. Tech.* **8**, 689–703 (2015).

179. Cook, F. *et al.* A pyroelectric thermal sensor for automated ice nucleation detection. *Atmos. Meas. Tech* **13**, (2785).

180. Stan, C. A., Tang, S. K. Y. & Whitesides, G. M. Independent control of drop size and velocity in microfluidic flow-focusing generators using variable temperature and flow rate. *Anal. Chem.* **81**, 2399–2402 (2009).

181. Narayanan, S. Gas Assisted Thin-Film Evaporation From Confined Spaces. (Georgia Tech, 2011).

182. Wood, S. E., Baker, M. B. & Swanson, B. D. Instrument for studies of homogeneous and heterogeneous ice nucleation in free-falling supercooled water droplets. *Rev. Sci. Instrum.* **73**, 3988 (2002).

183. Ray, M. D., Nesnidal, M. P. & Socha, D. M. Optical detection of airborne ice crystals and liquid water droplets. *1st AIAA Atmos. Sp. Environ. Conf.* (2009). doi:10.2514/6.2009-3863

184. Prileszky, T. A. & Furst, E. M. Crystallization Kinetics of Partially Crystalline Emulsion Droplets in a Microfluidic Device. *Langmuir* **32**, 5141–5146 (2016).

185. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning (Adaptive Computation and Machine Learning series)*. (The MIT Press, 2016).

186. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).

187. Hindmarsh, J. P., Russell, A. B. & Chen, X. D. Experimental and numerical analysis of the temperature transition of a suspended freezing water droplet. *Int. J. Heat Mass Transf.* **46**, 1199–1213 (2003).

188. Graeber, G., Schutzius, T. M., Eghlidi, H. & Poulikakos, D. Spontaneous self-dislodging of freezing water droplets and the role of wettability. *Proc. Natl. Acad. Sci.* **114**, 11040–11045 (2017).

189. Marín, A. G., Enríquez, O. R., Brunet, P., Colinet, P. & Snoeijer, J. H. Universality of Tip Singularity Formation in Freezing Water Drops. *Phys. Rev. Lett.* **113**, 054301 (2014).

190. Ching, T. M. & Slabaugh, W. H. X-ray diffraction analysis of ice crystals in coniferous pollen. *Cryobiology* **2**, 321–327 (1966).

191. Schnell, R. C. & Vali, G. Atmospheric ice nuclei from decomposing vegetation. *Nature* **236**, 163–165 (1972).

192. Rosinski, J., Haagenson, P. L., Nagamoto, C. T. & Parungo, F. Ice-forming nuclei of maritime origin. *J. Aerosol Sci.* **17**, 23–46 (1986).

193. Matthias-Maser, S. & Jaenicke, R. The size distribution of primary biological aerosol particles with radii > 0.2 μm in an urban/rural influenced region. *Atmos. Res.* **39**, 279–286 (1995).

194. Jaenicke, R. Abundance of cellular material and proteins in the atmosphere. *Science (80-.).* **308**, 73 (2005).

195. Andreae, M. O. & Crutzen, P. J. Atmospheric aerosols: Biogeochemical sources and role in atmospheric chemistry. *Science (80-.).* **276**, 1052–1058 (1997).

196. Tobo, Y. *et al.* Biological aerosol particles as a key determinant of ice nuclei populations in a forest ecosystem. *J. Geophys. Res. Atmos.* **118**, 10,100-10,110 (2013).

197. Cascajo-Castresana, M., David, R. O., Iriarte-Alonso, M. A., Bittner, A. M. & Marcolli, C. Protein aggregates nucleate ice: The example of apoferritin. *Atmos. Chem. Phys.* **20**, 3291–3315 (2020).

198. DeMott, P. J. *et al.* Ice nucleation by particles containing long-chain fatty acids of relevance to freezing by sea spray aerosols. *Environ. Sci. Process. Impacts* **20**, 1559–1569 (2018).

199. Dreischmeier, K., Budke, C., Wiehemeier, L., Kottke, T. & Koop, T. Boreal pollen contain ice-nucleating as well as ice-binding 'antifreeze' polysaccharides. *Sci. Rep.* **7**, 1–13 (2017).

200. Hiranuma, N. *et al.* Ice nucleation by cellulose and its potential contribution to ice formation in clouds. *Nat. Geosci.* **8**, 273–277 (2015).

201. Akila, M., Priyamvada, H., Ravikrishna, R. & Gunthe, S. S. Characterization of bacterial diversity and ice-nucleating ability during different monsoon seasons over

a southern tropical Indian region. *Atmos. Environ.* **191**, 387–394 (2018).

202. Hu, W. *et al.* Biological Aerosol Particles in Polluted Regions. *Current Pollution Reports* **6**, 65–89 (2020).

203. Prenni, A. J. *et al.* Relative roles of biogenic emissions and saharan dust as ice nuclei in the amazon basin. *Nat. Geosci.* **2**, 402–405 (2009).

204. Mason, R. H. *et al.* Ice nucleating particles at a coastal marine boundary layer site: Correlations with aerosol type and meteorological conditions. *Atmos. Chem. Phys.* **15**, 12547–12566 (2015).

205. Wright, T. P., Hader, J. D., McMeeking, G. R. & Petters, M. D. High relative humidity as a trigger for widespread release of ice nuclei. *Aerosol Sci. Technol.* **48**, i–v (2014).

206. Bigg, E. K., Soubeyrand, S. & Morris, C. E. Persistent after-effects of heavy rain on concentrations of ice nuclei and rainfall suggest a biological cause. *Atmos. Chem. Phys.* **15**, 2313–2326 (2015).

207. Prenni, A. J. *et al.* The impact of rain on ice nuclei populations at a forested site in Colorado. *Geophys. Res. Lett.* **40**, 227–231 (2013).

208. Huffman, J. A. *et al.* High concentrations of biological aerosol particles and ice nuclei during and after rain. *Atmos. Chem. Phys.* **13**, 6151–6164 (2013).

209. Tobo, Y. *et al.* Organic matter matters for ice nuclei of agricultural soil origin.

Atmos. Chem. Phys. **14**, 8521–8531 (2014).

210. Suski, K. J. *et al.* Agricultural harvesting emissions of ice-nucleating particles. *Atmos. Chem. Phys.* **18**, 13755–13771 (2018).

211. Christner, B. C. *et al.* Geographic, seasonal, and precipitation chemistry influence on the abundance and activity of biological ice nucleators in rain and snow. *Proc. Natl. Acad. Sci. U. S. A.* **105**, 18854–18859 (2008).

212. Delort, A. M. *et al.* A short overview of the microbial population in clouds: Potential roles in atmospheric chemistry and nucleation processes. *Atmos. Res.* **98**, 249–260 (2010).

213. Joly, M. *et al.* Quantification of ice nuclei active at near 0 °c temperatures in low-altitude clouds at the Puy de Dôme atmospheric station. *Atmos. Chem. Phys.* **14**, 8185–8195 (2014).

214. Petters, M. D. & Wright, T. P. Revisiting ice nucleation from precipitation samples. *Geophys. Res. Lett.* **42**, 8758–8766 (2015).

215. Gute, E. & Abbatt, J. P. D. Ice nucleating behavior of different tree pollen in the immersion mode. *Atmos. Environ.* **231**, 117488 (2020).

216. Gute, E., David, R. O., Kanji, Z. A. & Abbatt, J. P. D. Ice Nucleation Ability of Tree Pollen Altered by Atmospheric Processing. *ACS Earth Sp. Chem.* **4**, 2312–2319 (2020).

217. Diehl, K., Matthias-Maser, S., Jaenicke, R. & Mitra, S. K. The ice nucleating ability of pollen: Part II. Laboratory studies in immersion and contact freezing modes. *Atmos. Res.* **61**, 125–133 (2002).

218. Fröhlich-Nowoisky, J. *et al.* Ice nucleation activity in the widespread soil fungus Mortierella alpina. *Biogeosciences* **12**, 1057–1071 (2015).

219. Iannone, R., Chernoff, D. I., Pringle, A., Martin, S. T. & Bertram, A. K. The ice nucleation ability of one of the most abundant types of fungal spores found in the atmosphere. *Atmos. Chem. Phys.* **11**, 1191–1201 (2011).

220. Huang, S. *et al.* Overview of biological ice nucleating particles in the atmosphere. *Environment International* **146**, (2021).

221. Maki, L. R., Galyan, E. L., Chang-Chien, M. M. & Caldwell, D. R. Ice nucleation induced by pseudomonas syringae. *Appl. Microbiol.* **28**, 456–459 (1974).

222. DeLeon-Rodriguez, N. *et al.* Microbiome of the upper troposphere: Species composition and prevalence, effects of tropical storms, and atmospheric implications. *Proc. Natl. Acad. Sci. U. S. A.* **110**, 2575–2580 (2013).

223. Murray, B. J., O'sullivan, D., Atkinson, J. D. & Webb, M. E. Ice nucleation by particles immersed in supercooled cloud droplets. *Chem. Soc. Rev.* **41**, 6519–6554 (2012).

224. Garnham, C. P., Campbell, R. L., Walker, V. K. & Davies, P. L. Novel dimeric β-helical model of an ice nucleation protein with bridged active sites. *BMC Struct.*

Biol. **11**, 36 (2011).

225. Graether, S. P. & Jia, Z. Modeling Pseudomonas syringae ice-nucleation protein as a β-helical protein. *Biophys. J.* **80**, 1169–1173 (2001).

226. Graether, S. P. *et al.* β-Helix structure and ice-binding properties of a hyperactive antifreeze protein from an insect. *Nature* **406**, 325–328 (2000).

227. Liu, F. *et al.* The Role of Protein Loss and Denaturation in Determining Outcomes of Heating, Cryotherapy, and Irreversible Electroporation on Cardiomyocytes. *J. Biomech. Eng.* **140**, (2018).

228. Shao, Q. *et al.* Engineering T cell response to cancer antigens by choice of focal therapeutic conditions. *Int. J. Hyperth.* **36**, 130–138 (2019).

229. Efimova, Y. M., Haemers, S., Wierczinski, B., Norde, W. & Well, A. A. van. Stability of globular proteins in H2O and D2O. *Biopolymers* **85**, 264–273 (2007).

230. Tobo, Y. An improved approach for measuring immersion freezing in large droplets over a wide temperature range. *Sci. Rep.* **6**, 1–9 (2016).

231. Polen, M., Lawlis, E. & Sullivan, R. C. The unstable ice nucleation properties of Snomax® bacterial particles. *J. Geophys. Res. Atmos.* **121**, 11666–11678 (2016).

232. Wex, H. *et al.* Intercomparing different devices for the investigation of ice nucleating particles using Snomax® as test substance. *Atmos. Chem. Phys.* **15**, 1463–1485 (2015).

233. Beydoun, H., Polen, M. & Sullivan, R. C. A new multi-component heterogeneous ice nucleation model and its application to Snomax bacterial particles and a Snomax-illte mineral particle mixture. *Atmos. Chem. Phys. Discuss.* 1–29 (2017). doi:10.5194/acp-2017-372

234. Vali, G. Quantitative Evaluation of Experimental Results an the Heterogeneous Freezing Nucleation of Supercooled Liquids. *J. Atmos. Sci.* **28**, 402–409 (1971).

235. Turner, M. A., Arellano, F. & Kozloff, L. M. Three separate classes of bacterial ice nucleation structures. *J. Bacteriol.* **172**, 2521–2526 (1990).

236. Schwidetzky, R. *et al.* Inhibition of Bacterial Ice Nucleators Is Not an Intrinsic Property of Antifreeze Proteins. *J. Phys. Chem. B* **124**, 4889–4895 (2020).

237. Warren, G. & Corotto, L. The consensus sequence of ice nucleation proteins from Erwinia herbicola, Pseudomonas fluorescens and Pseudomonas syringae. *Gene* **85**, 239–242 (1989).

238. Davies, P. L. Ice-binding proteins: A remarkable diversity of structures for stopping and starting ice growth. *Trends in Biochemical Sciences* **39**, 548–555 (2014).

239. Li, Q. *et al.* Molecular characterization of an ice nucleation protein variant (InaQ) from Pseudomonas syringae and the analysis of its transmembrane transport activity in Escherichia coli. *Int. J. Biol. Sci.* **8**, 1097–1108 (2012).

240. Roeters, S. J. *et al.* Ice-nucleating proteins are activated by low temperatures to control the structure of interfacial water. *Nat. Commun.* **12**, 1–9 (2021).

241. Lukas, M. *et al.* Interfacial Water Ordering Is Insufficient to Explain Ice-Nucleating Protein Activity. *J. Phys. Chem. Lett.* 218–223 (2021). doi:10.1021/acs.jpclett.0c03163

242. Lewis, E. R. & Schwartz, S. E. Sea salt aerosol production: Mechanisms, methods, measurements and models—A critical review. in *Geophysical Monograph Series* **152**, 1–408 (American Geophysical Union, 2004).

243. Cochran, R. E., Ryder, O. S., Grassian, V. H. & Prather, K. A. Sea spray aerosol: The chemical link between the oceans, atmosphere, and climate. *Accounts of Chemical Research* **50**, 599–604 (2017).

244. Brooks, S. D. & Thornton, D. C. O. Marine aerosols and clouds. *Ann. Rev. Mar. Sci.* **10**, 289–313 (2018).

245. Fall, R. & Schnell, R. C. Association of an ice-nucleating pseudomonad with cultures of the marine dinoflagellate, Heterocapsa niei. *J. Mar. Res.* **43**, 257–265 (1985).

246. Irish, V. E. *et al.* Revisiting properties and concentrations of ice-nucleating particles in the sea surface microlayer and bulk seawater in the Canadian Arctic during summer. *Atmos. Chem. Phys.* **19**, 7775–7787 (2019).

247. Wilson, T. W. *et al.* A marine biogenic source of atmospheric ice-nucleating particles. *Nature* **525**, 234–238 (2015).

248. Irish, V. E. *et al.* Ice-nucleating particles in Canadian Arctic sea-surface microlayer

and bulk seawater. *Atmos. Chem. Phys.* **17**, 10583–10595 (2017).

249. Rosinski, J. *et al.* Ice-forming nuclei in air masses over the Gulf of Mexico. *J. Aerosol Sci.* **19**, 539–551 (1988).

250. Schnell, R. C. Ice Nuclei in Seawater, Fog Water and Marine Air off the Coast of Nova Scotia: Summer 1975. *J. Atmos. Sci.* **34**, 1299–1305 (1977).

251. Schnell, R. C. & Vali, G. Freezing nuclei in marine waters. *Tellus* **27**, 321–323 (1975).

252. Parker, L. V, Sullivan, C. W., Forest, T. W. & Ackley, S. F. Ice nucleation activity of antarctic marine microorganisms. *Antarct. J.* **20**, 126–127 (1985).

253. Creamean, J. M. *et al.* Ice Nucleating Particles Carried From Below a Phytoplankton Bloom to the Arctic Atmosphere. *Geophys. Res. Lett.* **46**, 8572–8581 (2019).

254. Wilbourn, E. K. *et al.* Ice Nucleation by Marine Aerosols Over the North Atlantic Ocean in Late Spring. *J. Geophys. Res. Atmos.* **125**, (2020).

255. Gong, X. *et al.* Characterization of aerosol particles at Cabo Verde close to sea level and at the cloud level – Part 2: Ice-nucleating particles in air, cloud and seawater. *Atmos. Chem. Phys.* **20**, 1451–1468 (2020).

256. Wolf, M. J. *et al.* A link between the ice nucleation activity and the biogeochemistry of seawater. *Atmos. Chem. Phys.* **20**, 15341–15356 (2020).

257. McCluskey, C. S. *et al.* A Dynamic Link between Ice Nucleating Particles Released

in Nascent Sea Spray Aerosol and Oceanic Biological Activity during Two Mesocosm Experiments. *J. Atmos. Sci.* **74**, 151–166 (2016).

258. Mccluskey, C. S. *et al.* A mesocosm double feature: Insights into the chemical makeup of marine ice nucleating particles. *J. Atmos. Sci.* **75**, 2405–2423 (2018).

259. Trenberth, K. E. & Fasullo, J. T. Simulation of present-day and twenty-first-century energy budgets of the southern oceans. *J. Clim.* **23**, 440–454 (2010).

260. Riemer, N., Ault, A. P., West, M., Craig, R. L. & Curtis, J. H. Aerosol Mixing State: Measurements, Modeling, and Impacts. *Rev. Geophys.* **57**, 187–249 (2019).

261. Reicher, N. *et al.* Size-dependent ice nucleation by airborne particles during dust events in the eastern Mediterranean. *Atmos. Chem. Phys.* **19**, 11143–11158 (2019).

262. Mitts, B. A. *et al.* Importance of Supermicron Ice Nucleating Particles in Nascent Sea Spray. *Geophysical Research Letters* **48**, e2020GL089633 (2021).

263. Zhang, C. *et al.* The effects of morphology, mobility size, and secondary organic aerosol (SOA) material coating on the ice nucleation activity of black carbon in the cirrus regime. *Atmos. Chem. Phys.* **20**, 13957–13984 (2020).

264. Hasenkopf, C. A. *et al.* Ice nucleation, shape, and composition of aerosol particles in one of the most polluted cities in the world: Ulaanbaatar, Mongolia. *Atmos. Environ.* **139**, 222–229 (2016).

265. Wagner, R. & Möhler, O. Heterogeneous ice nucleation ability of crystalline sodium

chloride dihydrate particles. *J. Geophys. Res. Atmos.* **118**, 4610–4622 (2013).

266. Wagner, R. *et al.* Heterogeneous Ice Nucleation Ability of NaCl and Sea Salt Aerosol Particles at Cirrus Temperatures. *J. Geophys. Res. Atmos.* **123**, 2841–2860 (2018).

267. Wagner, R. *et al.* Heterogeneous ice nucleation ability of aerosol particles generated from Arctic sea surface microlayer and surface seawater samples at cirrus temperatures. *Atmos. Chem. Phys. Discuss.* 1–45 (2021).

268. Chernoff, D. I. & Bertram, A. K. Effects of sulfate coatings on the ice nucleation properties of a biological ice nucleus and several types of minerals. *J. Geophys. Res. Atmos.* **115**, D20205 (2010).

269. Lee, H. D. *et al.* Physicochemical Mixing State of Sea Spray Aerosols: Morphologies Exhibit Size Dependence. *ACS Earth Sp. Chem.* **4**, 1604–1611 (2020).

270. Reicher, N., Segev, L. & Rudich, Y. The WeIzmann Supercooled Droplets Observation on a Microarray (WISDOM) and application for ambient dust. *Atmos. Meas. Tech.* **11**, 233–248 (2018).

271. Roy, P., House, M. L. & Dutcher, C. S. A Microfluidic Device for Automated High Throughput Detection of Ice Nucleation of Snomax®. *Micromachines* **12**, 296 (2021).

272. Roy, P., Liu, S. & Dutcher, C. S. Droplet Interfacial Tensions and Phase Transitions

Measured in Microfluidic Channels. *Annu. Rev. Phys. Chem.* **72**, 73–97 (2021).

273. Sauer, J. S. *et al.* The Sea Spray Chemistry and Particle Evolution Study (SeaSCAPE): Overview and Experimental Methods. *Environ. Sci. Process. Impacts*

274. Cunliffe, M. & Wurl, O. Sampling the Sea Surface Microlayer. in 255–261 (Springer, Berlin, Heidelberg, 2015). doi:10.1007/8623_2015_83

275. Barry, K. R. *et al.* Pragmatic protocols for working cleanly when measuring ice nucleating particles. *Atmos. Res.* **250**, 105419 (2021).

276. Hiranuma, N. *et al.* A comprehensive laboratory study on the immersion freezing behavior of illite NX particles: A comparison of 17 ice nucleation measurement techniques. *Atmos. Chem. Phys.* **15**, 2489–2518 (2015).

277. Agresti, A. & Coull, B. A. Approximate is better than "Exact" for interval estimation of binomial proportions. *Am. Stat.* **52**, 119–126 (1998).

278. Polen, M., Brubaker, T., Somers, J. & Sullivan, R. C. Cleaning up our water: Reducing interferences from nonhomogeneous freezing of 'pure' water in droplet freezing assays of ice-nucleating particles. *Atmos. Meas. Tech.* **11**, 5315–5334 (2018).

279. Wang, A., Freeman, J. J., Jolliff, B. L. & Chou, I. M. Sulfates on Mars: A systematic Raman spectroscopic study of hydration states of magnesium sulfates. *Geochim. Cosmochim. Acta* **70**, 6118–6135 (2006).

280. Ault, A. P. *et al.* Raman microspectroscopy and vibrational sum frequency generation spectroscopy as probes of the bulk and surface compositions of size-resolved sea spray aerosol particles. *Phys. Chem. Chem. Phys.* **15**, 6206–6214 (2013).

281. Schulz, H., Baranska, M. & Baranski, R. Potential of NIR-FT-Raman spectroscopy in natural carotenoid analysis. *Biopolymers* **77**, 212–221 (2005).

282. Cochran, R. E. *et al.* Molecular Diversity of Sea Spray Aerosol Particles: Impact of Ocean Biology on Particle Composition and Hygroscopicity. *Chem* **2**, 655–667 (2017).

