

Extensions to Stimulated Raman Scattering Microscopy For The Study of Dynamic Metabolic  
Processes in Live Cells

By

Gabriel Francis Dorlhiac

A dissertation submitted in partial satisfaction of the  
requirements for the degree of

Doctor of Philosophy

in

Biophysics

in the  
Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Markita Landry, Co-chair

Professor Aaron Streets, Co-chair

Professor Na Ji

Professor Wenjun Zhang

Fall 2022



## Abstract

### Extensions to Stimulated Raman Scattering Microscopy For The Study of Dynamic Metabolic Processes in Live Cells

by

Gabriel Dorlhiac

Doctor of Philosophy in Biophysics

University of California, Berkeley

Professor Markita Landry, Co-Chair; Professor Aaron Streets, Co-Chair

The fundamental properties of the stimulated Raman scattering (SRS) interaction make it attractive for application in biological microscopy. The signal is linear in both the concentration of scatterers, as well as the power of both excitation sources used. In turn, this allows for quantitative concentration measurements to be made within a sample. Furthermore, the SRS spectrum, an exact match to the spontaneous Raman spectrum, is chemically specific. I.e., individual molecules can be specifically and uniquely identified by their spectra. As a result, in principle, SRS microscopy allows for quantitative label-free measurements to be made of biological molecules.

In practice, however, outside of a few applications, SRS microscopy is neither widely adopted nor applied in biology. It is often thought that the specificity of fluorescence cannot be achieved in a label-free manner with SRS due to the complexity of the biological systems under investigation. Additionally, as the standard implementation of SRS microscopy requires laser scanning for the formation of an image, fundamental limits on acquisition speed are thought to make the application of SRS microscopy to the study of dynamic processes challenging, if not infeasible.

The following dissertation presents new experimental and computational methods to broaden the applications of SRS imaging within the life sciences. An overview of the fabrication of the standard SRS imaging system is provided, with attention to the implications of certain design choices. This helps establish the limits within which the SRS microscopy technique currently operates, as well as make the technique more accessible to a general audience. Best practices for performing measurements and analyzing the results are presented. The application of SRS microscopy to the imaging of small molecule neurotransmitters, in a specific manner, is demonstrated, highlighting the capability of SRS microscopy to achieve the specificity of fluorescence imaging. Finally, extensions to the standard imaging system are presented which lay the groundwork for future applications with faster acquisition speeds. In aggregate, the work presented in this dissertation serves to extend the range of problems to which SRS microscopy can be applied in biology.

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>1</b>  |
| 1.0.1    | References . . . . .   | 2         |
| <b>2</b> | <b>Construction of an SRS Microscope</b>   | <b>4</b>  |
| 2.1      | Introduction . . . . .   | 4         |
| 2.2      | Hardware . . . . .   | 7         |
| 2.2.1    | General construction . . . . .   | 7         |
| 2.2.2    | Pulse shaping and spectral focusing . . . . .  | 10        |
| 2.2.3    | Detection characteristics and noise model . . . . .  | 20        |
| 2.3      | Software . . . . .   | 25        |
| 2.4      | References . . . . .   | 25        |
| 2.5      | Appendix: Parts list . . . . .   | 28        |
| <b>3</b> | <b>Image processing and information quantification</b>   | <b>31</b> |
| 3.1      | Introduction . . . . .   | 31        |
| 3.2      | Results . . . . .  | 32        |
| 3.2.1    | The importance of preprocessing . . . . .  | 32        |
| 3.2.2    | Cellular and sub-cellular segmentation . . . . .   | 35        |
| 3.2.3    | Multiscale entropy . . . . .   | 38        |
| 3.3      | Discussion . . . . .   | 43        |
| 3.4      | Materials and Methods . . . . .  | 44        |
| 3.4.1    | HEK293T cell culture . . . . .   | 44        |
| 3.4.2    | Undifferentiated induced-pluripotent stem cells (iPSCs) . . . . .                                | 44        |
| 3.4.3    | SRS imaging . . . . .  | 44        |
| 3.5      | References . . . . .   | 45        |
| <b>4</b> | <b>Neurotransmitter Isotopologue Imaging</b>   | <b>47</b> |
| 4.1      | Abstract . . . . .   | 47        |
| 4.2      | Introduction . . . . .   | 48        |
| 4.3      | Results . . . . .  | 52        |
| 4.3.1    | SRS microscopy allows observation of uptake and release of d <sub>4</sub> -DA . . . . .          | 52        |
| 4.3.2    | PC12 cells display cell-to-cell heterogeneity of d <sub>4</sub> -DA uptake and release . . . . . | 54        |
| 4.3.3    | d <sub>6</sub> -GABA uptake is also observable with SRS microscopy . . . . .                     | 58        |
| 4.4      | Discussion . . . . .   | 61        |
| 4.5      | Materials and Methods . . . . .  | 63        |