283. Sofos, J. N. Antimicrobial effects of sodium and other ions in foods: a review. *J. Food Saf.* **6**, 45–78 (1984).

284. Ickes, L. *et al.* The ice-nucleating activity of Arctic sea surface microlayer samples and marine algal cultures. *Atmos. Chem. Phys.* **20**, 11089–11117 (2020).

285. Wang, X. *et al.* Microbial Control of Sea Spray Aerosol Composition: A Tale of Two Blooms. *ACS Cent. Sci.* **1**, 124–131 (2015).

286. Knopf, D. A., Alpert, P. A., Wang, B. & Aller, J. Y. Stimulation of ice nucleation by marine diatoms. (2010). doi:10.1038/NGEO1037

287. Wagstaff, B. A., Rejzek, M. & Field, R. A. Identification of a Kdn biosynthesis pathway in the haptophyte Prymnesium parvum suggests widespread sialic acid biosynthesis among microalgae. *J. Biol. Chem.* **293**, 16277–16290 (2018).

288. Perro, A. *et al.* Combining microfluidics and FT-IR spectroscopy: Towards spatially resolved information on chemical processes. *React. Chem. Eng.* **1**, 577–594 (2016).

289. Suea-Ngam, A., Srisa-Art, M. & Furutani, Y. PDMS-based microfluidic device for infrared-transmission spectro-electrochemistry. *Bull. Chem. Soc. Jpn.* **91**, 728–734 (2018).

290. Srisa-Art, M., Noblitt, S. D., Krummel, A. T. & Henry, C. S. IR-Compatible PDMS microfluidic devices for monitoring of enzyme kinetics. *Anal. Chim. Acta* **1021**, 95–102 (2018).

Appendix A

Static trap temperature simulations using COMSOL Multiphysics

A.1 Temperature of device for internal ambient temperature -10°C

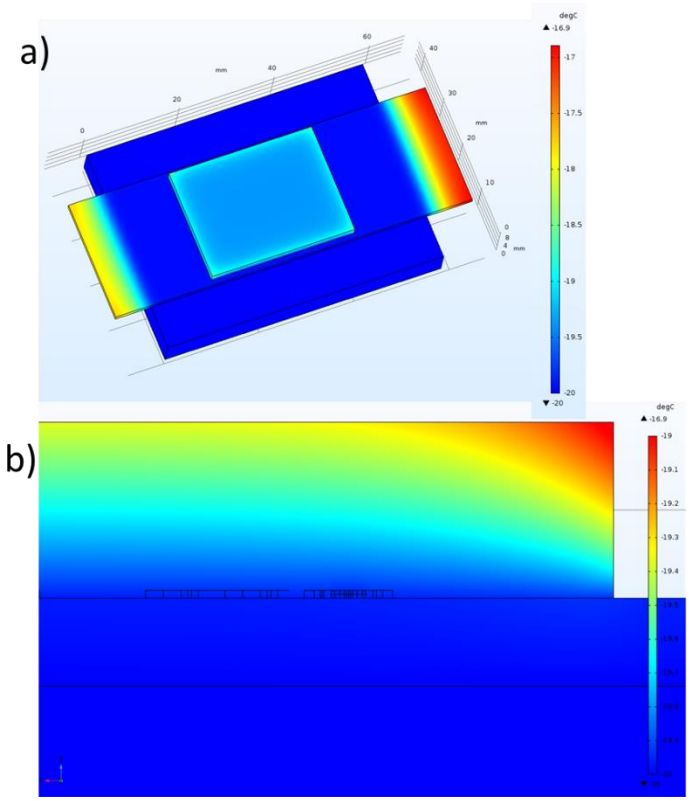


Figure A-1: Device temperature for cold plate temperature of -20°C and ambient air temperature of -10°C. (a) Device surface temperatures are well below -18.5°C throughout the region directly

above the cold plate. (b) The droplet wells are shown in a cross section. Note the different color legends.

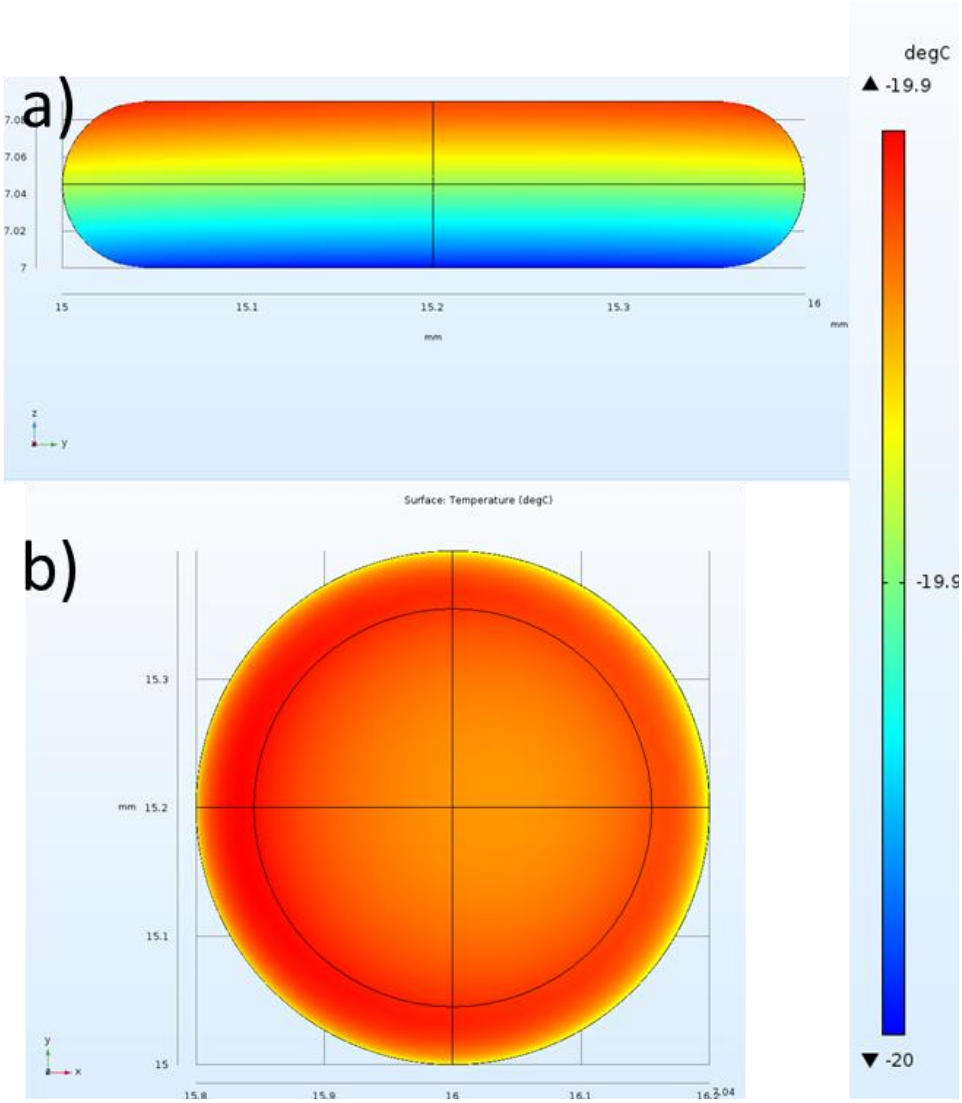


Figure A-2 Droplet temperature for cold plate temperature of 20°C and ambient air temperature of -10°C. (a) Side view. (b) Top view. Note that the droplet temperature variation is within 0.1°C from top to bottom.

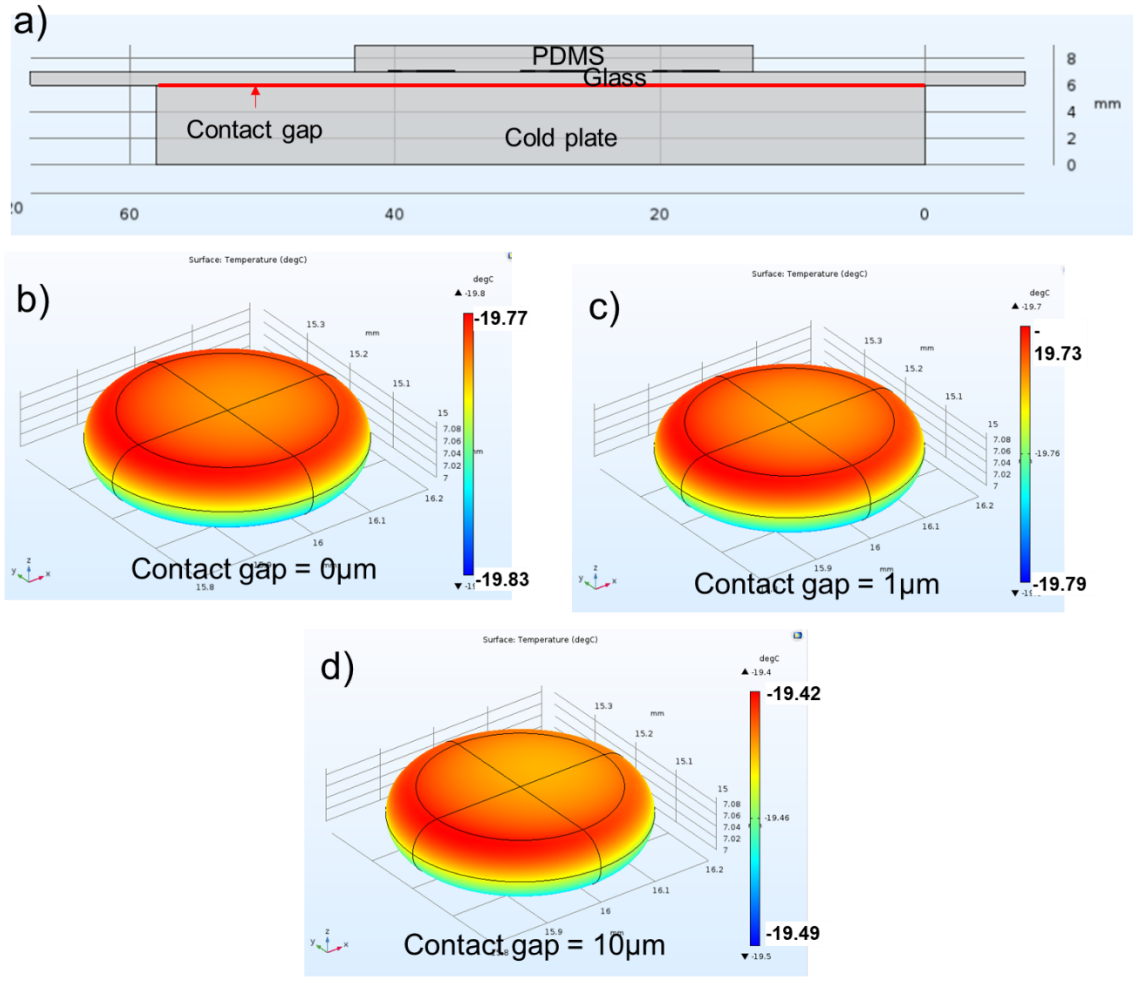


Figure A-3: Droplet temperatures for different contact gaps between the glass substrate of the device and the cold plate. The cold plate temperature is -20°C and the ambient air temperature is +20°C. (a) A schematic showing the region where contact gap was applied. (b) Droplet temperature for contact gap of 0µm. (c) Contact gap 1µm. (d) Contact gap 10µm. The 10µm value is typical for two mating metallic flat surfaces.

Appendix B

Droplet relative humidity calculation using thermodynamic models

B.1 Relative humidity calculation using E-AIM

For the RH calculation, 3-MGA was created as new a compound in E-AIM (http://www.aim.env.uea.ac.uk/aim/aim.php) model II for ammonium sulfate (AS) + 3-methylglutaric acid (3-MGA) systems with activity equations defined using the group contribution method from UNIFAC. Partitioning of all trace gases and formation of solid phases was disabled to allow the solution to reach supersaturation. 3-MGA was already available as a predefined compound in the AIOMFAC-web library of organics. The ternary solution density from the E-AIM output was used to convert the solution molarity concentration from Equation 2-2 to molality at the phase transition point. The relative humidity at this point was also recorded from the E-AIM output. The mass fractions of solutes were calculated from the solution density based on E-AIM output. Similar methodology was followed for the other systems described in Chapter 2. A step by step process is described below:

1. Use E-AIM density calculator to calculate the molar volume of pure liquid organic compound in cm^3/mol at 298.15 K.

(http://www.aim.env.uea.ac.uk/aim/density/density.php)

2. Recreate the same organic compound in the E-AIM library with UNIFAC activity equation selected and enter the functional groups according to the instructions there. Enter the molar volume from step 1 to allow solution density calculation later. (http://www.aim.env.uea.ac.uk/aim/phpmain/edit_compound.php?CreateCompound=)

3. Go to the organics page and add the newly created organic from step 2 into the current calculation.
(http://www.aim.env.uea.ac.uk/aim/phpmain/select_compounds.php)

4. Select the model you want to use (I, II, III or IV) and select "Parametrically varying relative humidity or total water". Then select column output. Enter the range of RH you want to calculate.

5. Enter moles/m^3 of each solute (including organics) in correct OIR for 1 m^3 of total sample volume.

6. Suppress partitioning of trace gases into vapor phase and formation of all solid phases by using the checkboxes. Then click Run.

7. You will notice among the rows, moles of all solutes remain constant, but moles of water changes, as does the relative humidity.

8. Calculate the molarity of all rows using the equation $\frac{moles_{organic}}{Total\ volume}$ which is equal to $\frac{moles_{organic}}{\frac{\sum MW \times moles}{Density}}$. In the second equation, the density is an E-AIM model output and provided in a column. Here MW refers to molecular weight. The density calculation is an approximation based on data from inorganic salt solutions and assuming that the apparent molar volumes of organic compounds, at all concentrations and temperatures, are equal to the molar volumes (at 298.15 K) input by the user.

9. Calculate the initial molarity of the organic in the solution by using the solution volume and the mass of dissolved organic used while preaparing the solution.

10. Get the final molarity of the organic at the LLPS point using the $MV = constant$ method.

11. Find the rows from the E-AIM output that matches molarity values from steps 9 and 10. These are the initial and final RH values of the droplet.

12. Also calculate the mass fraction of the organic using the equation $\frac{MW_{organic} \times moles_{organic}}{\sum MW \times moles}$. This can be used for calculating the RH from AIOMFAC.

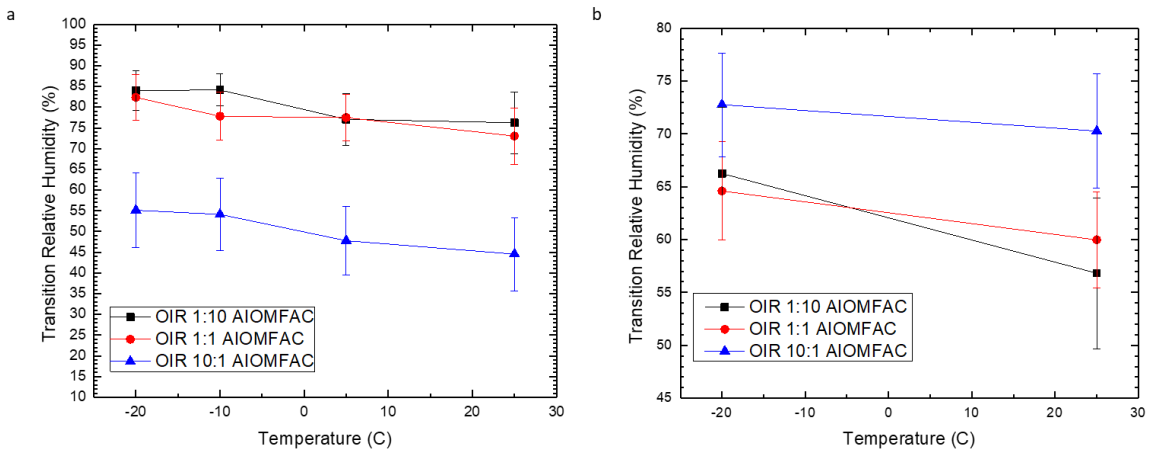


Figure B-1: Relative humidities of phase transition at different organic inorganic ratios and at different temperatures for (a) AS+3-MGA droplets and (b) NaCl+3-MGA calculated with AIOMFAC. For the OIR 1:10 and 1:1 phase transition refers to LLPS and for OIR 10:1 phase transition refers to efflorescence by nucleation of solid coating as there is no LLPS.

B.2 Relative humidity calculation using AIOMFAC

For the AIOMFAC calculations, mass fractions of the salt and the organic reported by E-AIM at phase transition points were given as inputs at the corresponding temperatures to AIOMFAC-web (https://aiomfac.lab.mcgill.ca/model.html). The results from the two models could be used to compare the supersaturated droplet equilibrium relative humidity at the phase transition point. We found that the model differences were on average anywhere between 2 to 5% RH for both the AS + 3-MGA and AS + NaCl droplets studied. This discrepancy is well within the margin of uncertainty of our experimental methods described in the main paper as well as the uncertainties within the models themselves related to the activity of group contribution, temperature dependence of activity

coefficients and the temperature dependence of the density of the ternary solutions. However, we note that both models show a comparable trend in the transition RH values as a function of temperature.

B.3 AIOMFAC model results for AS + 3-MGA and NaCl + 3-MGA droplets

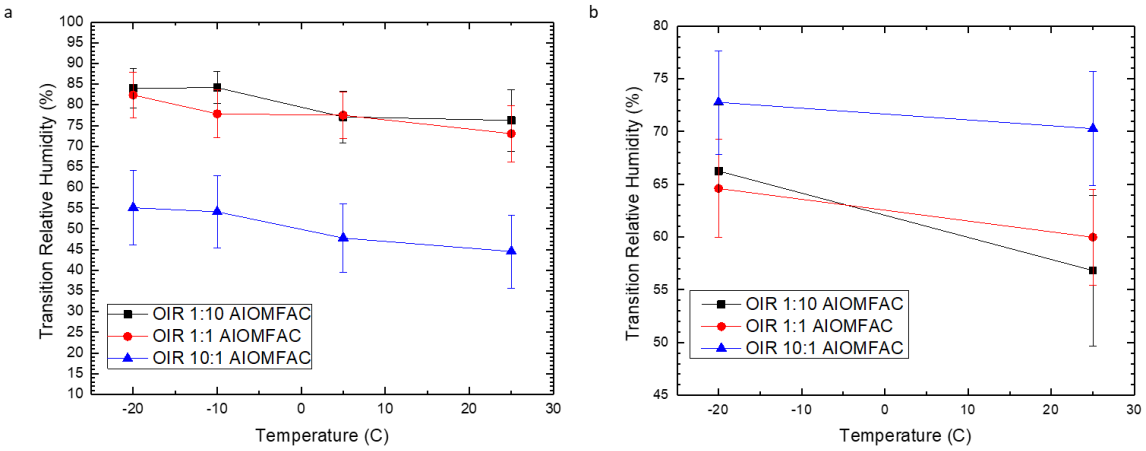


Figure B-2: Relative humidities of phase transition at different organic inorganic ratios and at different temperatures for (a) AS+3-MGA droplets and (b) NaCl+3-MGA calculated with AIOMFAC. For the OIR 1:10 and 1:1 phase transition refers to LLPS and for OIR 10:1 phase transition refers to efflorescence by nucleation of solid coating as there is no LLPS.

B.4 Transition RH values for all systems studied as a function of temperature

Table B-1 lists the RH values at phase transition for the AS + 3MGA systems calculated using both E-AIM and AIOMFAC models. The listed errors are the standard deviations of the measurements from 3 to 9 droplets across all conditions.

Table B-1: Transition RH values for all systems studied as a function of temperature

			Transition RH (E-AIM)				**Transition RH (AIOMFAC)**			
System	**OIR**	**Type of transition**	**25°C**	**5°C**	**-10°C**	**-20°C**	**25°C**	**5°C**	**-10°C**	**-20°C**
Ammonium sulfate + 3-MGA	1:10	LLPS	77.7% ± 7.6%	78.9% ± 6.4%	85.6% ± 4.1%	86.5% ± 4.8%	76.3% ± 7.5%	77.1% ± 6.3%	84.2% ± 3.9%	84.0% ± 4.7%
Ammonium sulfate + 3-MGA	1:1	LLPS	76.6% ± 6.8%	80.7% ± 5.7%	80.0% ± 5.8%	85.3% ± 5.7%	73.0% ± 6.8%	77.5% ± 5.6%	77.8% ± 5.8%	82.4% ± 5.6%
Ammonium sulfate + 3-MGA	10:1	Efflorescence (Coating formation)	39.4% ± 8.9%	43.6% ± 8.3%	51.7% ± 8.9%	53.1% ± 9.1%	44.6% ± 8.9%	47.8% ± 8.3%	54.1% ± 8.8%	53.1% ± 9.0%
NaCl + 3-MGA	1:10	Efflorescence (Single crystal)	57.9% ± 7.2%	-	-	67.2% ± 6.3%	56.8% ± 7.1%	-	-	66.3% ± 6.3%
NaCl + 3-MGA	1:1	Efflorescence (Single crystal)	62.1% ± 4.5%	-	-	67.3% ± 8.9%	60.0% ± 4.4%	-	-	64.6% ± 4.7%
NaCl + 3-MGA	10:1	Efflorescence (Multiple crystals)	69.0% ± 5.4%	-	-	71.5% ± 4.8%	70.3% ± 5.4%	-	-	72.8% ± 4.9%

Appendix C

PRTD fabrication protocol and operational LabVIEW codes

C.1 PRTD fabrication protocol

C.1.1 Photoresist coating

Glass slide cleaning (Keller cleanroom)

1. Piranha clean glass slides or wafers. 5:1 H2SO4:H2O2 bath @ 120C. For slides, use Keller bay 3 wet bench. For wafers use Keller Bay 1 Piranha clean bench. Bay 3 drains can be used for dumping liquid, Bay 1 just leave the bath as it is (follow instructions on bench to prepare the bath in bay 1, i.e. fill H2SO4 in the bath upto the mark indicated, then add 20ml of H2O2. In Bay 1 start heating up the bath to 120C. In Bay 3 set the hotplate to 120C. After a few minutes when the temperature reaches close to 120C, put wafers/slides into boats with attached handle and slowly lower into the baths. Keep in the bath for 10-15 minutes.

2. Use SRD1 in Bay 1 to clean wafers. Use water rinse and blow dry with N2 for Bay 3 glass slides. Bring cleaned slides/wafers to Bay 2. Use hot plate at 120C for 5 minutes to dry completely.

Spin coat positive photoresist

1. Soak in Bay 2 HMDS vapor chamber for 3 minutes.

2. Spin AZ1518 3300 rpm for 30 seconds at 3000 rpm/s ramp to get 2um coating thickness.

Pre-bake

1. Use hotplate to bake at 100C for 50 seconds for AZ1518.

Exposure

1. Use MA6 mask aligner in Bay 2.

2. Hard Ct – contact mode. 9 seconds exposure on MA6-P mask aligner.

Develop

1. Use Microposit 351 for 40 seconds to develop.

2. Rinse with DI water and blow dry with N2 gun.

Post Bake

1. 115C for 50 seconds on hotplate.

C.1.2 Metal coating

AJA I sputterer in Keller Bay 3

1. Enable AJA I and also the Keller Chase 2 AJA I Gold Target. Get the Platinum target from the gold target box in chase 2. Measure weight and log it.

2. Go to Bay 3 AJA I and login with username apex. Use PIN 0554 (Priyatanu). USE SOP FOR DETERMINING WHEN VENTING IS DONE.

3. Check if Ti is in Gun 4. If not, retrieve the Ti target and replace it in the gun.

4. Raise hoist and lock in place. Unscrew the gun chimney. Unscrew the target. Replace with Pt. This is most likely needed for Gun 2.

5. Unlock the hoist. Load samples into the plate inside the main chamber. If plate is not into main chamber, push in prongs from the side chamber. Lower hoist down onto seal.

6. Pumpdown the system. Use SOP to tell when pumpdown is completed.

7. Use the motor controls to move the plate to down position. Use dial to move from Jog to Run, set speed 40.

8. Use the program Gun_42_Ti_Pt_40_600sec to start the process.

C.1.3 Lift Off Lithography

Soak in acetone in a beaker overnight in the CFMF lab. Next day use acetone loaded spray can to spray flecks off. Rub with acetone soaked kimwipe to get rid of the rest of the photoresist coated metal.

C.1.4 Anneal PRTD arrays (Next day in Keller cleanroom)

1. Set flow of purge N_2 to 30 sccm.

2. Set temperature to 500°C, leave samples in for 12 hours.

C.2 PRTD resistance measurement LabVIEW code

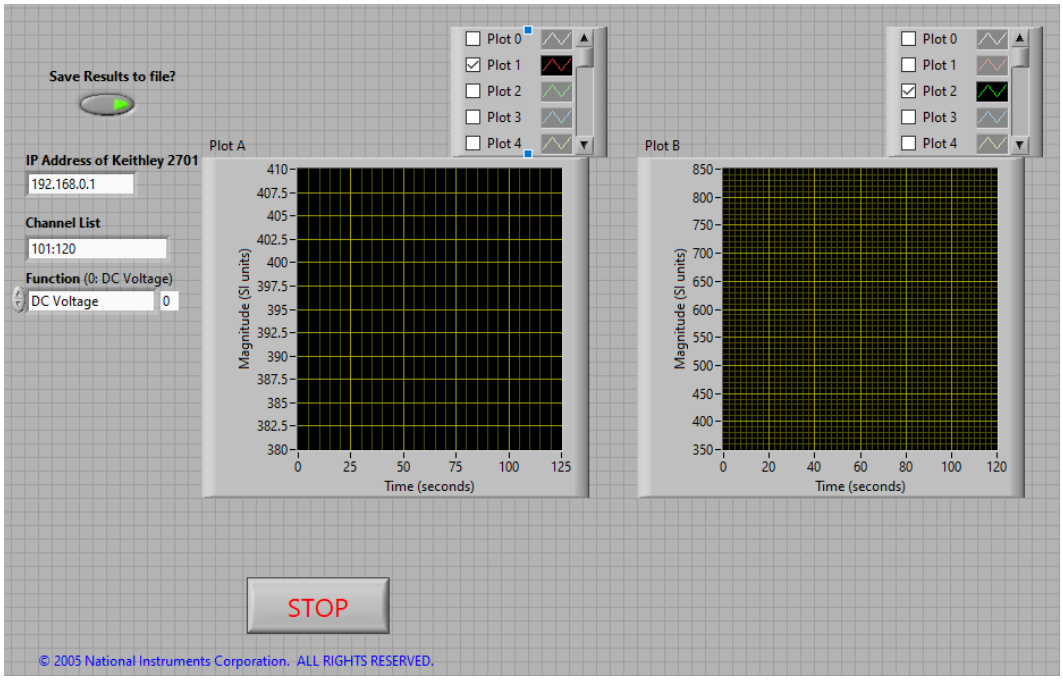


Figure C-1: The front panel of the LabVIEW code for collecting data from all 20 channels of the digital multimeter (DMM) at the same time.

The LabVIEW VI front panel created to collect data from all 19 PRTDs in the array as well as the extra reference resistor is shown in Figure C-1: The front panel of the LabVIEW code for collecting data from all 20 channels of the digital multimeter (DMM) at the same time. The IP address of the DMM has to be set using its buttons. The same IP address must be entered. For the channel list, 101 to 120 should be entered and the

measurement will be taken with DC voltage. Two separate plots can be used to monitor real time measurements of any number of channels, selected using the checkboxes.

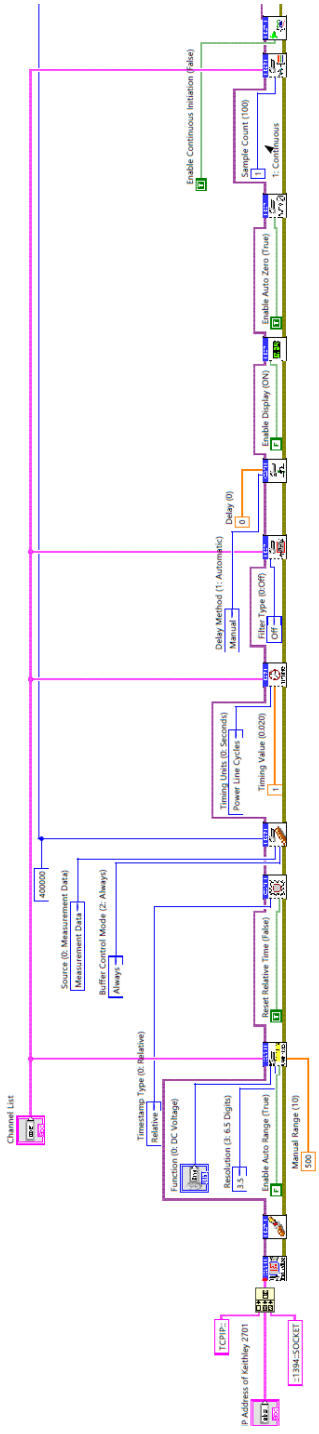


Figure C-2: Left half of the code, shows the instrument settings and initialization part, using the Keithley supplied LabVIEW drivers for the digital multimeter.

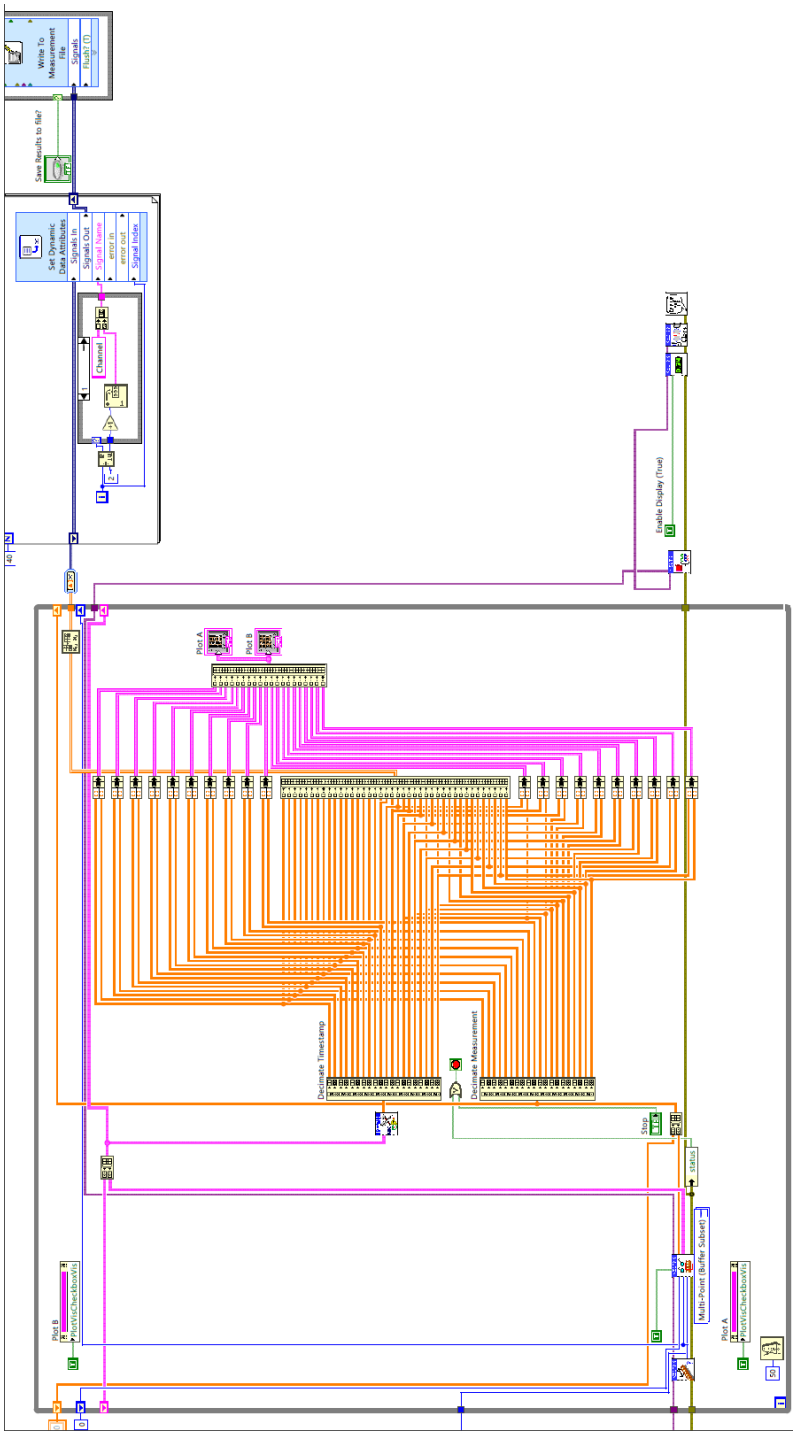


Figure C-3: Right half of the code. This shows a while loop with the necessary wiring to parse the data from all 40 channels into a coherent time series which is both plotted to a graph at runtime and written to an Excel file upon termination of the program.

The block diagram of the VI is provided in Figure C-2: Left half of the code, shows the instrument settings and initialization part, using the Keithley supplied LabVIEW drivers for the digital multimeter and Figure C-3: Right half of the code. This shows a while loop with the necessary wiring to parse the data from all 40 channels into a coherent time series which is both plotted to a graph at runtime and written to an Excel file upon termination of the program. The code was too wide to fit into a single figure, it has been broken up into a left half and a right half. This VI essentially opens a communication session with the multimeter using the IP address provided by the user. Then it proceeds to initializing its memory and preparing to read the data before moving into the while loop. Here it loops through and records data from its buffer, then parses it into different chunks and stores them in memory during the while loop execution i.e. the running of the program. When the user clicks the stop button, it dumps the contents of its parsed memory into an excel file with alternate columns of time and voltage data. i.e. two columns for each measurement channel.

Appendix D

MATLAB code for droplet freezing detection with polarized light

D.1 Polarized freezing detection code usage instructions

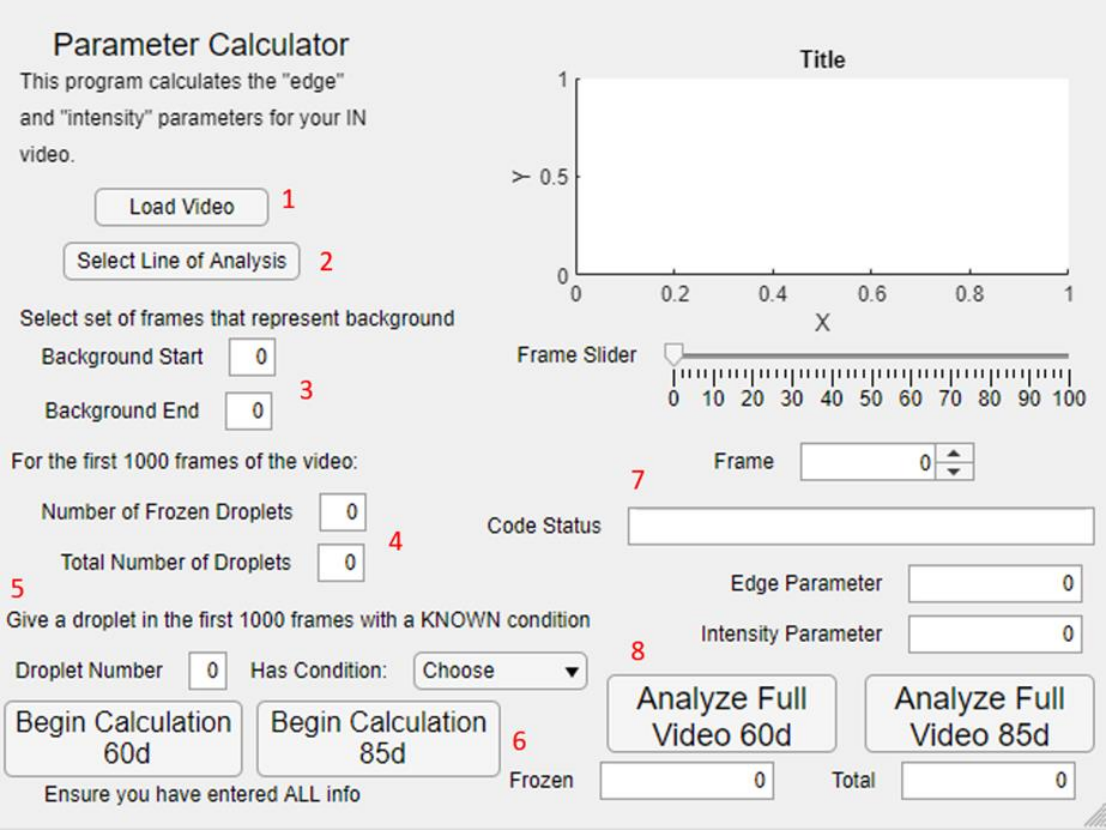


Figure D-1: Front end user interface for the polarized droplet freezing detection code.

The MATLAB app for the polarized freezing detection code is included in this appendix.

D.1.1 Instructions

1. Click the “Load Video” button and select your video. Be sure the file is in AVI format.

2. Click the “Select Line of Analysis” button. A figure will pop up showing the first frame of your video. Click once where you would like your Y axis line of analysis to begin and click again were you would like it to end. The tool will not stop drawing a line after this. Press enter to confirm your line selection and exit the figure.

3. Under “Select set of frames that represent background”, choose a range of at least 20 consecutive frames in which the only the background is in the line of analysis. Type in the number of the first frame under “Background Start” and the number of the last frame under “Background End”.

4. Under “For the first 1000 frames of the video”, type in the number of frozen droplets that pass through the line of analysis completely in the first 1000 frames of the video and the total number of droplets that pass through the line of analysis completely in the first 1000 frames of the video.

5. Under “Give a droplet in the first 1000 frames with a KNOWN condition”, choose a droplet in the first 1000 frames of the video that you know to be either a liquid

droplet or a bright frozen droplet. Type in the number of this droplet as it appears in the video and select its state from the drop-down menu. For example, if the third droplet in the video is liquid, type "3" into "Droplet Number" and choose "Liquid" from the drop-down menu.

6. For a video taken at or close to 60° polarizer angle, click "Begin Calculation 60d". For a video taken at or close to 85° polarizer angle, click "Begin Calculation 85d".

7. Observe the "Code Status" bar for updates. You will see "Edge Parameter" change first, and then "Intensity Parameter", before the code finishes. If this step takes longer than 10min, try repeating steps 1 – 6 with a slightly different line of analysis.

8. Once the "Code Status" bar reads "Finished", click "Analyze Full Video 60d" for a 60° video or click "Analyze Full Video 85d" for an 85° video. The number of total droplets will populate first in "Total", followed by the number of frozen droplets in "Frozen". The code is finished when neither number changes for a period of 1min.