|          |   |            |
|----------|---|------------|
| 4.5.1    | PC12 cell culture . . . . .   | 63         |
| 4.5.2    | Flow cell construction and PC12 cell culture on device . . . . .            | 63         |
| 4.5.3    | Primary hippocampal neuron dissociation and culture . . . . .               | 64         |
| 4.5.4    | SRS microscope optical setup . . . . .                                      | 64         |
| 4.5.5    | Deuterated neurotransmitter imaging . . . . .                               | 64         |
| 4.5.6    | Two-photon excited fluorescence imaging of FFN102 . . . . .                 | 65         |
| 4.5.7    | Image processing . . . . .  | 65         |
| 4.6      | Acknowledgements . . . . .  | 66         |
| 4.7      | Funding . . . . .   | 66         |
| 4.8      | References . . . . .  | 66         |
| 4.9      | Supplementary Material . . . . .  | 74         |
| 4.10     | Appendix . . . . .  | 79         |
| 4.10.1   | A note on cell culture . . . . .  | 79         |
| 4.10.2   | A note on sensitivity . . . . .   | 82         |
| 4.10.3   | References (Appendix) . . . . .   | 84         |
| <b>5</b> | <b>Compressed sensing SRS (CSSRS) microscopy</b>                            | <b>86</b>  |
| 5.1      | Introduction . . . . .  | 86         |
| 5.2      | Results . . . . .   | 88         |
| 5.2.1    | DMD-based detection supports multiple experimental modes . . . . .          | 88         |
| 5.2.2    | Mathematical description of DMD-based CSSRS . . . . .                       | 91         |
| 5.2.3    | Compressed spectrum reconstruction . . . . .                                | 94         |
| 5.2.4    | <i>in silico</i> image reconstruction from spectral focusing data . . . . . | 97         |
| 5.2.5    | Extending the model to general images . . . . .                             | 100        |
| 5.3      | Discussion . . . . .  | 103        |
| 5.4      | Additional methods . . . . .  | 105        |
| 5.4.1    | Spectrum and image retrieval . . . . .                                      | 105        |
| 5.4.2    | SRS imaging of plastic beads . . . . .                                      | 106        |
| 5.5      | References . . . . .  | 107        |
| 5.6      | Supplementary Material . . . . .  | 112        |
| 5.6.1    | Additional parts list . . . . .   | 114        |
| <b>6</b> | <b>Conclusion</b>   | <b>115</b> |
| <b>A</b> | <b><i>srsmicro</i> API</b>  | <b>117</b> |

# Chapter 1

## Introduction

In the approximately fifteen years since its introduction, stimulated Raman scattering (SRS) microscopy has established itself as a useful technique for biomedical imaging. By using molecular vibrations to provide image contrast, SRS microscopy can be employed to study biological systems without the need for the exogenous dyes required by fluorescence microscopy. The chemical specificity of the Raman, and as a result, the SRS, spectrum makes SRS microscopy a powerful tool for the study of certain classes of metabolic targets. In particular, it has been widely applied to the imaging of large biomolecules, such as proteins, lipids, and carbohydrates, in aggregate. Using just these targets, SRS microscopy can be used to emulate histology without the use of dyes (Freudiger et al., 2012; Yang, Chen, and Ji, 2017), even in clinical contexts (Orringer et al., 2017; Lee et al., 2021). An overview of the application of SRS to the study of lipid metabolism is presented in our review paper (Gupta, Dorlhiac, and Streets, 2019). Furthermore, introduction of isotopes can be used to study the total metabolic activity of cells, through monitoring of the synthesis of new macromolecules (Shi et al., 2018; Wei et al., 2019). Yet, despite the theoretically unique nature of any chemical's SRS spectrum, the complexity of the cellular environment makes identifying the specific spectral signatures of a particular molecule challenging. This means that while label-free in principle, SRS microscopy is unable to reach the level of specificity of fluorescence without the introduction of exogenous tags or dyes.