D.2 Polarized detection code

```
classdef parameter_calculator < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
UIFigure matlab.ui.Figure
AnalyzeFullVideo90dButton matlab.ui.control.Button
BeginCalculation90dButton matlab.ui.control.Button
TotalEditField matlab.ui.control.NumericEditField
TotalEditFieldLabel matlab.ui.control.Label
FrozenEditField matlab.ui.control.NumericEditField
```

```
        FrozenEditFieldLabel matlab.ui.control.Label
        AnalyzeFullVideo60dButton matlab.ui.control.Button
        Giveadropletinthefirst1000frameswithaKNOWNconditionLabel
matlab.ui.control.Label
        DropletNumberEditField matlab.ui.control.NumericEditField
        DropletNumberEditFieldLabel matlab.ui.control.Label
        CodeStatusEditField matlab.ui.control.EditField
        CodeStatusEditFieldLabel matlab.ui.control.Label
        IntensityParameterEditField_2 matlab.ui.control.NumericEditField
        IntensityParameterEditField_2Label matlab.ui.control.Label
        EdgeParameterEditField matlab.ui.control.NumericEditField
        EdgeParameterEditFieldLabel matlab.ui.control.Label
        BeginCalculation60dButton matlab.ui.control.Button
        EnsureyouhaveenteredALLinfoLabel matlab.ui.control.Label
        NumberofFrozenDropletsEditField matlab.ui.control.NumericEditField
        NumberofFrozenDropletsEditFieldLabel matlab.ui.control.Label
        Forthefirst1000framesofthevideoLabel matlab.ui.control.Label
        TotalNumberofDropletsEditField matlab.ui.control.NumericEditField
        TotalNumberofDropletsEditFieldLabel matlab.ui.control.Label
        HasConditionDropDown matlab.ui.control.DropDown
        HasConditionDropDownLabel matlab.ui.control.Label
        BackgroundEndEditField matlab.ui.control.NumericEditField
        BackgroundEndEditFieldLabel matlab.ui.control.Label
        Selectsetoffframesthatrepresentbackground Label matlab.ui.control.Label
        BackgroundStartEditField matlab.ui.control.NumericEditField
        BackgroundStartEditFieldLabel matlab.ui.control.Label
        SelectLineofAnalysisButton matlab.ui.control.Button
        LoadVideoButton matlab.ui.control.Button
        videoLabel matlab.ui.control.Label
        andintensityparametersforyourINLabel matlab.ui.control.Label
        ThisprogramcalculatestheedgeLabel matlab.ui.control.Label
        ParameterCalculatorLabel matlab.ui.control.Label
        FrameSpinner matlab.ui.control.Spinner
        FrameSpinnerLabel matlab.ui.control.Label
        FrameSlider matlab.ui.control.Slider
        FrameSliderLabel matlab.ui.control.Label
        UIAxes matlab.ui.control.UIAxes
        end

        properties (Access = private)
        v; % Global variable to load the video file into
        framenumber;
        setdrop;
        setfro;
        initialguess;
```

```
noise;
beg;
last;
numFrames;
grayImage;
cx;
cy;
c;
xi2;
yi2;
liqdrop;
frodrop;
totdrop;
ar;
dropcase;
knowndrop;
intvar;
counter;
end
methods (Access = private)
function loadvideofun(app,videofile)
app.v = VideoReader(videofile);
% Count total number of frames in video
%numFrames = ceil(app.v.FrameRate*app.v.Duration*0.9);
%app.TotalFramesEditField.Value = numFrames;
app.numFrames = app.v.NumberOfFrames;
app.FrameSlider.Limits = [1 app.numFrames];
% Load 1st frame of the video as a grayimage
app.grayImage = read(app.v,1);
% Show current frame in UIAxes object
imshow(app.grayImage,'Parent', app.UIAxes);
end
function updateVideoframe(app,currentframe)
% Load 1st frame of the video as a grayimage
app.grayImage = read(app.v,currentframe);
% Show current frame in UIAxes object
imshow(app.grayImage,'Parent',app.UIAxes);
end
end

% Callbacks that handle component events
methods (Access = private)

% Button pushed function: LoadVideoButton
function LoadVideoButtonPushed(app, event)
```

```
% Select only avi and mp4 files
filterspec = {'*.avi;*.mp4;'};
% Show dialog box for loading file and load into variable
[vload, pload] = uigetfile(filterspec);
figure(app.UIFigure)
% Make sure user didn't cancel uigetfile dialog
if (ischar(pload))
fname = [pload vload];
loadvideofun(app, fname);
end
end

% Value changed function: FrameSlider
function FrameSliderValueChanged(app, event)
app.framenumber = round(app.FrameSlider.Value);
updateVideoframe(app,app.framenumber);
app.FrameSpinner.Value=app.framenumber;
end

% Value changed function: FrameSpinner
function FrameSpinnerValueChanged(app, event)
app.framenumber = round(app.FrameSpinner.Value);
updateVideoframe(app,app.framenumber);
app.FrameSlider.Value=app.framenumber;
end

% Button pushed function: SelectLineofAnalysisButton
function SelectLineofAnalysisButtonPushed(app, event)
app.grayImage = read(app.v,1);
if app.v.Width>1
figure(1)
imshow(app.grayImage)
[app.cx,app.cy,app.c,app.xi2,app.yi2] = improfile;
else
app.MessagesEditField.Value='Error: Video frame contains only 1 pixel';
end
end

% Value changed function: TotalNumberofDropletsEditField
function TotalNumberofDropletsEditFieldValueChanged(app, event)
app.setdrop= app.TotalNumberofDropletsEditField.Value;
end

% Value changed function: NumberofFrozenDropletsEditField
function NumberofFrozenDropletsEditFieldValueChanged(app, event)
```

```
app.setfro = app.NumberofFrozenDropletsEditField.Value;
end

% Value changed function: HasConditionDropDown
function HasConditionDropDownValueChanged(app, event)
if strcmpi(app.HasConditionDropDown.Value, 'Liquid')
app.dropcase=1;
app.intvar=2;
elseif strcmpi(app.HasConditionDropDown.Value,'Frozen (Bright)')
app.dropcase=2;
app.intvar=2;
end
end

% Value changed function: BackgroundStartEditField
function BackgroundStartEditFieldValueChanged(app, event)
app.beg = app.BackgroundStartEditField.Value;
end

% Value changed function: BackgroundEndEditField
function BackgroundEndEditFieldValueChanged(app, event)
app.last = app.BackgroundEndEditField.Value;
end

% Button pushed function: BeginCalculation60dButton
function BeginCalculation60dButtonPushed(app, event)

i=app.beg;
while i<=app.last
frame1=read(app.v,i);
frameIa(i,:)=frame1(round(app.xi2(1)),round (app.yi2(1)):round (app.yi2(2)));
i=i+1;
end
avg=mean(frameIa);
app.noise=-min(avg);
a=1;
while a==1
i=1;
app.counter=1;
app.ar=[0 0];
app.totdrop=0;
while i<=1000
frame2=read(app.v,i);
frameI(i,:)=double(frame2(round(app.xi2(1)),round (app.yi2(1)):round
(app.yi2(2))))-avg;
```

```
minI(i)=min(frameI(i,:));
if minI(i)< -app.noise
dro=zeros(5,length(frameI(i,:)));
dro(:,:)=-Inf;
app.ar(app.counter,1)=i;
while min(dro')<-app.noise &i<1000
for g=1:5
frame2=read(app.v,i+g);
dro(g,:)=double(frame2(round(app.xi2(1)),round (app.yi2(1)):round
(app.yi2(2))))-avg;
end
i=i+1;
app.ar(app.counter,2)=i;
end
if app.ar(app.counter,2)-app.ar(app.counter,1)>20
app.totdrop=app.totdrop+1;
app.counter=app.counter+1;
else
end
elseif minI(i)> 2*-app.noise
app.totdrop=app.totdrop+1;
app.counter=app.counter+1;
end
i=i+1;
end
if app.totdrop<app.setdrop
app.noise=app.noise-0.001*(app.setdrop-app.totdrop);
app.CodeStatusEditField.Value = 'Undercounted, trying again';
app.EdgeParameterEditField.Value=app.noise;
elseif app.totdrop>app.setdrop
app.noise=app.noise+0.003*(app.totdrop-app.setdrop);
app.CodeStatusEditField.Value = 'Overcounted, trying again';
app.EdgeParameterEditField.Value=app.noise;
else
a=2;
end
end
app.EdgeParameterEditField.Value=app.noise;
app.CodeStatusEditField.Value = 'Finished Edge Parameter';
a=1;
while a==1
app.CodeStatusEditField.Value = 'Calculating intensity parameter';
app.frodrop=0;
top=app.ar(app.knowndrop,1);
bot=app.ar(app.knowndrop,2);
```

```
m=1;
for h=top:bot
frame3=read(app.v,h);
anaframe(m)=-(double(frame3(72,round(app.xi2(1))))-avg(72));
m=m+1;
end
[maxana,maxind]=max(anaframe);
maxloc=bot-maxind;
miniana=anaframe(round(maxind+5):length(anaframe));
[tr,endvarana]=(max(miniana));
if app.dropcase==1
for j=1:app.totdrop
for k=app.ar(j,1):app.ar(j,2)
frame4=read(app.v,k);
dropframe(j,k)=-(double(frame4(72,round(app.xi2(1))))-avg(72));
end
[maxdro,maxdroind]=max(dropframe(j,:));
mini=dropframe(j,round(maxdroind+5):app.ar(j,2));
[tr,endvar]=(max(mini));
if min(dropframe(j,maxdroind:app.ar(j,2)-
10))>min(anaframe(maxind:length(anaframe)-10))+app.intvar
app.frodrop=app.frodrop+1;
elseif min(dropframe(j,maxdroind:app.ar(j,2)-
10))<min(anaframe(maxind:length(anaframe)-10))-app.intvar
app.frodrop=app.frodrop+1;
else
end
end
elseif app.dropcase==2
for j=1:app.totdrop
for k=app.ar(j,1):app.ar(j,2)
frame4=read(app.v,k);
dropframe(j,k)=-(double(frame4(72,round(app.xi2(1))))-avg(72));
end
[maxdro,maxdroind]=max(dropframe(j,:));
mini=dropframe(j,round(maxdroind+5):app.ar(j,2));
[tr,endvar]=(max(mini));
if min(dropframe(j,maxdroind:app.ar(j,2)-(length(mini)-
endvar)))>min(anaframe(maxind:length(anaframe)-(length(miniana)-
endvarana)))+app.intvar
elseif min(dropframe(j, maxdroind:app.ar(j,2)-(length(mini)-
endvar)))>min(anaframe(maxind:length(anaframe)-(length(miniana)-
endvarana)))+2*app.intvar
app.frodrop=app.frodrop+1;
else
```

```
app.frodrop=app.frodrop+1;
end
end
end
if app.frodrop>app.setfro
app.intvar=app.intvar+0.04*abs(app.setfro-app.frodrop);
app.CodeStatusEditField.Value ='Overcounted frozen, trying again';
app.IntensityParameterEditField_2.Value=app.intvar;
elseif app.frodrop<app.setfro
app.intvar=app.intvar-0.05*abs(app.setfro-app.frodrop);
app.CodeStatusEditField.Value ='Undercounted frozen, trying again';
app.IntensityParameterEditField_2.Value=app.intvar;
else
a=2;
end
end
app.IntensityParameterEditField_2.Value=app.intvar;
app.CodeStatusEditField.Value ='Finished';
end

% Value changed function: DropletNumberEditField
function DropletNumberEditFieldValueChanged(app, event)
app.knowndrop=app.DropletNumberEditField.Value;
end

% Button pushed function: AnalyzeFullVideo60dButton
function AnalyzeFullVideo60dButtonPushed(app, event)
i=app.beg;
d=app.v.Duration;
f=app.v.FrameRate;
while i<=app.last
frame1=read(app.v,i);
frameIa(i,:)=frame1(round(app.xi2(1)),round (app.yi2(1)):round (app.yi2(2)));
i=i+1;
end
avg=mean(frameIa);
i=1;
app.counter=1;
app.ar=[0 0];
app.totdrop=0;
while i<=100000
frame2=read(app.v,i);
frameI(i,:)=double(frame2(round(app.xi2(1)),round (app.yi2(1)):round
(app.yi2(2))))-avg;
minI(i)=min(frameI(i,:));
```

```
        if minI(i)< -app.noise
        dro=zeros(5,length(frameI(i,:)));
        dro(:,:)=-Inf;
        app.ar(app.counter,1)=i;
        while min(dro')<-app.noise
        for g=1:5
        frame2=read(app.v,i+g);
        dro(g,:)=double(frame2(round(app.xi2(1)),round (app.yi2(1)):round
(app.yi2(2))))-avg;
        end
        i=i+1;
        app.ar(app.counter,2)=i;
        end
        if app.ar(app.counter,2)-app.ar(app.counter,1)>20
        app.totdrop=app.totdrop+1;
        app.TotalEditField.Value=app.totdrop;
        app.counter=app.counter+1;
        else
        end
        else
        end
        i=i+1;
        end
        app.frodrop=0;
        top=app.ar(app.knowndrop,1);
        bot=app.ar(app.knowndrop,2);
        m=1;
        for h=top:bot
        frame3=read(app.v,h);
        anaframe(m)=-(double(frame3(72,round(app.xi2(1))))-avg(72));
        m=m+1;
        end
        [maxana,maxind]=max(anaframe);
        maxloc=bot-maxind;
        miniana=anaframe(round(maxind+5):length(anaframe));
        [tr,endvarana]=(max(miniana));
        if app.dropcase==1
        for j=1:app.totdrop
        for k=app.ar(j,1):app.ar(j,2)
        frame4=read(app.v,k);
        dropframe(j,k)=-(double(frame4(72,round(app.xi2(1))))-avg(72));
        end
        [maxdro,maxdroind]=max(dropframe(j,:));
        mini=dropframe(j,round(maxdroind+5):app.ar(j,2));
        [tr,endvar]=(max(mini));
```

```
        if                                          min(dropframe(j,maxdroind:app.ar(j,2)-
10))>min(anaframe(maxind:length(anaframe)-10))+app.intvar
        app.frodrop=app.frodrop+1;
        app.FrozenEditField.Value=app.frodrop;
        elseif                                      min(dropframe(j,maxdroind:app.ar(j,2)-
10))<min(anaframe(maxind:length(anaframe)-10))-app.intvar
        app.frodrop=app.frodrop+1;
        app.FrozenEditField.Value=app.frodrop;
        else
        end
        end
        elseif app.dropcase==2
        for j=1:app.totdrop
        for k=app.ar(j,1):app.ar(j,2)
        frame4=read(app.v,k);
        dropframe(j,k)=-(double(frame4(72,round(app.xi2(1))))-avg(72));
        end
        [maxdro,maxdroind]=max(dropframe(j,:));
        mini=dropframe(j,round(maxdroind+5):app.ar(j,2));
        [tr,endvar]=(max(mini));
        if                          min(dropframe(j,maxdroind:app.ar(j,2)-(length(mini)-
endvar)))>min(anaframe(maxind:length(anaframe)-(length(miniana)-
endvarana)))+app.intvar
        elseif                      min(dropframe(j,maxdroind:app.ar(j,2)-(length(mini)-
endvar)))>min(anaframe(maxind:length(anaframe)-(length(miniana)-
endvarana)))+2*app.intvar
        app.frodrop=app.frodrop+1;
        app.FrozenEditField.Value=app.frodrop;
        else
        app.frodrop=app.frodrop+1;
        app.FrozenEditField.Value=app.frodrop;
        end
        end
        end
        end

        % Button pushed function: BeginCalculation90dButton
        function BeginCalculation90dButtonPushed(app, event)
        i=app.beg;
        while i<=app.last
        frame1=read(app.v,i);
        frameIa(i,:)=frame1(round(app.xi2(1)),round (app.yi2(1)):round (app.yi2(2)));
        i=i+1;
        end
        avg=mean(frameIa);
```

```
app.noise=2;
a=1;
while a==1
i=1;
app.counter=1;
app.ar=[0 0];
app.totdrop=0;
while i<=1000
frame2=read(app.v,i);
frameI(i,:)=-(double(frame2(72,round(app.xi2(1))))-avg(72));
if frameI(i)>app.noise
app.ar(app.counter,1)=i;
app.ar(app.counter,2)=i+20;
app.totdrop=app.totdrop+1;
i=i+50;
app.counter=app.counter+1;
else
end
i=i+1;
end
if app.totdrop<app.setdrop
app.noise=app.noise-0.01*(app.setdrop-app.totdrop);
app.CodeStatusEditField.Value = 'Undercounted, trying again';
app.EdgeParameterEditField.Value=app.noise;
elseif app.totdrop>app.setdrop
app.noise=app.noise+0.03*(app.totdrop-app.setdrop);
app.CodeStatusEditField.Value = 'Overcounted, trying again';
app.EdgeParameterEditField.Value=app.noise;
else
a=2;
end
end
app.EdgeParameterEditField.Value=app.noise;
app.CodeStatusEditField.Value = 'Finished Edge Parameter';
a=1;
while a==1
app.CodeStatusEditField.Value ='Calculating intensity parameter';
app.frodrop=0;
top=app.ar(app.knowndrop,1);
bot=app.ar(app.knowndrop,2);
m=1;
for h=top:bot
frame3=read(app.v,h);
anaframe(m)=-(double(frame3(72,round(app.xi2(1))))-avg(72));
m=m+1;
```

```
end
[maxana,maxind]=max(anaframe);
maxloc=bot-maxind;
miniana=anaframe(round(maxind+5):length(anaframe));
[tr,endvarana]=(max(miniana));
if app.dropcase==1
for j=1:app.totdrop
for k=app.ar(j,1):app.ar(j,2)
frame4=read(app.v,k);
dropframe(j,k)=-(double(frame4(72,round(app.xi2(1))))-avg(72));
end
[maxdro,maxdroind]=max(dropframe(j,:));
mini=dropframe(j,round(maxdroind+5):app.ar(j,2));
[tr,endvar]=(max(mini));
if min(dropframe(j,maxdroind:app.ar(j,2)-
10))>min(anaframe(maxind:length(anaframe)-10))+app.intvar
app.frodrop=app.frodrop+1;
elseif min(dropframe(j,maxdroind:app.ar(j,2)-
10))<min(anaframe(maxind:length(anaframe)-10))-app.intvar
app.frodrop=app.frodrop+1;
else
end
end
elseif app.dropcase==2
for j=1:app.totdrop
for k=app.ar(j,1):app.ar(j,2)
frame4=read(app.v,k);
dropframe(j,k)=-(double(frame4(72,round(app.xi2(1))))-avg(72));
end
[maxdro,maxdroind]=max(dropframe(j,:));
mini=dropframe(j,round(maxdroind+5):app.ar(j,2));
[tr,endvar]=(max(mini));
if min(dropframe(j,maxdroind:app.ar(j,2)-(length(mini)-
endvar)))>min(anaframe(maxind:length(anaframe)-(length(miniana)-
endvarana)))+app.intvar
elseif min(dropframe(j, maxdroind:app.ar(j,2)-(length(mini)-
endvar)))>min(anaframe(maxind:length(anaframe)-(length(miniana)-
endvarana)))+2*app.intvar
app.frodrop=app.frodrop+1;
else
app.frodrop=app.frodrop+1;
end
end
end
if app.frodrop+3<app.setfro
```

```
app.intvar=app.intvar-0.05*abs(app.setfro-app.frodrop);
app.CodeStatusEditField.Value ='Undercounted frozen, trying again';
app.IntensityParameterEditField_2.Value=app.intvar;
elseif app.frodrop>app.setfro
app.intvar=app.intvar+0.0004*abs(app.setfro-app.frodrop);
app.CodeStatusEditField.Value ='Overcounted frozen, trying again';
app.IntensityParameterEditField_2.Value=app.intvar;
elseif app.frodrop<app.setfro
app.intvar=app.intvar-0.0005*abs(app.setfro-app.frodrop);
app.CodeStatusEditField.Value ='Undercounted frozen, trying again';
app.IntensityParameterEditField_2.Value=app.intvar;
else
a=2;
end
end
app.IntensityParameterEditField_2.Value=app.intvar;
app.CodeStatusEditField.Value ='Finished';
end

% Button pushed function: AnalyzeFullVideo90dButton
function AnalyzeFullVideo90dButtonPushed(app, event)
i=app.beg;
while i<=app.last
frame1=read(app.v,i);
frameIa(i,:)=frame1(round(app.xi2(1)),round (app.yi2(1)):round (app.yi2(2)));
i=i+1;
end
avg=mean(frameIa);
i=1;
app.counter=1;
app.ar=[0 0];
app.totdrop=0;
while i<=100000
frame2=read(app.v,i);
frameI(i,:)=-(double(frame2(72,round(app.xi2(1))))-avg(72));
if frameI(i)>app.noise
app.ar(app.counter,1)=i;
app.ar(app.counter,2)=i+20;
app.totdrop=app.totdrop+1;
app.TotalEditField.Value= app.totdrop;
i=i+50;
app.counter=app.counter+1;
else
end
i=i+1;
```

```
        end
        app.frodrop=0;
        top=app.ar(app.knowndrop,1);
        bot=app.ar(app.knowndrop,2);
        m=1;
        for h=top:bot
        frame3=read(app.v,h);
        anaframe(m)=-(double(frame3(72,round(app.xi2(1))))-avg(72));
        m=m+1;
        end
        [maxana,maxind]=max(anaframe);
        maxloc=bot-maxind;
        miniana=anaframe(round(maxind+5):length(anaframe));
        [tr,endvarana]=(max(miniana));
        if app.dropcase==1
        for j=1:app.totdrop
        for k=app.ar(j,1):app.ar(j,2)
        frame4=read(app.v,k);
        dropframe(j,k)=-(double(frame4(72,round(app.xi2(1))))-avg(72));
        end
        [maxdro,maxdroind]=max(dropframe(j,:));
        mini=dropframe(j,round(maxdroind+5):app.ar(j,2));
        [tr,endvar]=(max(mini));
        if                                          min(dropframe(j,maxdroind:app.ar(j,2)-
10))>min(anaframe(maxind:length(anaframe)-10))+app.intvar
        app.frodrop=app.frodrop+1;
        app.FrozenEditField.Value=app.frodrop;
        elseif                                      min(dropframe(j,maxdroind:app.ar(j,2)-
10))<min(anaframe(maxind:length(anaframe)-10))-app.intvar
        app.frodrop=app.frodrop+1;
        app.FrozenEditField.Value=app.frodrop;
        else
        end
        end
        elseif app.dropcase==2
        for j=1:app.totdrop
        for k=app.ar(j,1):app.ar(j,2)
        frame4=read(app.v,k);
        dropframe(j,k)=-(double(frame4(72,round(app.xi2(1))))-avg(72));
        end
        [maxdro,maxdroind]=max(dropframe(j,:));
        mini=dropframe(j,round(maxdroind+5):app.ar(j,2));
        [tr,endvar]=(max(mini));
```

```
        if                              min(dropframe(j,maxdroind:app.ar(j,2)-(length(mini)-
endvar)))>min(anaframe(maxind:length(anaframe)-(length(miniana)-
endvarana)))+app.intvar
        elseif          min(dropframe(j,          maxdroind:app.ar(j,2)-(length(mini)-
endvar)))>min(anaframe(maxind:length(anaframe)-(length(miniana)-
endvarana)))+2*app.intvar
        app.frodrop=app.frodrop+1;
        app.FrozenEditField.Value=app.frodrop;
        else
        app.frodrop=app.frodrop+1;
        app.FrozenEditField.Value=app.frodrop;
        end
        end
        end
        end
        end

        % Component initialization
        methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

        % Create UIFigure and hide until all components are created
        app.UIFigure = uifigure('Visible', 'off');
        app.UIFigure.Position = [100 100 640 480];
        app.UIFigure.Name = 'UI Figure';

        % Create UIAxes
        app.UIAxes = uiaxes(app.UIFigure);
        title(app.UIAxes, 'Title')
        xlabel(app.UIAxes, 'X')
        ylabel(app.UIAxes, 'Y')
        app.UIAxes.PlotBoxAspectRatio = [2.51785714285714 1 1];
        app.UIAxes.Position = [295 284 330 168];

        % Create FrameSliderLabel
        app.FrameSliderLabel = uilabel(app.UIFigure);
        app.FrameSliderLabel.HorizontalAlignment = 'right';
        app.FrameSliderLabel.Position = [295 262 74 22];
        app.FrameSliderLabel.Text = 'Frame Slider';

        % Create FrameSlider
        app.FrameSlider = uislider(app.UIFigure);
```

```
        app.FrameSlider.ValueChangedFcn = createCallbackFcn(app,
@FrameSliderValueChanged, true);
        app.FrameSlider.Position = [390 271 227 3];

        % Create FrameSpinnerLabel
        app.FrameSpinnerLabel = uilabel(app.UIFigure);
        app.FrameSpinnerLabel.HorizontalAlignment = 'right';
        app.FrameSpinnerLabel.Position = [408 201 40 22];
        app.FrameSpinnerLabel.Text = 'Frame';

        % Create FrameSpinner
        app.FrameSpinner = uispinner(app.UIFigure);
        app.FrameSpinner.ValueChangedFcn = createCallbackFcn(app,
@FrameSpinnerValueChanged, true);
        app.FrameSpinner.Position = [463 201 100 22];

        % Create ParameterCalculatorLabel
        app.ParameterCalculatorLabel = uilabel(app.UIFigure);
        app.ParameterCalculatorLabel.FontSize = 18;
        app.ParameterCalculatorLabel.Position = [34 440 176 22];
        app.ParameterCalculatorLabel.Text = 'Parameter Calculator';

        % Create ThisprogramcalculatestheedgeLabel
        app.ThisprogramcalculatestheedgeLabel = uilabel(app.UIFigure);
        app.ThisprogramcalculatestheedgeLabel.Position = [15 419 305 22];
        app.ThisprogramcalculatestheedgeLabel.Text = 'This program calculates the
"edge"';

        % Create andintensityparametersforyourINLabel
        app.andintensityparametersforyourINLabel = uilabel(app.UIFigure);
        app.andintensityparametersforyourINLabel.Position = [15 398 205 22];
        app.andintensityparametersforyourINLabel.Text = 'and "intensity" parameters for
your IN';

        % Create videoLabel
        app.videoLabel = uilabel(app.UIFigure);
        app.videoLabel.Position = [15 377 44 22];
        app.videoLabel.Text = 'video. ';

        % Create LoadVideoButton
        app.LoadVideoButton = uibutton(app.UIFigure, 'push');
        app.LoadVideoButton.ButtonPushedFcn = createCallbackFcn(app,
@LoadVideoButtonPushed, true);
        app.LoadVideoButton.Position = [58 347 100 22];
        app.LoadVideoButton.Text = 'Load Video';
```

```
        % Create SelectLineofAnalysisButton
        app.SelectLineofAnalysisButton = uibutton(app.UIFigure, 'push');
        app.SelectLineofAnalysisButton.ButtonPushedFcn = createCallbackFcn(app,
@SelectLineofAnalysisButtonPushed, true);
        app.SelectLineofAnalysisButton.Position = [40 316 136 22];
        app.SelectLineofAnalysisButton.Text = 'Select Line of Analysis';

        % Create BackgroundStartEditFieldLabel
        app.BackgroundStartEditFieldLabel = uilabel(app.UIFigure);
        app.BackgroundStartEditFieldLabel.HorizontalAlignment = 'right';
        app.BackgroundStartEditFieldLabel.Position = [22 261 98 22];
        app.BackgroundStartEditFieldLabel.Text = 'Background Start';

        % Create BackgroundStartEditField
        app.BackgroundStartEditField = uieditfield(app.UIFigure, 'numeric');
        app.BackgroundStartEditField.ValueChangedFcn = createCallbackFcn(app,
@BackgroundStartEditFieldValueChanged, true);
        app.BackgroundStartEditField.Position = [135 261 27 22];

        % Create SelectsetoframesthatrepresentbackgroundLabel
        app.SelectsetoframesthatrepresentbackgroundLabel = uilabel(app.UIFigure);
        app.SelectsetoframesthatrepresentbackgroundLabel.Position = [15 283 258 22];
        app.SelectsetoframesthatrepresentbackgroundLabel.Text = 'Select set of frames
that represent background';

        % Create BackgroundEndEditFieldLabel
        app.BackgroundEndEditFieldLabel = uilabel(app.UIFigure);
        app.BackgroundEndEditFieldLabel.HorizontalAlignment = 'right';
        app.BackgroundEndEditFieldLabel.Position = [24 230 94 22];
        app.BackgroundEndEditFieldLabel.Text = 'Background End';

        % Create BackgroundEndEditField
        app.BackgroundEndEditField = uieditfield(app.UIFigure, 'numeric');
        app.BackgroundEndEditField.ValueChangedFcn = createCallbackFcn(app,
@BackgroundEndEditFieldValueChanged, true);
        app.BackgroundEndEditField.Position = [133 230 27 22];

        % Create HasConditionDropDownLabel
        app.HasConditionDropDownLabel = uilabel(app.UIFigure);
        app.HasConditionDropDownLabel.HorizontalAlignment = 'right';
        app.HasConditionDropDownLabel.Position = [142 81 84 22];
        app.HasConditionDropDownLabel.Text = 'Has Condition:';

        % Create HasConditionDropDown
```

```
        app.HasConditionDropDown = uidropdown(app.UIFigure);
        app.HasConditionDropDown.Items = {'Choose', 'Liquid', 'Frozen (Bright)', ''};
        app.HasConditionDropDown.ValueChangedFcn = createCallbackFcn(app,
@HasConditionDropDownValueChanged, true);
        app.HasConditionDropDown.Position = [241 81 100 22];
        app.HasConditionDropDown.Value = 'Choose';

        % Create TotalNumberofDropletsEditFieldLabel
        app.TotalNumberofDropletsEditFieldLabel = uilabel(app.UIFigure);
        app.TotalNumberofDropletsEditFieldLabel.HorizontalAlignment = 'right';
        app.TotalNumberofDropletsEditFieldLabel.Position = [33 143 138 22];
        app.TotalNumberofDropletsEditFieldLabel.Text = 'Total Number of Droplets';

        % Create TotalNumberofDropletsEditField
        app.TotalNumberofDropletsEditField = uieditfield(app.UIFigure, 'numeric');
        app.TotalNumberofDropletsEditField.ValueChangedFcn =
createCallbackFcn(app, @TotalNumberofDropletsEditFieldValueChanged, true);
        app.TotalNumberofDropletsEditField.Position = [186 143 27 22];

        % Create Forthefirst1000framesofthevideoLabel
        app.Forthefirst1000framesofthevideoLabel = uilabel(app.UIFigure);
        app.Forthefirst1000framesofthevideoLabel.Position = [10 201 206 22];
        app.Forthefirst1000framesofthevideoLabel.Text = 'For the first 1000 frames of the
video:';

        % Create NumberofFrozenDropletsEditFieldLabel
        app.NumberofFrozenDropletsEditFieldLabel = uilabel(app.UIFigure);
        app.NumberofFrozenDropletsEditFieldLabel.HorizontalAlignment = 'right';
        app.NumberofFrozenDropletsEditFieldLabel.Position = [22 172 150 22];
        app.NumberofFrozenDropletsEditFieldLabel.Text = 'Number of Frozen Droplets';

        % Create NumberofFrozenDropletsEditField
            app.NumberofFrozenDropletsEditField = uieditfield(app.UIFigure,
'numeric');
            app.NumberofFrozenDropletsEditField.ValueChangedFcn =
createCallbackFcn(app, @NumberofFrozenDropletsEditFieldValueChanged, true);
            app.NumberofFrozenDropletsEditField.Position = [187 172 27 22];

            % Create EnsureyouhaveenteredALLinfoLabel
            app.EnsureyouhaveenteredALLinfoLabel = uilabel(app.UIFigure);
            app.EnsureyouhaveenteredALLinfoLabel.Position = [29 10 187 22];
            app.EnsureyouhaveenteredALLinfoLabel.Text = 'Ensure you have entered
ALL info';

            % Create BeginCalculation60dButton
```

```
            app.BeginCalculation60dButton = uibutton(app.UIFigure, 'push');
            app.BeginCalculation60dButton.ButtonPushedFcn                    =
createCallbackFcn(app, @BeginCalculation60dButtonPushed, true);
            app.BeginCalculation60dButton.FontSize = 16;
            app.BeginCalculation60dButton.Position = [6 31 137 44];
            app.BeginCalculation60dButton.Text = {'Begin Calculation'; ' 60d'; ''};

            % Create EdgeParameterEditFieldLabel
            app.EdgeParameterEditFieldLabel = uilabel(app.UIFigure);
            app.EdgeParameterEditFieldLabel.HorizontalAlignment = 'right';
            app.EdgeParameterEditFieldLabel.Position = [417 131 93 22];
            app.EdgeParameterEditFieldLabel.Text = 'Edge Parameter';

            % Create EdgeParameterEditField
            app.EdgeParameterEditField = uieditfield(app.UIFigure, 'numeric');
            app.EdgeParameterEditField.Position = [525 131 100 22];

            % Create IntensityParameterEditField_2Label
            app.IntensityParameterEditField_2Label = uilabel(app.UIFigure);
            app.IntensityParameterEditField_2Label.HorizontalAlignment = 'right';
            app.IntensityParameterEditField_2Label.Position = [400 102 110 22];
            app.IntensityParameterEditField_2Label.Text = 'Intensity Parameter';

            % Create IntensityParameterEditField_2
            app.IntensityParameterEditField_2 = uieditfield(app.UIFigure, 'numeric');
            app.IntensityParameterEditField_2.Position = [525 102 100 22];

            % Create CodeStatusEditFieldLabel
            app.CodeStatusEditFieldLabel = uilabel(app.UIFigure);
            app.CodeStatusEditFieldLabel.HorizontalAlignment = 'right';
            app.CodeStatusEditFieldLabel.Position = [277 164 72 22];
            app.CodeStatusEditFieldLabel.Text = 'Code Status';

            % Create CodeStatusEditField
            app.CodeStatusEditField = uieditfield(app.UIFigure, 'text');
            app.CodeStatusEditField.Position = [364 164 268 22];

            % Create DropletNumberEditFieldLabel
            app.DropletNumberEditFieldLabel = uilabel(app.UIFigure);
            app.DropletNumberEditFieldLabel.HorizontalAlignment = 'right';
            app.DropletNumberEditFieldLabel.Position = [7 81 90 22];
            app.DropletNumberEditFieldLabel.Text = 'Droplet Number';

            % Create DropletNumberEditField
            app.DropletNumberEditField = uieditfield(app.UIFigure, 'numeric');
```

```
            app.DropletNumberEditField.ValueChangedFcn = createCallbackFcn(app,
@DropletNumberEditFieldValueChanged, true);
            app.DropletNumberEditField.Position = [112 81 22 22];

            % Create GiveadropletinthefirstframeswithaKNOWNconditionLabel
            app.GiveadropletinthefirstframeswithaKNOWNconditionLabel =
uilabel(app.UIFigure);

app.GiveadropletinthefirstframeswithaKNOWNconditionLabel.Position = [7 110
346 22];
            app.GiveadropletinthefirstframeswithaKNOWNconditionLabel.Text
= 'Give a droplet in the first 1000 frames with a KNOWN condition';

            % Create AnalyzeFullVideo60dButton
            app.AnalyzeFullVideo60dButton = uibutton(app.UIFigure, 'push');
            app.AnalyzeFullVideo60dButton.ButtonPushedFcn =
createCallbackFcn(app, @AnalyzeFullVideo60dButtonPushed, true);
            app.AnalyzeFullVideo60dButton.FontSize = 18;
            app.AnalyzeFullVideo60dButton.Position = [352 45 132 45];
            app.AnalyzeFullVideo60dButton.Text = {'Analyze Full'; 'Video 60d'; ''; ''};

            % Create FrozenEditFieldLabel
            app.FrozenEditFieldLabel = uilabel(app.UIFigure);
            app.FrozenEditFieldLabel.HorizontalAlignment = 'right';
            app.FrozenEditFieldLabel.Position = [290 18 43 22];
            app.FrozenEditFieldLabel.Text = 'Frozen';

            % Create FrozenEditField
            app.FrozenEditField = uieditfield(app.UIFigure, 'numeric');
            app.FrozenEditField.Position = [348 18 100 22];

            % Create TotalEditFieldLabel
            app.TotalEditFieldLabel = uilabel(app.UIFigure);
            app.TotalEditFieldLabel.HorizontalAlignment = 'right';
            app.TotalEditFieldLabel.Position = [475 18 31 22];
            app.TotalEditFieldLabel.Text = 'Total';

            % Create TotalEditField
            app.TotalEditField = uieditfield(app.UIFigure, 'numeric');
            app.TotalEditField.Position = [521 18 100 22];

            % Create BeginCalculation90dButton
            app.BeginCalculation90dButton = uibutton(app.UIFigure, 'push');
            app.BeginCalculation90dButton.ButtonPushedFcn =
createCallbackFcn(app, @BeginCalculation90dButtonPushed, true);
```

```
            app.BeginCalculation90dButton.FontSize = 16;
            app.BeginCalculation90dButton.Position = [151 31 140 44];
            app.BeginCalculation90dButton.Text = {'Begin Calculation'; ' 90d'; ''};

            % Create AnalyzeFullVideo90dButton
            app.AnalyzeFullVideo90dButton = uibutton(app.UIFigure, 'push');
            app.AnalyzeFullVideo90dButton.ButtonPushedFcn                    =
createCallbackFcn(app, @AnalyzeFullVideo90dButtonPushed, true);
            app.AnalyzeFullVideo90dButton.FontSize = 18;
            app.AnalyzeFullVideo90dButton.Position = [500 45 132 45];
            app.AnalyzeFullVideo90dButton.Text = {'Analyze Full'; 'Video 90d'; ''; ''};

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = parameter_calculator

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

Appendix E

MATLAB code for droplet freezing detection with polarized light

E.1 DNN droplet freezing detection

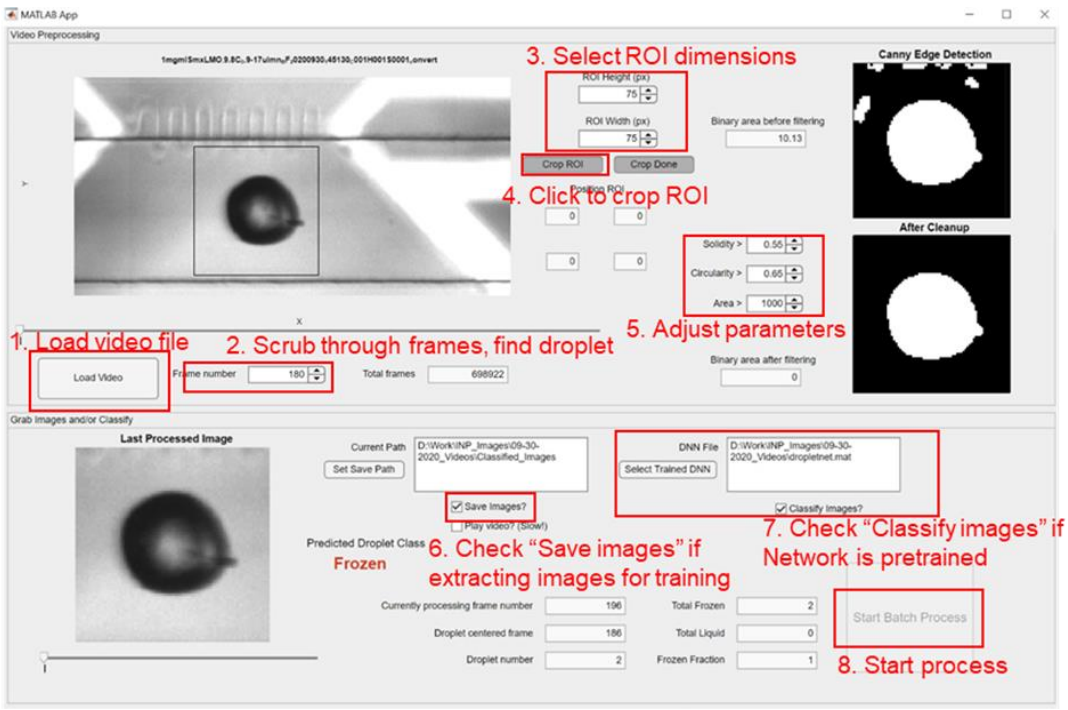


Figure E-1: MATLAB GUI app to crop video, extract droplet images and classify them using a DNN.

The main app INP_Image_Cropper_Grabber_Classifier.mlapp processes videos, extracts images, saves images, and if provided with a pretrained network, also classifies and gives a running frozen fraction count. To use this app, user first loads videos taken with the highspeed camera and converted to AVI format. Then the user scrubs through the video frames either by using the slider or by entering a frame number in the indicated box until a droplet is fully in view. Then the user enters an ROI size such that it is larger than a single droplet. Then the user clicks the crop ROI button and move the ROI window with the mouse and clicks on a droplet. Following this, the binarized droplet image is shown in the frame titled 'Canny Edge Detection'. At this stage there may be some other artifacts left in the frame. The user adjusts the solidity, circularity and area parameters until only the droplet is visible in the 'After Cleanup' frame. The user repeats this for a few droplets to make sure the cleaned up image only shows the droplet and nothing else. When satisfied with the results, the user selects a save path to save extracted images, if desired. Note that this step is essential when extracting images to train a network. If a pretrained network exists, the user selects the trained DNN and checks the corresponding checkbox. Finally, the user clicks the "Start Batch Process" button. The code runs through all the video frames, crops to the ROI, selects images where the droplet is centered in the ROI, saves it, runs it through a trained DNN (depending on options selected). A running count can be seen in the textboxes to the left of the Start Batch Process button.

E.2 Manual image classification

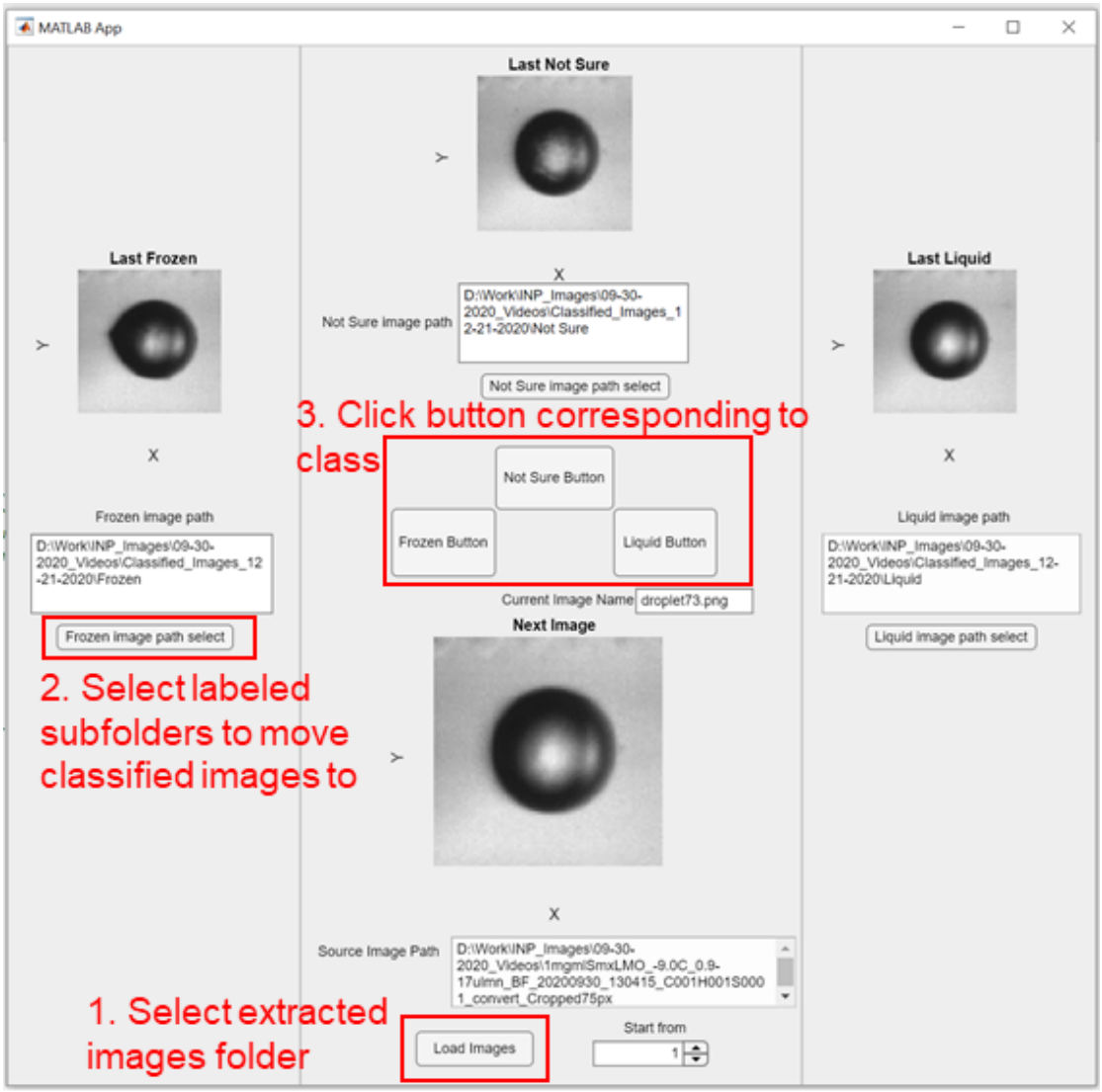


Figure E-2: MATLAB GUI app to manually classify images.

This is a secondary app *INP_Image_Sorter.mlapp* which is useful for classifying droplet images saved through the primary app. The user first selects the folder where droplet images are saved. Then the user selects the paths for “Liquid”, “Frozen” and “Not Sure” (i.e. droplets which cannot be placed in the liquid or frozen class at a first glance)

folders. The first image in the source folder should be loaded in the Next Image frame. If not, the user can select the image number from the "Start from" list. Once the image is visible, the user can click the button in the middle corresponding to a class. The next image should load and the process can be repeated to move through the images in the source folders. Note that the process can be resumed from any numbered droplet image in the source folder by changing the "Start from" value.

E.3 DNN training

The DNN training can be done with the DropletNetCode.m with liquid and frozen droplets placed in subfolders under \Training Data. This must be done before using the INP_Image_Cropper_Grabber_Classifier.mlapp. The natsort.m code is also required to be in working folder for the codes in this appendix to work.

E.4 DNN training code: DropletNetCode.m