Furthermore, while the SRS process provides an orders of magnitude increase in signal compared to the spontaneous process (McCamant, Kukura, and Mathies, 2003; Freudiger et al., 2008), allowing for imaging speeds on the order of fluorescence microscopy, the study of dynamic biological processes remains challenging. Despite a handful of demonstrations of SRS microscopy with video-rate imaging speeds (see e.g., Saar et al., 2010; Liao et al., 2015), the technological challenges behind their implementation have meant they have not been widely adopted, even as the field of experts in the technique continues to grow. As a result, the vast majority of applications of SRS microscopy in biology have remained confined to the study of the slower metabolic processes to which it was originally applied, and which it has become very well-suited to. In this dissertation, new experimental methods are presented with the goal of extending the use of SRS microscopy to the study of specific molecules as well as dynamic processes.

Chapter two begins by introducing the construction of a traditional SRS microscope. Attention is paid to common problems encountered in the construction and maintenance processes, and possible solutions are proposed. Fundamental characteristics of the signal and noise of the detected signal are evaluated. Observations based on these characteristics lay the foundation for future work

introduced in chapter five. Finally, a new pulse characterization algorithm is introduced which is particularly well suited to SRS, and other nonlinear, microscopies. In principle it allows for accurate temporal and spectral characterization of optical pulses with minimal modification of the experimental setup. In turn, this allows for the highest spectral resolution measurements to be made.

Chapter three introduces a number of computational procedures and metrics for analysis of SRS images, of any type. The importance of preprocessing steps is discussed. Following this, simple algorithms are introduced which perform well in extracting data across a wide-variety of common SRS images. Finally, a new metric is introduced for evaluating the information content in images. When applied to SRS micrographs, this allows for performance evaluation of the imaging system, and can be used to guide decisions regarding background signal reduction.

Chapter four presents a novel experimental method for the imaging of small molecule neurotransmitters. By using isotopologues, SRS signal corresponding to specific neurotransmitters can be detected in the absence of contribution from the majority of biomolecules present in a living system. This strategy, on its own, is not new; however, it has thus far only been applied when making bulk measurements of, e.g. all proteins, all lipids. The methodology in chapter four allows for making specific measurements of a **single** target molecule, similarly to how this is done with fluorescence microscopy, but with comparatively minimal system perturbation, maintaining the (quasi-)label-free nature of SRS microscopy. An analogous approach can therefore be adopted for other metabolites with similar concentration profiles, which have thus far been difficult to investigate by other means.

Motivated by observations made in chapter two, chapter five lays the theoretical groundwork for a new imaging modality termed compressed sensing SRS, or CSSRS. While the fundamental tradeoff between SNR and acquisition speed in SRS cannot be eliminated, by making use of the structure of images and sparse, multiplexed, sampling, images can be acquired more rapidly. This offers a way to extend the use of SRS microscopy to the study of dynamic processes in living systems. Furthermore, the characteristics of the detection mechanism have implications for increasing the signal-to-noise ratio (SNR) of measurements, particularly when using low optical powers, as is necessary for avoiding phototoxicity in sensitive samples. *in silico* experimental results are presented under simulated experimental noise, as well as using actual SRS experimental data.

## 1.0.1 References

Freudiger, C. W., Min, W., Saar, B. G., Lu, S., Holtom, G. R., Chengwei, H., Tsai, J. C., Kang, J. X., and Xie, X. S. (2008), "Label-free biomedical imaging with high sensitivity by stimulated Raman scattering microscopy", Vol. 322 No. 5909, pp. 1857-1861..

Freudiger, C. W., Pfannl, R., Orringer, D. A., Saar, B. G., Ji, M., Zeng, Q., Ottoboni, L., Ying, W., Waeber, C., Sims, J. R., De Jager, P. L., Sagher, O., Philbert, M. A., Xu, X., Kesari, S., Xie, X. S., Young, G. S. (2012) "Multicolored stain-free histopathology with coherent Raman imaging", *Laboratory Investigation*, Vol. 92, pp. 1492-1502.