```
clear all;

dropletds = imageDatastore('~\Training Data','IncludeSubfolders',...
   true,'LabelSource','foldernames');
labelnames = dropletds.Labels;

[drpTrainImgs,drpTestImgs,drpVldImgs] = splitEachLabel(dropletds,...
                                        0.6,0.2,...
                                        'randomized');

trainds = augmentedImageDatastore([227 227],drpTrainImgs,...
                        'ColorPreprocessing','gray2rgb');
testds = augmentedImageDatastore([227 227],drpTestImgs,...
                        'ColorPreprocessing','gray2rgb');
validds = augmentedImageDatastore([227 227],drpVldImgs,...
                        'ColorPreprocessing','gray2rgb');
```

```
load alexnet.mat;
dropletnet = net;
prebuilt_layers = dropletnet.Layers(1:end-3);
layers = [prebuilt_layers;fullyConnectedLayer(2);softmaxLayer;...
      classificationLayer];

topts = trainingOptions('sgdm', ...
                  'MiniBatchSize',32, ...
                  'MaxEpochs',5, ...
                  'InitialLearnRate',1e-4, ...
                  'Shuffle','every-epoch', ...
                  'ValidationData',validds, ...
                  'ValidationFrequency',90, ...
                  'Verbose',false, ...
                  'ExecutionEnvironment','multi-gpu', ...
                  'Plots','training-progress');


[dropletnet,info] = trainNetwork(trainds,layers,topts);

preds = classify(dropletnet,testds);

truetest = drpTestImgs.Labels;
accuracy = nnz(preds == truetest)/numel(preds)

confusionchart(truetest,preds);
```

E.5 DNN droplet freezing detection code

```
classdef INP_Image_Cropper_Grabber_Classifier < matlab.apps.AppBase

  % Properties that correspond to app components
  properties (Access = public)
    UIFigure                    matlab.ui.Figure
    VideoPreprocessingPanel     matlab.ui.container.Panel
    CropDoneButton              matlab.ui.control.StateButton
    CropROIButton               matlab.ui.control.StateButton
    ROIWidthpxSpinner           matlab.ui.control.Spinner
    ROIWidthpxSpinnerLabel      matlab.ui.control.Label
    ROIHeightpxSpinner          matlab.ui.control.Spinner
    ROIHeightpxSpinnerLabel     matlab.ui.control.Label
    EditField_4                 matlab.ui.control.NumericEditField
    EditField_3                 matlab.ui.control.NumericEditField
    EditField_2                 matlab.ui.control.NumericEditField
```

EditField matlab.ui.control.NumericEditField
TotalFramesEditField matlab.ui.control.NumericEditField
TotalframesLabel matlab.ui.control.Label
BlobAreaEditField_2 matlab.ui.control.NumericEditField
BinaryareaafterfilteringLabel matlab.ui.control.Label
AreaSpinner matlab.ui.control.Spinner
AreaSpinnerLabel matlab.ui.control.Label
CircularitySpinner matlab.ui.control.Spinner
CircularitySpinnerLabel matlab.ui.control.Label
SolidparticleSpinner matlab.ui.control.Spinner
SolidParticleSpinnerLabel matlab.ui.control.Label
BlobAreaEditField matlab.ui.control.NumericEditField
BinaryareabeforefilteringLabel matlab.ui.control.Label
FramenumberSpinner matlab.ui.control.Spinner
FramenumberSpinnerLabel matlab.ui.control.Label
PositionROILabel matlab.ui.control.Label
FrameSlider matlab.ui.control.Slider
LoadVideoButton matlab.ui.control.Button
VideoAxes matlab.ui.control.UIAxes
ProcessedAxes_2 matlab.ui.control.UIAxes
ProcessedAxes matlab.ui.control.UIAxes
GrabImagesandorClassifyPanel matlab.ui.container.Panel
ClassLabel matlab.ui.control.Label
PredictedDropletClassLabel matlab.ui.control.Label
FrozenFractionEditField matlab.ui.control.NumericEditField
FrozenFractionEditFieldLabel matlab.ui.control.Label
TotalLiquidEditField matlab.ui.control.NumericEditField
TotalLiquidEditFieldLabel matlab.ui.control.Label
TotalFrozenEditField matlab.ui.control.NumericEditField
TotalFrozenEditFieldLabel matlab.ui.control.Label
DNNFileTextArea matlab.ui.control.TextArea
DNNFileTextAreaLabel matlab.ui.control.Label
SelectTrainedDNNButton matlab.ui.control.Button
SaveImagesCheckBox matlab.ui.control.CheckBox
ClassifyImagesCheckBox matlab.ui.control.CheckBox
PlayvideoSlowCheckBox matlab.ui.control.CheckBox
SaveProgressSlider matlab.ui.control.Slider
CurrentPathTextArea matlab.ui.control.TextArea
CurrentPathTextAreaLabel matlab.ui.control.Label
SetSavePathButton matlab.ui.control.Button
DropletNumberEditField matlab.ui.control.NumericEditField
DropletnumberLabel matlab.ui.control.Label
SavedFrameEditField matlab.ui.control.NumericEditField
DropletcenteredframeLabel matlab.ui.control.Label
CurrentFrameEditField matlab.ui.control.NumericEditField

```
    CurrentlyprocessingframenumberLabel  matlab.ui.control.Label
    StartBatchProcessButton         matlab.ui.control.Button
    SavedAxes                       matlab.ui.control.UIAxes
end

properties (Access = private)
    v; % Global variable to load the video file into
    framenumber = 1;
    numFrames;
    grayImage;
    grayImageCrop;
    cropRect;
    binaryImage;
    binaryImage2;
    binaryImage3;
    binaryImage4;
    thresh_solidity = 0.55;
    thresh_circularity = 0.65;
    thresh_area = 1000;
    pathSave = pwd;
    dropletnet;
    FrozenCounter = 0;
    LiquidCounter = 0;
    cropflag = 0;
    cropRectangle;
end

methods (Access = private)
    function loadvideofun(app,videofile)
        app.v = VideoReader(videofile);
        [~,videoname,~] = fileparts(videofile);
        app.VideoAxes.Title.String = string(videoname);
        % Count total nmumber of frames in video
        %numFrames = ceil(app.v.FrameRate*app.v.Duration*0.9);
        %app.TotalFramesEditField.Value = numFrames;
        app.numFrames = app.v.NumberOfFrames;
        app.TotalFramesEditField.Value = app.numFrames;

        app.FrameSlider.Limits = [1 app.numFrames];
        app.SaveProgressSlider.Limits = [1 app.numFrames];

        % Load 1st frame of the video as a grayimage
        app.grayImage = read(app.v,1);
        % Show current frame in UIAxes object
        imshow(app.grayImage,'Parent',app.VideoAxes);
```

```
            %Rect_cords = round([1 1 75 75]);
            %app.cropRect = drawrectangle(app.VideoAxes, 'Position', Rect_cords);

            %app.grayImageCrop = imcrop(app.GrayImage,cropRect);

            % Call function to draw processed Binary Image
            %processframe(app, app.grayImageCrop);
            processframe(app, app.grayImage);
        end

        function updateVideoframe(app,currentframe)
            % Load currentframe of the video as a grayimage
            app.grayImage = read(app.v,currentframe);
            % Show current frame in UIAxes object
            imshow(app.grayImage,'Parent',app.VideoAxes);
            if app.cropflag == 1
                app.grayImageCrop = imcrop(app.grayImage,app.cropRectangle);
                rectangle(app.VideoAxes,'position',app.cropRectangle)
                processframe(app, app.grayImageCrop);
            else
                processframe(app, app.grayImage);
            end
        end

%       function updateProcessedframe(app,currentframe)
%
% end
function processframe(app,inputframe)
% Load processed video frame
invImage = 1-im2double(inputframe);
%level = graythresh(invImage);
%app.binaryImage = imbinarize(invImage,level)
app.binaryImage = edge(invImage,'approxcanny');
app.binaryImage = imdilate(app.binaryImage,strel('disk',1));
app.binaryImage = imfill(app.binaryImage, 'holes');
app.binaryImage = imerode(app.binaryImage,strel('disk',1));
app.binaryImage = imclearborder(app.binaryImage);
% Show binary image after edge detection
imshow(app.binaryImage,'Parent',app.ProcessedAxes);
% Print Area after edge detection
app.BlobAreaEditField.Value = bwarea(app.binaryImage);
props = regionprops(app.binaryImage, 'Solidity', 'Area', 'perimeter');
allAreas = [props.Area];
allSolidities = [props.Solidity];
```

```
        allCircularities = 4 * pi * allAreas ./ [props.Perimeter] .^ 2;
        % Get rid of blobs less solid than 0.85%
        app.binaryImage2 = bwpropfilt(app.binaryImage, 'Solidity', [app.thresh_solidity,
inf]);
        % Get rid of blobs with low circularity.
        highCircularityIndexes = find(allCircularities > app.thresh_circularity);
        labeledImage = bwlabel(app.binaryImage2);
        app.binaryImage3 = ismember(labeledImage, highCircularityIndexes);
        % Get rid of blobs with low area.
        app.binaryImage4 = bwpropfilt(app.binaryImage2, 'Area', [app.thresh_area, inf]);
        % Show binary image after cleanup
        imshow(app.binaryImage4,'Parent',app.ProcessedAxes_2);
        % Print Area after cleanup
        app.BlobAreaEditField_2.Value = bwarea(app.binaryImage4);

        end
        function calculateImage(app,currentframe)
        app.grayImage = read(app.v,currentframe);
        if app.cropflag == 1
        app.grayImage = imcrop(app.grayImage,app.cropRectangle);
        end
        invImage = 1-im2double(app.grayImage);
        %level = graythresh(invImage);
        %app.binaryImage = imbinarize(invImage,level)
        app.binaryImage = edge(invImage,'approxcanny');
        app.binaryImage = imdilate(app.binaryImage,strel('disk',1));
        app.binaryImage = imfill(app.binaryImage, 'holes');
        app.binaryImage = imerode(app.binaryImage,strel('disk',1));
        app.binaryImage = imclearborder(app.binaryImage);
        props = regionprops(app.binaryImage, 'Area', 'perimeter');
        allAreas = [props.Area];
        allCircularities = 4 * pi * allAreas ./ [props.Perimeter] .^ 2;
        app.binaryImage2 = bwpropfilt(app.binaryImage, 'Solidity', [app.thresh_solidity,
inf]);
        highCircularityIndexes = find(allCircularities > app.thresh_circularity);
        labeledImage = bwlabel(app.binaryImage2);
        app.binaryImage3 = ismember(labeledImage, highCircularityIndexes);
        app.binaryImage4 = bwpropfilt(app.binaryImage3, 'Area', [app.thresh_area, inf]);
        end
        end

        % Callbacks that handle component events
        methods (Access = private)

        % Code that executes after component creation
```

```
function startupFcn(app)
% app.CurrentFrameEditField.ValueDisplayFormat = '%d';
% app.SavedFrameEditField.ValueDisplayFormat = '%d';
end

% Button pushed function: LoadVideoButton
function LoadVideoButtonPushed(app, event)
% Select only avi and mp4 files
filterspec = {'*.avi;*.mp4;'};
% Show dialogbox for loading file and load into variable
[vload, pload] = uigetfile(filterspec);
figure(app.UIFigure)
% Make sure user didn't cancel uigetfile dialog
if (ischar(pload))
fname = [pload vload];
loadvideofun(app, fname);
end
end

% Value changed function: FramenumberSpinner
function FramenumberSpinnerValueChanged(app, event)
app.framenumber = app.FramenumberSpinner.Value;
updateVideoframe(app,app.framenumber);
app.FrameSlider.Value = app.framenumber;
% app.Curr
end

% Value changed function: SolidparticleSpinner
function SolidityParticleSpinnerValueChanged(app, event)
app.thresh_solidity = app.SolidityParticleSpinner.Value;
end

% Value changed function: CircularityParticleSpinner
function CircularityParticleSpinnerValueChanged(app, event)
app.thresh_circularity = app.CircularityParticleSpinner.Value;
end

% Value changed function: AreaSpinner
function AreaSpinnerValueChanged(app, event)
app.thresh_area = app.AreaSpinner.Value;
end

% Value changed function: FrameSlider
function FrameSliderValueChanged(app, event)
app.framenumber = round(app.FrameSlider.Value);
```

```
updateVideoframe(app,app.framenumber);
app.FramenumberSpinner.Value = app.framenumber;
end

% Button pushed function: StartBatchProcessButton
function StartBatchProcessButtonPushed(app, event)
app.StartBatchProcessButton.Enable = 0;
app.ProcessedAxes.Visible = 'off';
app.ProcessedAxes_2.Visible = 'off';
% all_frames = zeros(app.numFrames,2);
nullframe_counter = 0;
droplet_number = 1;
% droplet_frames = [];
size_droplet_frames = 0;
folder_name = app.v.name;
folder_name = erase(folder_name,".avi");
directory = [app.pathSave,'\',folder_name];
mkdir(directory);

for i = app.framenumber : app.numFrames
calculateImage(app,i);

% Play video while saving frames
if app.PlayvideoSlowCheckBox.Value == 1
app.grayImage = read(app.v,i);
imshow(app.grayImage,'Parent',app.VideoAxes);
rectangle(app.VideoAxes,'position',app.cropRectangle);
end
app.CurrentFrameEditField.Value = i;
app.SaveProgressSlider.Value = i;
%all_frames(i,1) = i;
%all_frames(i,2) = bwarea(app.binaryImage4);
if bwarea(app.binaryImage4) > 0
%droplet_frames = [droplet_frames;all_frames(i,:)];
%size_droplet_frames = size(droplet_frames,1);
size_droplet_frames = size_droplet_frames + 1;
nullframe_counter = 0;
end
if nullframe_counter > 2 && size_droplet_frames > 2 %Set number of frames between no droplets and droplet size as criteria for saving a droplet image
frame_number_to_write = floor((i-3)-size_droplet_frames/2); %Find the median frame number of the droplet passing through
file_name = ['droplet',num2str(droplet_number),'.png'];
fulldestination = fullfile(directory, file_name);
dropletImage = read(app.v,frame_number_to_write);
```

```
        if app.cropflag == 1
        dropletImage = imcrop(dropletImage,app.cropRectangle);
        end
        if app.SaveImagesCheckBox.Value == 1 %Saving images or not based on
checkbox
        imwrite(dropletImage, fulldestination);
        end
        if app.ClassifyImagesCheckBox.Value == 1 %Classifying images or not based on
checkbox
        inputSize = app.dropletnet.Layers(1).InputSize;
        I = imresize(dropletImage,inputSize(1:2));
        I = I(:,:,[1 1 1]); %Convert gray to rgb
        label = classify(app.dropletnet,I);
        %app.DropletClassEditField.Value = string(label);
        app.ClassLabel.Text = string(label);
        if strcmp(string(label), 'Frozen')
        app.FrozenCounter = app.FrozenCounter + 1;
        app.TotalFrozenEditField.Value = app.FrozenCounter;
        else
        app.LiquidCounter = app.LiquidCounter + 1;
        app.TotalLiquidEditField.Value = app.LiquidCounter;
        end
        app.FrozenFractionEditField.Value                                    =
app.FrozenCounter/(app.FrozenCounter+app.LiquidCounter);
        end
        imshow(dropletImage,'Parent',app.SavedAxes);
        app.DropletNumberEditField.Value = droplet_number;
        app.SavedFrameEditField.Value = frame_number_to_write;
        droplet_number = droplet_number + 1;
        size_droplet_frames = 0;
        end
        nullframe_counter = nullframe_counter + 1;
        end
        end

        % Button pushed function: SetSavePathButton
        function SetSavePathButtonPushed(app, event)
        % Display uigetfile dialog
        app.pathSave = uigetdir;
        figure(app.UIFigure)
        % Make sure user didn't cancel uigetdir dialog
        if (ischar(app.pathSave))
        app.CurrentPathTextArea.Value = app.pathSave;
        end
        end
```

```
        % Button pushed function: SelectTrainedDNNButton
        function SelectTrainedDNNButtonPushed(app, event)
        [file,path] = uigetfile('*.mat', 'Select a Pretrained DNN .mat file');
        if isequal(file,0)
        disp('User selected Cancel')
        else
        load(fullfile(path,file))
        app.dropletnet = dropletnet;
        app.DNNFileTextArea.Value = fullfile(path,file);
        end
        end

        % Value changed function: CropROIButton
        function CropROIButtonValueChanged(app, event)
        value = app.CropROIButton.Value;
        app.cropflag = 1;
        Rect_cords = round([1 1 app.ROIHeightpxSpinner.Value
app.ROIWidthpxSpinner.Value]);
        app.cropRect = drawrectangle(app.VideoAxes, 'Position', Rect_cords);
        uiwait;
        close(gcf);
        end

        % Value changed function: CropDoneButton
        function CropDoneButtonValueChanged(app, event)
        value = app.CropDoneButton.Value;
        app.cropRectangle = app.cropRect.Position;
        uiresume;
        app.grayImageCrop = imcrop(app.grayImage,app.cropRectangle);
        rectangle(app.VideoAxes,'position',app.cropRectangle);
        processframe(app,app.grayImageCrop);
        end
        end

        % Component initialization
        methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

        % Create UIFigure and hide until all components are created
        app.UIFigure = uifigure('Visible', 'off');
        app.UIFigure.Position = [100 100 1321 845];
        app.UIFigure.Name = 'MATLAB App';
```

```
        % Create GrabImagesandorClassifyPanel
        app.GrabImagesandorClassifyPanel = uipanel(app.UIFigure);
        app.GrabImagesandorClassifyPanel.Title = 'Grab Images and/or Classify';
        app.GrabImagesandorClassifyPanel.Position = [11 9 1302 361];

        % Create SavedAxes
        app.SavedAxes = uiaxes(app.GrabImagesandorClassifyPanel);
        title(app.SavedAxes, 'Last Processed Image')
        app.SavedAxes.PlotBoxAspectRatio = [1 1 1];
        app.SavedAxes.XTick = [];
        app.SavedAxes.YTick = [];
        app.SavedAxes.Position = [46 60 327 279];

        % Create StartBatchProcessButton
        app.StartBatchProcessButton = uibutton(app.GrabImagesandorClassifyPanel,
'push');
        app.StartBatchProcessButton.ButtonPushedFcn = createCallbackFcn(app,
@StartBatchProcessButtonPushed, true);
        app.StartBatchProcessButton.FontSize = 16;
        app.StartBatchProcessButton.Position = [1042 39 158 136];
        app.StartBatchProcessButton.Text = 'Start Batch Process';

        % Create CurrentlyprocessingframenumberLabel
        app.CurrentlyprocessingframenumberLabel =
uilabel(app.GrabImagesandorClassifyPanel);
        app.CurrentlyprocessingframenumberLabel.HorizontalAlignment = 'right';
        app.CurrentlyprocessingframenumberLabel.Position = [458 110 195 22];
        app.CurrentlyprocessingframenumberLabel.Text = 'Currently processing frame
number';

        % Create CurrentFrameEditField
        app.CurrentFrameEditField = uieditfield(app.GrabImagesandorClassifyPanel,
'numeric');
        app.CurrentFrameEditField.ValueDisplayFormat = '%d';
        app.CurrentFrameEditField.Editable = 'off';
        app.CurrentFrameEditField.Position = [668 110 100 22];

        % Create DropletcenteredframeLabel
        app.DropletcenteredframeLabel = uilabel(app.GrabImagesandorClassifyPanel);
        app.DropletcenteredframeLabel.HorizontalAlignment = 'right';
        app.DropletcenteredframeLabel.Position = [525 76 128 22];
        app.DropletcenteredframeLabel.Text = 'Droplet centered frame';

        % Create SavedFrameEditField
```

```
        app.SavedFrameEditField = uieditfield(app.GrabImagesandorClassifyPanel,
'numeric');
        app.SavedFrameEditField.ValueDisplayFormat = '%d';
        app.SavedFrameEditField.Editable = 'off';
        app.SavedFrameEditField.Position = [668 76 100 22];

        % Create DropletnumberLabel
        app.DropletnumberLabel = uilabel(app.GrabImagesandorClassifyPanel);
        app.DropletnumberLabel.HorizontalAlignment = 'right';
        app.DropletnumberLabel.Position = [565 43 88 22];
        app.DropletnumberLabel.Text = 'Droplet number';

        % Create DropletNumberEditField
        app.DropletNumberEditField = uieditfield(app.GrabImagesandorClassifyPanel,
'numeric');
        app.DropletNumberEditField.Editable = 'off';
        app.DropletNumberEditField.Position = [668 43 100 22];

        % Create SetSavePathButton
        app.SetSavePathButton = uibutton(app.GrabImagesandorClassifyPanel, 'push');
        app.SetSavePathButton.ButtonPushedFcn = createCallbackFcn(app,
@SetSavePathButtonPushed, true);
        app.SetSavePathButton.Position = [393 279 100 22];
        app.SetSavePathButton.Text = 'Set Save Path';

        % Create CurrentPathTextAreaLabel
        app.CurrentPathTextAreaLabel = uilabel(app.GrabImagesandorClassifyPanel);
        app.CurrentPathTextAreaLabel.HorizontalAlignment = 'right';
        app.CurrentPathTextAreaLabel.Position = [421 306 74 22];
        app.CurrentPathTextAreaLabel.Text = 'Current Path';

        % Create CurrentPathTextArea
        app.CurrentPathTextArea = uitextarea(app.GrabImagesandorClassifyPanel);
        app.CurrentPathTextArea.Position = [505 262 216 68];

        % Create SaveProgressSlider
        app.SaveProgressSlider = uislider(app.GrabImagesandorClassifyPanel);
        app.SaveProgressSlider.Limits = [0 1000];
        app.SaveProgressSlider.MajorTickLabels = {''};
        app.SaveProgressSlider.MinorTicks = [];
        app.SaveProgressSlider.Position = [46 55 339 3];

        % Create PlayvideoSlowCheckBox
        app.PlayvideoSlowCheckBox = uicheckbox(app.GrabImagesandorClassifyPanel);
        app.PlayvideoSlowCheckBox.Text = 'Play video? (Slow!)';
```

```
            app.PlayvideoSlowCheckBox.Position = [551 209 125 22];

            % Create ClassifyImagesCheckBox
            app.ClassifyImagesCheckBox = uicheckbox(app.GrabImagesandorClassifyPanel);
            app.ClassifyImagesCheckBox.Text = 'Classify Images?';
            app.ClassifyImagesCheckBox.Position = [954 230 114 22];
            app.ClassifyImagesCheckBox.Value = true;

            % Create SaveImagesCheckBox
            app.SaveImagesCheckBox = uicheckbox(app.GrabImagesandorClassifyPanel);
            app.SaveImagesCheckBox.Text = 'Save Images?';
            app.SaveImagesCheckBox.Position = [551 233 99 22];
            app.SaveImagesCheckBox.Value = true;

            % Create SelectTrainedDNNButton
            app.SelectTrainedDNNButton = uibutton(app.GrabImagesandorClassifyPanel,
'push');
            app.SelectTrainedDNNButton.ButtonPushedFcn = createCallbackFcn(app,
@SelectTrainedDNNButtonPushed, true);
            app.SelectTrainedDNNButton.Position = [759 279 122 22];
            app.SelectTrainedDNNButton.Text = 'Select Trained DNN';

            % Create DNNFileTextAreaLabel
            app.DNNFileTextAreaLabel = uilabel(app.GrabImagesandorClassifyPanel);
            app.DNNFileTextAreaLabel.HorizontalAlignment = 'right';
            app.DNNFileTextAreaLabel.Position = [829 306 54 22];
            app.DNNFileTextAreaLabel.Text = 'DNN File';

            % Create DNNFileTextArea
            app.DNNFileTextArea = uitextarea(app.GrabImagesandorClassifyPanel);
            app.DNNFileTextArea.Position = [893 262 216 68];

            % Create TotalFrozenEditFieldLabel
            app.TotalFrozenEditFieldLabel = uilabel(app.GrabImagesandorClassifyPanel);
            app.TotalFrozenEditFieldLabel.HorizontalAlignment = 'right';
            app.TotalFrozenEditFieldLabel.Position = [819 110 72 22];
            app.TotalFrozenEditFieldLabel.Text = 'Total Frozen';

            % Create TotalFrozenEditField
            app.TotalFrozenEditField = uieditfield(app.GrabImagesandorClassifyPanel,
'numeric');
            app.TotalFrozenEditField.ValueDisplayFormat = '%d';
            app.TotalFrozenEditField.Editable = 'off';
            app.TotalFrozenEditField.Position = [906 110 100 22];
```

```
        % Create TotalLiquidEditFieldLabel
        app.TotalLiquidEditFieldLabel = uilabel(app.GrabImagesandorClassifyPanel);
        app.TotalLiquidEditFieldLabel.HorizontalAlignment = 'right';
        app.TotalLiquidEditFieldLabel.Position = [825 76 66 22];
        app.TotalLiquidEditFieldLabel.Text = 'Total Liquid';

        % Create TotalLiquidEditField
        app.TotalLiquidEditField = uieditfield(app.GrabImagesandorClassifyPanel,
'numeric');
        app.TotalLiquidEditField.ValueDisplayFormat = '%d';
        app.TotalLiquidEditField.Editable = 'off';
        app.TotalLiquidEditField.Position = [906 76 100 22];

        % Create FrozenFractionEditFieldLabel
        app.FrozenFractionEditFieldLabel = uilabel(app.GrabImagesandorClassifyPanel);
        app.FrozenFractionEditFieldLabel.HorizontalAlignment = 'right';
        app.FrozenFractionEditFieldLabel.Position = [801 43 90 22];
        app.FrozenFractionEditFieldLabel.Text = 'Frozen Fraction';

        % Create FrozenFractionEditField
        app.FrozenFractionEditField = uieditfield(app.GrabImagesandorClassifyPanel,
'numeric');
        app.FrozenFractionEditField.Editable = 'off';
        app.FrozenFractionEditField.Position = [906 43 100 22];

        % Create PredictedDropletClassLabel
        app.PredictedDropletClassLabel = uilabel(app.GrabImagesandorClassifyPanel);
        app.PredictedDropletClassLabel.FontSize = 14;
        app.PredictedDropletClassLabel.Position = [372 188 152 22];
        app.PredictedDropletClassLabel.Text = 'Predicted Droplet Class';

        % Create ClassLabel
        app.ClassLabel = uilabel(app.GrabImagesandorClassifyPanel);
        app.ClassLabel.HorizontalAlignment = 'center';
        app.ClassLabel.FontSize = 20;
        app.ClassLabel.FontWeight = 'bold';
        app.ClassLabel.FontColor = [1 0 0];
        app.ClassLabel.Position = [384 162 109 24];
        app.ClassLabel.Text = 'Class';

        % Create VideoPreprocessingPanel
        app.VideoPreprocessingPanel = uipanel(app.UIFigure);
        app.VideoPreprocessingPanel.Title = 'Video Preprocessing';
        app.VideoPreprocessingPanel.Position = [11 378 1302 468];
```

```
        % Create ProcessedAxes
        app.ProcessedAxes = uiaxes(app.VideoPreprocessingPanel);
        title(app.ProcessedAxes, 'Canny Edge Detection')
        app.ProcessedAxes.PlotBoxAspectRatio = [1 1 1];
        app.ProcessedAxes.XTick = [];
        app.ProcessedAxes.XTickLabel = {};
        app.ProcessedAxes.YTick = [];
        app.ProcessedAxes.YTickLabel = {};
        app.ProcessedAxes.Position = [1002 215 299 231];

        % Create ProcessedAxes_2
        app.ProcessedAxes_2 = uiaxes(app.VideoPreprocessingPanel);
        title(app.ProcessedAxes_2, 'After Cleanup')
        app.ProcessedAxes_2.DataAspectRatio = [0.821078431372549 1 1];
        app.ProcessedAxes_2.PlotBoxAspectRatio = [1 1 1];
        app.ProcessedAxes_2.XTick = [];
        app.ProcessedAxes_2.XTickLabel = '';
        app.ProcessedAxes_2.YTick = [];
        app.ProcessedAxes_2.Position = [1014 1 274 231];

        % Create VideoAxes
        app.VideoAxes = uiaxes(app.VideoPreprocessingPanel);
        title(app.VideoAxes, 'Raw Video Frame')
        xlabel(app.VideoAxes, 'X')
        ylabel(app.VideoAxes, 'Y')
        app.VideoAxes.LabelFontSizeMultiplier = 1;
        app.VideoAxes.PlotBoxAspectRatio = [1.9 1 1];
        app.VideoAxes.FontSize = 10;
        app.VideoAxes.TitleFontSizeMultiplier = 1;
        app.VideoAxes.Position = [16 100 660 335];

        % Create LoadVideoButton
        app.LoadVideoButton = uibutton(app.VideoPreprocessingPanel, 'push');
        app.LoadVideoButton.ButtonPushedFcn = createCallbackFcn(app,
@LoadVideoButtonPushed, true);
        app.LoadVideoButton.Position = [38 8 150 51];
        app.LoadVideoButton.Text = 'Load Video';

        % Create FrameSlider
        app.FrameSlider = uislider(app.VideoPreprocessingPanel);
        app.FrameSlider.Limits = [0 1000];
        app.FrameSlider.MajorTickLabels = {''};
        app.FrameSlider.ValueChangedFcn = createCallbackFcn(app,
@FrameSliderValueChanged, true);
        app.FrameSlider.MinorTicks = [];
```

```
            app.FrameSlider.Position = [16 91 720 3];

            % Create PositionROILabel
            app.PositionROILabel = uilabel(app.VideoPreprocessingPanel);
            app.PositionROILabel.Position = [699 256 73 22];
            app.PositionROILabel.Text = 'Position ROI';

            % Create FramenumberSpinnerLabel
            app.FramenumberSpinnerLabel = uilabel(app.VideoPreprocessingPanel);
            app.FramenumberSpinnerLabel.HorizontalAlignment = 'right';
            app.FramenumberSpinnerLabel.Position = [197 22 84 22];
            app.FramenumberSpinnerLabel.Text = 'Frame number';

            % Create FramenumberSpinner
            app.FramenumberSpinner = uispinner(app.VideoPreprocessingPanel);
            app.FramenumberSpinner.ValueDisplayFormat = '%.0f';
            app.FramenumberSpinner.ValueChangedFcn = createCallbackFcn(app,
@FramenumberSpinnerValueChanged, true);
            app.FramenumberSpinner.Position = [296 22 98 22];
            app.FramenumberSpinner.Value = 1;

            % Create BinaryareabeforefilteringLabel
            app.BinaryareabeforefilteringLabel = uilabel(app.VideoPreprocessingPanel);
            app.BinaryareabeforefilteringLabel.HorizontalAlignment = 'right';
            app.BinaryareabeforefilteringLabel.Position = [869 340 146 22];
            app.BinaryareabeforefilteringLabel.Text = 'Binary area before filtering';

            % Create BlobAreaEditField
            app.BlobAreaEditField = uieditfield(app.VideoPreprocessingPanel, 'numeric');
            app.BlobAreaEditField.Editable = 'off';
            app.BlobAreaEditField.Position = [892 319 100 22];

            % Create SoliditySpinnerLabel
            app.SoliditySpinnerLabel = uilabel(app.VideoPreprocessingPanel);
            app.SoliditySpinnerLabel.HorizontalAlignment = 'right';
            app.SoliditySpinnerLabel.Position = [859 188 54 22];
            app.SoliditySpinnerLabel.Text = 'Solidity >';

            % Create SoliditySpinner
            app.SoliditySpinner = uispinner(app.VideoPreprocessingPanel);
            app.SoliditySpinner.Step = 0.01;
            app.SoliditySpinner.ValueChangedFcn = createCallbackFcn(app,
@SoliditySpinnerValueChanged, true);
            app.SoliditySpinner.Position = [918 188 72 22];
            app.SoliditySpinner.Value = 0.55;
```

```
        % Create CircularitySpinnerLabel
        app.CircularitySpinnerLabel = uilabel(app.VideoPreprocessingPanel);
        app.CircularitySpinnerLabel.HorizontalAlignment = 'right';
        app.CircularitySpinnerLabel.Position = [844 152 69 22];
        app.CircularitySpinnerLabel.Text = 'Circularity >';

        % Create CircularitySpinner
        app.CircularitySpinner = uispinner(app.VideoPreprocessingPanel);
        app.CircularitySpinner.Step = 0.01;
        app.CircularitySpinner.ValueChangedFcn = createCallbackFcn(app,
@CircularitySpinnerValueChanged, true);
        app.CircularitySpinner.Position = [918 152 72 22];
        app.CircularitySpinner.Value = 0.65;

        % Create AreaSpinnerLabel
        app.AreaSpinnerLabel = uilabel(app.VideoPreprocessingPanel);
        app.AreaSpinnerLabel.HorizontalAlignment = 'right';
        app.AreaSpinnerLabel.Position = [872 115 41 22];
        app.AreaSpinnerLabel.Text = 'Area >';

        % Create AreaSpinner
        app.AreaSpinner = uispinner(app.VideoPreprocessingPanel);
        app.AreaSpinner.Step = 100;
        app.AreaSpinner.ValueChangedFcn = createCallbackFcn(app,
@AreaSpinnerValueChanged, true);
        app.AreaSpinner.Position = [918 115 72 22];
        app.AreaSpinner.Value = 1000;

        % Create BinaryareaafterfilteringLabel
        app.BinaryareaafterfilteringLabel = uilabel(app.VideoPreprocessingPanel);
        app.BinaryareaafterfilteringLabel.HorizontalAlignment = 'right';
        app.BinaryareaafterfilteringLabel.Position = [868 45 136 22];
        app.BinaryareaafterfilteringLabel.Text = 'Binary area after filtering';

        % Create BlobAreaEditField_2
        app.BlobAreaEditField_2 = uieditfield(app.VideoPreprocessingPanel, 'numeric');
        app.BlobAreaEditField_2.Editable = 'off';
        app.BlobAreaEditField_2.Position = [886 24 100 22];

        % Create TotalframesLabel
        app.TotalframesLabel = uilabel(app.VideoPreprocessingPanel);
        app.TotalframesLabel.HorizontalAlignment = 'right';
        app.TotalframesLabel.Position = [433 22 71 22];
        app.TotalframesLabel.Text = 'Total frames';
```

```
% Create TotalFramesEditField
app.TotalFramesEditField = uieditfield(app.VideoPreprocessingPanel, 'numeric');
app.TotalFramesEditField.ValueDisplayFormat = '%d';
app.TotalFramesEditField.Editable = 'off';
app.TotalFramesEditField.Position = [519 22 100 22];

% Create EditField
app.EditField = uieditfield(app.VideoPreprocessingPanel, 'numeric');
app.EditField.Editable = 'off';
app.EditField.Position = [668 223 42 22];

% Create EditField_2
app.EditField_2 = uieditfield(app.VideoPreprocessingPanel, 'numeric');
app.EditField_2.Editable = 'off';
app.EditField_2.Position = [753 223 41 22];

% Create EditField_3
app.EditField_3 = uieditfield(app.VideoPreprocessingPanel, 'numeric');
app.EditField_3.Editable = 'off';
app.EditField_3.Position = [669 167 41 22];

% Create EditField_4
app.EditField_4 = uieditfield(app.VideoPreprocessingPanel, 'numeric');
app.EditField_4.Editable = 'off';
app.EditField_4.Position = [753 167 41 22];

% Create ROIHeightpxSpinnerLabel
app.ROIHeightpxSpinnerLabel = uilabel(app.VideoPreprocessingPanel);
app.ROIHeightpxSpinnerLabel.HorizontalAlignment = 'right';
app.ROIHeightpxSpinnerLabel.Position = [709 395 89 22];
app.ROIHeightpxSpinnerLabel.Text = 'ROI Height (px)';

% Create ROIHeightpxSpinner
app.ROIHeightpxSpinner = uispinner(app.VideoPreprocessingPanel);
app.ROIHeightpxSpinner.Position = [709 374 100 22];
app.ROIHeightpxSpinner.Value = 75;

% Create ROIWidthpxSpinnerLabel
app.ROIWidthpxSpinnerLabel = uilabel(app.VideoPreprocessingPanel);
app.ROIWidthpxSpinnerLabel.HorizontalAlignment = 'right';
app.ROIWidthpxSpinnerLabel.Position = [713 340 85 22];
app.ROIWidthpxSpinnerLabel.Text = 'ROI Width (px)';

% Create ROIWidthpxSpinner
```

```
        app.ROIWidthpxSpinner = uispinner(app.VideoPreprocessingPanel);
        app.ROIWidthpxSpinner.Position = [709 319 100 22];
        app.ROIWidthpxSpinner.Value = 75;

        % Create CropROIButton
        app.CropROIButton = uibutton(app.VideoPreprocessingPanel, 'state');
        app.CropROIButton.ValueChangedFcn = createCallbackFcn(app,
@CropROIButtonValueChanged, true);
        app.CropROIButton.Text = 'Crop ROI';
        app.CropROIButton.Position = [640 287 100 22];

        % Create CropDoneButton
        app.CropDoneButton = uibutton(app.VideoPreprocessingPanel, 'state');
        app.CropDoneButton.ValueChangedFcn = createCallbackFcn(app,
@CropDoneButtonValueChanged, true);
        app.CropDoneButton.Text = 'Crop Done';
        app.CropDoneButton.Position = [753 287 100 22];

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
        end
        end

        % App creation and deletion
        methods (Access = public)

        % Construct app
        function app = INP_Image_Cropper_Grabber_Classifier

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargout == 0
        clear app
        end
        end

        % Code that executes before app deletion
        function delete(app)
```

```
% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
```

Manual image classification code

```
classdef INP_Image_Sorter < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                        matlab.ui.Figure
        GridLayout                      matlab.ui.container.GridLayout
        LeftPanel                       matlab.ui.container.Panel
        FrozenimagepathselectButton     matlab.ui.control.Button
        FrozenimagepathTextArea         matlab.ui.control.TextArea
        FrozenimagepathTextAreaLabel    matlab.ui.control.Label
        LastFrozenAxes                  matlab.ui.control.UIAxes
        CenterPanel                     matlab.ui.container.Panel
        NotSureimagepathselectButton    matlab.ui.control.Button
        NotSureimagepathTextArea        matlab.ui.control.TextArea
        NotSureimagepathTextAreaLabel   matlab.ui.control.Label
        SourceImagePathTextArea         matlab.ui.control.TextArea
        SourceImagePathTextAreaLabel    matlab.ui.control.Label
        LiquidButton                    matlab.ui.control.Button
        FrozenButton                    matlab.ui.control.Button
        CurrentImageNameEditField       matlab.ui.control.EditField
        CurrentImageNameEditFieldLabel  matlab.ui.control.Label
        StartfromSpinner                matlab.ui.control.Spinner
        StartfromSpinnerLabel           matlab.ui.control.Label
        NotSureButton                   matlab.ui.control.Button
        LoadImagesButton                matlab.ui.control.Button
        LastNotSureAxes                 matlab.ui.control.UIAxes
        NextImageAxes                   matlab.ui.control.UIAxes
        RightPanel                      matlab.ui.container.Panel
        LiquidimagepathselectButton     matlab.ui.control.Button
        LiquidimagepathTextArea         matlab.ui.control.TextArea
        LiquidimagepathTextAreaLabel    matlab.ui.control.Label
        LastLiquidAxes                  matlab.ui.control.UIAxes
    end

    % Properties that correspond to apps with auto-reflow
    properties (Access = private)
        onePanelWidth = 576;
```

```
            twoPanelWidth = 768;
        end


        properties (Access = private)
            sourceFileList; % File list in source file directory
            sourcepath; % Directory path for the source files (Unsorted)
            numTotal; % Total number of image files in source directory
            startimagenumber; % first file to start the image sorting from
            CurrentImageToSort; % Full image name for the current image shown
            pathLiquid;
            pathFrozen;
            pathNotSure;
            i = 0; % Current image number in source file list
        end

        methods (Access = private)

            function loadimagespathfun(app,imagespath)
                app.sourceFileList = dir([imagespath '\*.png']); % Creating global variable with all source filenames
                app.numTotal = size(app.sourceFileList,1); % Counting total number of source files
                [~,Index] = natsort({app.sourceFileList.name}); %This step won't work without natsort.m being in same folder as the app, for now
                app.sourceFileList = app.sourceFileList(Index); % Sorting the filelist using the natural sorting function
                app.SourceImagePathTextArea.Value = imagespath;
            end