Gupta, A., Dorlhiac, G. F., and Streets, A. M. (2019) "Quantitative imaging of lipid droplets in single cells", *Analyst*, Vol. 144, No. 3, pp. 753-765.

Lee, M., Herrington, C. S., Ravindra, M., Sepp, K., Davies, A., Hulme, A. N., and Brunton, V. G. (2021) “Recent advances in the use of stimulated Raman scattering in histopathology”, *Analyst*, Vol. 146, pp. 789-802.

Liao, C-S., Slipchenko, M. N., Wang, P., Li, J., Lee, S-Y., Oglesbee, R. A., and Cheng, J-X. (2015) “Microsecond scale vibrational spectroscopic imaging by multiplex stimulated Raman scattering microscopy”, *Light: Sciences & Applications*, Vol. 4 No. e265.

McCamant, D. W., Kukura, P., and Mathies R. A. (2003) “Femtosecond broadband stimulated Raman: A new approach for high-performance vibrational spectroscopy” *Appl. Spectrosc.*, Vol. 57, pp. 1317-1323.

Orringer, D. A., Pandian, B., Niknafs, Y. S., Hollon, T. C., Boyle, J., Lewis S., Garrard, M., Hervey-Jumper, S. L., Garton, H. J. L., Maher, C. O., Heth, J. A., Sagher, O., Wilkinson, D. A., Snuderl, M., Venneti, S., Ramkissoon, S. H., McFadden, K. A., Fisher-Hubbard, A., Lieberman, A. P., Johnson, T. D., Xie, X. S., Trautman, J. K., Freudiger, C. W., and Camelo-Piragua, S. (2017), “Rapid intraoperative histology of unprocessed surgical specimens via fibre-laser-based stimulated Raman scattering microscopy”, *Nat Biomed Eng*, Vol. 1.

Saar, B., G., Freudiger, C. W., Reichman, J., Stanley, C. M., Holtom, G. R., and Xie, X. S. (2010), “Video-rate molecular imaging in vivo with stimulated Raman scattering”, *Science*, Vol. 330 No. 6009, pp. 1368-1370.

Shi, L., Zheng, C., Shen, Y., Chen, Z., Silveira, E. S., Zhang, L., Wei, M., Liu, C., de Sena-Tomas, C., Targoff, K., and Min, W. (2018) “Optical imaging of metabolic dynamics in animals”, *Nature Communications*, Vol. 9.

Wei, M., Shi, L., Shen, Y., Min, W. (2019) “Volumetric chemical imaging by clearing enhanced stimulated Raman scattering microscopy”, *PNAS*, Vol. 116 No. 14.

Yang, Y., Chen, L., Ji, M. (2017) “Stimulated Raman scattering microscopy for rapid brain tumor histology”, *Journal of Innovative Optical Health Sciences*, Vol. 10 No. 05.



# Chapter 2

## Construction of an SRS Microscope

\*This chapter contains results which have not been previously published.

### 2.1 Introduction

While fluorescence provides high sensitivity, in most cases, exogenous dyes (or proteins) need to be introduced to biological systems to employ it. Vibrations are an intrinsic feature of all molecules, and are, in addition, chemically specific. I.e., the frequencies at which one molecule vibrates are distinct from those of another. These vibrations can be probed in a label-free fashion as an alternative to using fluorescence-based methods. The two most common probes of molecular vibrations are infrared (IR) and Raman spectroscopy. In the former, light in the mid to far IR range ( $\lambda = 3\text{-}100\ \mu\text{m}$ ) with energies corresponding to the energies of excited molecular vibrational states is absorbed by the sample and the absorbance vs frequency is indicative of the types and number of bonds present. In the latter, light is inelastically scattered by the vibrations in the sample, and the energy gain or loss of the scattered light contains the information about the types of bonds. When the molecule is excited to a higher vibrational state from the ground state, the light loses energy, and the process is referred to as Stokes scattering. On the other hand, if the molecule is already in an excited vibrational state, it can transfer energy to the light, and de-excite back to the ground state. This is referred to as anti-Stokes scattering. The relative intensity of the two processes is governed by Boltzmann statistics, and at room temperature the Stokes process is more common, as most molecules are in their ground state (McCreery, 2005).