            % Functions to update images in the variuous figure boxes
            function updateNextImage(app, index)

                app.i = index;
                app.i = app.i + 1; % update index to point to the next unsorted file in source file list
                if app.i <= app.numTotal
                    imagefile = app.sourceFileList(app.i);
                    imagefilefull = fullfile(app.sourcepath, imagefile.name);
                    imshow(imagefilefull,'Parent',app.NextImageAxes);
                    app.CurrentImageToSort = imagefilefull;
                    app.CurrentImageNameEditField.Value = [imagefile.name];
                end
            end
```

```
        function updateLastLiquid(app, imagefile)
            imshow(imagefile,'Parent',app.LastLiquidAxes);
        end

        function updateLastFrozen(app, imagefile)
            imshow(imagefile,'Parent',app.LastFrozenAxes);
        end

        function updateLastNotSure(app, imagefile)
            imshow(imagefile,'Parent',app.LastNotSureAxes);
        end
    end


    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: LoadImagesButton
        function LoadImagesButtonPushed(app, event)

            % Display uigetfile dialog
            app.sourcepath = uigetdir;
            figure(app.UIFigure) %keep focus on window

            % Make sure user didn't cancel uigetdir dialog
            if (ischar(app.sourcepath))
              loadimagespathfun(app, app.sourcepath);
            end
        end

        % Value changed function: StartfromSpinner
        function StartfromSpinnerValueChanged(app, event)
            app.startimagenumber = app.StartfromSpinner.Value;
            sendindex = app.startimagenumber - 1; %subtracting 1 from index because
1 will be added in updateNextImage
            updateNextImage(app, app.startimagenumber);
        end

        % Button pushed function: LiquidButton
        function LiquidButtonPushed(app, event)
            updateLastLiquid(app,  app.CurrentImageToSort);  % Update last liquid
sorted image

            existingLiquid = dir([app.pathLiquid '\*.png']); % Count number of existing
files in the folder
```

```
                numLiquid = size(existingLiquid,1);
                numLiquid = numLiquid + 1;
                outputBaseFileName = sprintf('Liquid%i.png',numLiquid); % Create output filename by adding one to total number of existing files
                outputFullFileName = fullfile(app.pathLiquid, outputBaseFileName);
                copyfile(app.CurrentImageToSort, outputFullFileName); % Copy file from source folder to target folder and rename

                updateNextImage(app, app.i);

            end

            % Button pushed function: LiquidimagepathselectButton
            function LiquidimagepathselectButtonPushed(app, event)
                % Display uigetfile dialog
                app.pathLiquid = uigetdir;
        figure(app.UIFigure)
        % Make sure user didn't cancel uigetdir dialog
        if (ischar(app.pathLiquid))
        app.LiquidimagepathTextArea.Value = app.pathLiquid;
        end
        end

        % Button pushed function: FrozenimagepathselectButton
        function FrozenimagepathselectButtonPushed(app, event)
        % Display uigetfile dialog
        app.pathFrozen = uigetdir;
        figure(app.UIFigure)
        % Make sure user didn't cancel uigetdir dialog
        if (ischar(app.pathFrozen))
        app.FrozenimagepathTextArea.Value = app.pathFrozen;
        end
        end

        % Button pushed function: NotSureimagepathselectButton
        function NotSureimagepathselectButtonPushed(app, event)
        % Display uigetfile dialog
        app.pathNotSure = uigetdir;
        figure(app.UIFigure)
        % Make sure user didn't cancel uigetdir dialog
        if (ischar(app.pathNotSure))
        app.NotSureimagepathTextArea.Value = app.pathNotSure;
        end
        end
```

```
        % Button pushed function: FrozenButton
        function FrozenButtonPushed(app, event)
        updateLastFrozen(app, app.CurrentImageToSort); % Update last Frozen sorted
image
        existingFrozen = dir([app.pathFrozen '\*.png']); % Count number of existing files
in the folder
        numFrozen = size(existingFrozen,1);
        numFrozen = numFrozen + 1;
        outputBaseFileName = sprintf('Frozen%i.png',numFrozen); % Create output
filename by adding one to total number of existing files
        outputFullFileName = fullfile(app.pathFrozen, outputBaseFileName);
        copyfile(app.CurrentImageToSort, outputFullFileName); % Copy file from source
folder to target folder and rename
        updateNextImage(app, app.i);
        end

        % Button pushed function: NotSureButton
        function NotSureButtonPushed(app, event)
        updateLastNotSure(app, app.CurrentImageToSort); % Update last NotSure sorted
image
        existingNotSure = dir([app.pathNotSure '\*.png']); % Count number of existing
files in the folder
        numNotSure = size(existingNotSure,1);
        numNotSure = numNotSure + 1;
        outputBaseFileName = sprintf('NotSure%i.png',numNotSure); % Create output
filename by adding one to total number of existing files
        outputFullFileName = fullfile(app.pathNotSure, outputBaseFileName);
        copyfile(app.CurrentImageToSort, outputFullFileName); % Copy file from source
folder to target folder and rename
        updateNextImage(app, app.i);
        end

        % Key press function: UIFigure
        function UIFigureKeyPress(app, event)
        key = event.Key;
        switch key
        case 'rightarrow' %right arrow for liquid
        updateLastLiquid(app, app.CurrentImageToSort); % Update last liquid sorted
image
        existingLiquid = dir([app.pathLiquid '\*.png']); % Count number of existing files
in the folder
        numLiquid = size(existingLiquid,1);
        numLiquid = numLiquid + 1;
        outputBaseFileName = sprintf('Liquid%i.png',numLiquid); % Create output
filename by adding one to total number of existing files
```

```
            outputFullFileName = fullfile(app.pathLiquid, outputBaseFileName);
            copyfile(app.CurrentImageToSort, outputFullFileName); % Copy file from source
folder to target folder and rename
            updateNextImage(app, app.i);
            case 'leftarrow' %left arrow for frozen
            %app.FrozenButtonPushed(app, event)
            updateLastFrozen(app, app.CurrentImageToSort); % Update last Frozen sorted
image
            existingFrozen = dir([app.pathFrozen '\*.png']); % Count number of existing files
in the folder
            numFrozen = size(existingFrozen,1);
            numFrozen = numFrozen + 1;
            outputBaseFileName = sprintf('Frozen%i.png',numFrozen); % Create output
filename by adding one to total number of existing files
            outputFullFileName = fullfile(app.pathFrozen, outputBaseFileName);
            copyfile(app.CurrentImageToSort, outputFullFileName); % Copy file from source
folder to target folder and rename
            updateNextImage(app, app.i);
            case 'uparrow' %up arrow for not sure
            %app.NotSureButtonPushed(app, event)
            updateLastNotSure(app, app.CurrentImageToSort); % Update last NotSure sorted
image
            existingNotSure = dir([app.pathNotSure '\*.png']); % Count number of existing
files in the folder
            numNotSure = size(existingNotSure,1);
            numNotSure = numNotSure + 1;
            outputBaseFileName = sprintf('NotSure%i.png',numNotSure); % Create output
filename by adding one to total number of existing files
            outputFullFileName = fullfile(app.pathNotSure, outputBaseFileName);
            copyfile(app.CurrentImageToSort, outputFullFileName); % Copy file from source
folder to target folder and rename
            updateNextImage(app, app.i);
            end
            end

            % Changes arrangement of the app based on UIFigure width
            function updateAppLayout(app, event)
            currentFigureWidth = app.UIFigure.Position(3);
            if(currentFigureWidth <= app.onePanelWidth)
            % Change to a 3x1 grid
            app.GridLayout.RowHeight = {880, 880, 880};
            app.GridLayout.ColumnWidth = {'1x'};
            app.CenterPanel.Layout.Row = 1;
            app.CenterPanel.Layout.Column = 1;
            app.LeftPanel.Layout.Row = 2;
```

```
            app.LeftPanel.Layout.Column = 1;
            app.RightPanel.Layout.Row = 3;
            app.RightPanel.Layout.Column = 1;
        elseif (currentFigureWidth > app.onePanelWidth && currentFigureWidth <=
app.twoPanelWidth)
            % Change to a 2x2 grid
            app.GridLayout.RowHeight = {880, 880};
            app.GridLayout.ColumnWidth = {'1x', '1x'};
            app.CenterPanel.Layout.Row = 1;
            app.CenterPanel.Layout.Column = [1,2];
            app.LeftPanel.Layout.Row = 2;
            app.LeftPanel.Layout.Column = 1;
            app.RightPanel.Layout.Row = 2;
            app.RightPanel.Layout.Column = 2;
        else
            % Change to a 1x3 grid
            app.GridLayout.RowHeight = {'1x'};
            app.GridLayout.ColumnWidth = {248, '1x', 249};
            app.LeftPanel.Layout.Row = 1;
            app.LeftPanel.Layout.Column = 1;
            app.CenterPanel.Layout.Row = 1;
            app.CenterPanel.Layout.Column = 2;
            app.RightPanel.Layout.Row = 1;
            app.RightPanel.Layout.Column = 3;
        end
    end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.AutoResizeChildren = 'off';
            app.UIFigure.Position = [100 100 922 880];
            app.UIFigure.Name = 'MATLAB App';
            app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout,
true);
            app.UIFigure.KeyPressFcn = createCallbackFcn(app, @UIFigureKeyPress, true);

            % Create GridLayout
            app.GridLayout = uigridlayout(app.UIFigure);
```

```
            app.GridLayout.ColumnWidth = {248, '1x', 249};
            app.GridLayout.RowHeight = {'1x'};
            app.GridLayout.ColumnSpacing = 0;
            app.GridLayout.RowSpacing = 0;
            app.GridLayout.Padding = [0 0 0 0];
            app.GridLayout.Scrollable = 'on';

            % Create LeftPanel
            app.LeftPanel = uipanel(app.GridLayout);
            app.LeftPanel.Layout.Row = 1;
            app.LeftPanel.Layout.Column = 1;

            % Create LastFrozenAxes
            app.LastFrozenAxes = uiaxes(app.LeftPanel);
            title(app.LastFrozenAxes, 'Last Frozen')
            xlabel(app.LastFrozenAxes, 'X')
            ylabel(app.LastFrozenAxes, 'Y')
            app.LastFrozenAxes.Position = [6 525 219 185];

            % Create FrozenimagepathTextAreaLabel
            app.FrozenimagepathTextAreaLabel = uilabel(app.LeftPanel);
            app.FrozenimagepathTextAreaLabel.HorizontalAlignment = 'right';
            app.FrozenimagepathTextAreaLabel.Position = [69 469 106 22];
            app.FrozenimagepathTextAreaLabel.Text = 'Frozen image path';

            % Create FrozenimagepathTextArea
            app.FrozenimagepathTextArea = uitextarea(app.LeftPanel);
            app.FrozenimagepathTextArea.Position = [20 399 205 68];

            % Create FrozenimagepathselectButton
            app.FrozenimagepathselectButton = uibutton(app.LeftPanel, 'push');
            app.FrozenimagepathselectButton.ButtonPushedFcn = createCallbackFcn(app,
@FrozenimagepathselectButtonPushed, true);
            app.FrozenimagepathselectButton.Position = [42 368 150 22];
            app.FrozenimagepathselectButton.Text = 'Frozen image path select';

            % Create CenterPanel
            app.CenterPanel = uipanel(app.GridLayout);
            app.CenterPanel.Layout.Row = 1;
            app.CenterPanel.Layout.Column = 2;

            % Create NextImageAxes
            app.NextImageAxes = uiaxes(app.CenterPanel);
            title(app.NextImageAxes, 'Next Image')
            xlabel(app.NextImageAxes, 'X')
```

```
        ylabel(app.NextImageAxes, 'Y')
        app.NextImageAxes.Position = [30 137 370 263];

        % Create LastNotSureAxes
        app.LastNotSureAxes = uiaxes(app.CenterPanel);
        title(app.LastNotSureAxes, 'Last Not Sure')
        xlabel(app.LastNotSureAxes, 'X')
        ylabel(app.LastNotSureAxes, 'Y')
        app.LastNotSureAxes.Position = [95 678 234 196];

        % Create LoadImagesButton
        app.LoadImagesButton = uibutton(app.CenterPanel, 'push');
        app.LoadImagesButton.ButtonPushedFcn = createCallbackFcn(app,
@LoadImagesButtonPushed, true);
        app.LoadImagesButton.Position = [98 16 100 30];
        app.LoadImagesButton.Text = 'Load Images';

        % Create NotSureButton
        app.NotSureButton = uibutton(app.CenterPanel, 'push');
        app.NotSureButton.ButtonPushedFcn = createCallbackFcn(app,
@NotSureButtonPushed, true);
        app.NotSureButton.Position = [165 487 101 54];
        app.NotSureButton.Text = 'Not Sure Button';

        % Create StartfromSpinnerLabel
        app.StartfromSpinnerLabel = uilabel(app.CenterPanel);
        app.StartfromSpinnerLabel.HorizontalAlignment = 'right';
        app.StartfromSpinnerLabel.Position = [269 37 58 22];
        app.StartfromSpinnerLabel.Text = 'Start from';

        % Create StartfromSpinner
        app.StartfromSpinner = uispinner(app.CenterPanel);
        app.StartfromSpinner.ValueChangedFcn = createCallbackFcn(app,
@StartfromSpinnerValueChanged, true);
        app.StartfromSpinner.Position = [248 16 100 22];

        % Create CurrentImageNameEditFieldLabel
        app.CurrentImageNameEditFieldLabel = uilabel(app.CenterPanel);
        app.CurrentImageNameEditFieldLabel.HorizontalAlignment = 'right';
        app.CurrentImageNameEditFieldLabel.Position = [165 399 118 22];
        app.CurrentImageNameEditFieldLabel.Text = 'Current Image Name';

        % Create CurrentImageNameEditField
        app.CurrentImageNameEditField = uieditfield(app.CenterPanel, 'text');
        app.CurrentImageNameEditField.Position = [284 399 100 22];
```

```
        % Create FrozenButton
        app.FrozenButton = uibutton(app.CenterPanel, 'push');
        app.FrozenButton.ButtonPushedFcn = createCallbackFcn(app,
@FrozenButtonPushed, true);
        app.FrozenButton.Position = [77 430 89 58];
        app.FrozenButton.Text = 'Frozen Button';

        % Create LiquidButton
        app.LiquidButton = uibutton(app.CenterPanel, 'push');
        app.LiquidButton.ButtonPushedFcn = createCallbackFcn(app,
@LiquidButtonPushed, true);
        app.LiquidButton.Position = [265 430 88 58];
        app.LiquidButton.Text = 'Liquid Button';

        % Create SourceImagePathTextAreaLabel
        app.SourceImagePathTextAreaLabel = uilabel(app.CenterPanel);
        app.SourceImagePathTextAreaLabel.HorizontalAlignment = 'right';
        app.SourceImagePathTextAreaLabel.Position = [10 102 108 22];
        app.SourceImagePathTextAreaLabel.Text = 'Source Image Path';

        % Create SourceImagePathTextArea
        app.SourceImagePathTextArea = uitextarea(app.CenterPanel);
        app.SourceImagePathTextArea.Editable = 'off';
        app.SourceImagePathTextArea.Position = [128 66 292 60];

        % Create NotSureimagepathTextAreaLabel
        app.NotSureimagepathTextAreaLabel = uilabel(app.CenterPanel);
        app.NotSureimagepathTextAreaLabel.HorizontalAlignment = 'right';
        app.NotSureimagepathTextAreaLabel.Position = [13 634 116 22];
        app.NotSureimagepathTextAreaLabel.Text = 'Not Sure image path';

        % Create NotSureimagepathTextArea
        app.NotSureimagepathTextArea = uitextarea(app.CenterPanel);
        app.NotSureimagepathTextArea.Position = [134 611 195 68];

        % Create NotSureimagepathselectButton
        app.NotSureimagepathselectButton = uibutton(app.CenterPanel, 'push');
        app.NotSureimagepathselectButton.ButtonPushedFcn = createCallbackFcn(app,
@NotSureimagepathselectButtonPushed, true);
        app.NotSureimagepathselectButton.Position = [153 580 160 22];
        app.NotSureimagepathselectButton.Text = 'Not Sure image path select';

        % Create RightPanel
        app.RightPanel = uipanel(app.GridLayout);
```

```
            app.RightPanel.Layout.Row = 1;
            app.RightPanel.Layout.Column = 3;

            % Create LastLiquidAxes
            app.LastLiquidAxes = uiaxes(app.RightPanel);
            title(app.LastLiquidAxes, 'Last Liquid')
            xlabel(app.LastLiquidAxes, 'X')
            ylabel(app.LastLiquidAxes, 'Y')
            app.LastLiquidAxes.Position = [7 525 219 185];

            % Create LiquidimagepathTextAreaLabel
            app.LiquidimagepathTextAreaLabel = uilabel(app.RightPanel);
            app.LiquidimagepathTextAreaLabel.HorizontalAlignment = 'right';
            app.LiquidimagepathTextAreaLabel.Position = [76 469 100 22];
            app.LiquidimagepathTextAreaLabel.Text = 'Liquid image path';

            % Create LiquidimagepathTextArea
            app.LiquidimagepathTextArea = uitextarea(app.RightPanel);
            app.LiquidimagepathTextArea.Editable = 'off';
            app.LiquidimagepathTextArea.Position = [17 399 219 68];

            % Create LiquidimagepathselectButton
            app.LiquidimagepathselectButton = uibutton(app.RightPanel, 'push');
            app.LiquidimagepathselectButton.ButtonPushedFcn    =    createCallbackFcn(app,
@LiquidimagepathselectButtonPushed, true);
            app.LiquidimagepathselectButton.Position = [54 368 145 22];
            app.LiquidimagepathselectButton.Text = 'Liquid image path select';

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
            end
            end

            % App creation and deletion
            methods (Access = public)

            % Construct app
            function app = INP_Image_Sorter

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)
```

```
if nargout == 0
clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
```

Appendix F

Copyright permissions

Part of Chapter 1 was published in an Annual Reviews journal and is included in this thesis with permission on May 27, 2021. The Annual Review copyright policy (https://www.annualreviews.org/page/about/copyright-and-permissions) states: "Annual Reviews Authors: There is no need to obtain permission from Annual Reviews for the use of your own work(s). Our copyright transfer agreement provides you with all the necessary permissions. Our copyright transfer agreement provides: "…The nonexclusive right to use, reproduce, distribute, perform, update, create derivatives, and make copies of the work (electronically or in print) in connection with the author's teaching, conference presentations, lectures, and publications, provided proper attribution is given…""

Part of Chapter 2 was published in an American Chemical Society (ACS) journal and is included in this thesis with permission on May 27, 2021. The ACS copyright policy states: "Anyone who is doing personal, noncommercial research may view, print material, and temporarily store materials from this site, provided that ACS's copyright notices are not removed and that the reuse of material does not conflict with any of the prohibited uses.". "Reuse/Republication of the Entire Work in Theses or Collections: Authors may reuse all or part of the Submitted, Accepted or Published Work in a thesis or dissertation that the author writes and is required to submit to satisfy the criteria of degree- granting

institutions. Such reuse is permitted subject to the ACS' "Ethical Guidelines to Publication of Chemical Research"; the author should secure written confirmation (via letter or email) from the respective ACS journal editor(s) to avoid potential conflicts with journal prior publication/embargo policies. Appropriate citation of the Published Work must be made. If the thesis or dissertation to be published is in electronic format, a direct link to the Published Work must also be included using the ACS Articles on Request author-directed link."

Parts of Chapters 3 and 4 were published in an MDPI journal and is included in this thesis with permission on Jun 08, 2021. MDPI in an open access journal and their copyright policy states: "Articles published in *Micromachines* will be Open-Access articles distributed under the terms and conditions of the Creative Commons Attribution License (CC BY). The copyright is retained by the author(s). MDPI will insert the following note at the end of the published text: © 2021 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/)."

Part of Chapter 5 adapted with permission from Roy, P., Hill, T. C. J., Mael, L. E., Mehndiratta, L., Peiker, G., House, M. L., Grassian, V. H., Demott, P. J., Dutcher, C. S. (2021) "Ice nucleating activity and particle morphology of bulk seawater and sea surface microlayer", ACS Earth and Space Chemistry, submitted for publication. Unpublished work, copyright 2021 American Chemical Society.

Figure 3-18 is an original illustration created using 2D convolutional neural network visualization tool (URL https://www.cs.ryerson.ca/~aharley/vis/fc/flat.html) by permission from Adam W. Harley, "An interactive node-link visualization of convolutional neural networks", in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2015. Springer Verlag, pp. 867–877. doi: 10.1007/978-3-319-27857-5_77. Copyright 2015, Springer Nature.

Figure 4-1 is adapted by permission from Roeters, S. J.; Golbek, T. W.; Bregnhøj, M.; Drace, T.; Alamdari, S.; Roseboom, W.; Kramer, G.; Šantl-Temkiv, T.; Finster, K.; Pfaendtner, J.; Woutersen, S.; Boesen, T.; Weidner, T. Ice-Nucleating Proteins Are Activated by Low Temperatures to Control the Structure of Interfacial Water. Nature Communications. 2021, 12 (1), 1–9. https://doi.org/10.1038/s41467-021-21349-3. Copyright 2021, The Author(s) (CC BY 4.0).