Both IR and Raman scattering can be implemented for imaging purposes (see e.g., Messerschmidt and Harthcock, 1988; Turrel and Corset, 1996); however, Raman scattering has a number of advantages over IR in microscopy applications. One of these is spatial resolution. The spatial resolution in optical microscopy is fundamentally limited by the wavelength of light used. Wavelengths of IR light are much longer than those of the visible light typically used for fluorescence, which provides an immediate lower bound on the achievable spatial resolution. Mid-IR optics are also costly, and the difficulty of working with invisible light does not guarantee that this limit is reachable. Raman scattering, on the other hand, can be observed at wavelengths in the visible and near IR, giving it comparable spatial resolution to fluorescent imaging. Another challenge of IR is due to the strong absorption of water over the wavelengths that are typically of interest. This results in strong signal attenuation, and the broadness of the water peak may totally obscure some

of the relevant vibrations in signal. Indeed, IR spectra are generally acquired in D<sub>2</sub>O if the sample is required to be aqueous (Stuart, 2004). Vibrations of water are also Raman active, but since it is a scattering process, the intensity of light is not greatly attenuated by the interaction with the sample. Use of near IR excitation sources in Raman imaging can even improve penetration depth in tissue.

That being said, Raman scattering has a drawback for imaging – it is an inherently weak process. Raman scattering cross-sections are low, with on the order of 1 in 10<sup>6-7</sup> photons being scattered inelastically. Practically, this means that long acquisition times, from seconds to minutes, are often required to collect a spectrum. This impacts the ability of Raman scattering to probe time-dependent processes. Greater excitation powers may be used, but this results in a tradeoff between phototoxicity and acquisition times. A number of techniques, however, can be used to increase the intrinsic Raman scattering signal. Among these are the coherent Raman techniques.

Coherent Raman scattering refers to a set of nonlinear optical processes that have Raman or Raman-like frequency dependence, and so provide similar information to spontaneous Raman scattering, but have a large signal increase relative to the spontaneous process (Mukamel, 1999). Coherent anti-Stokes Raman scattering (CARS), and stimulated Raman scattering (SRS) are the two most commonly implemented of these techniques. These processes yield potential signal increases of 6 to 7 orders of magnitude over spontaneous Raman scattering, allowing them to be developed as imaging tools with acquisition speeds comparable to confocal scanning microscopy (McCamant, Kukura, and Mathies, 2003; Freudiger et al., 2008).

Both CARS and SRS are third order processes that require excitation with lasers of two different frequencies, which will be referred to as the pump (frequency  $\omega_p$ ) and Stokes (frequency  $\omega_s$ ). As they are third order processes, high peak powers are required, and so picosecond (ps) or femtosecond (fs) pulsed lasers are generally used. For microscopy, a scanning microscope combined with a high numerical aperture (NA) objective is used to maintain the high peak power density (Cheng, and Xie, 2016). These techniques are classified as coherent because the measured signals produced from different molecules in the excitation volume exhibit a defined phase relation to each other and the incoming fields, in contrast to spontaneous Raman scattering or fluorescence (Mukamel, 1999). The processes are related to the spontaneous process as the choice of input frequencies is chosen such that  $\omega_p - \omega_s = \omega_v$ , where the frequency  $\omega_v$  corresponds to the frequency of a molecular vibration of interest (Mukamel, 1999). Note that for two monochromatic input lasers of frequencies  $\omega_p$  and  $\omega_s$ , only a single vibrational frequency is being probed, in contrast to spontaneous Raman scattering, where an entire spectrum may be acquired simultaneously.

While both techniques provide large signal increases over the spontaneous Raman process, SRS has a number of advantages over CARS, and will be the primary focus of this dissertation. Chief among these is that the SRS signal is proportional to the imaginary part of the third order susceptibility, while CARS is proportional to the square of its magnitude (Mukamel, 1999). This means that the SRS signal has a linear dependence on the concentration of scatterers in the sample, and can therefore more easily be used to make quantitative determinations of sample concentration. Furthermore, CARS has an additional nonresonant electronic signal component that is absent from the SRS process, although due to the nature of SRS detection other background signals can arise, one of which is covered in more depth in chapter 3.

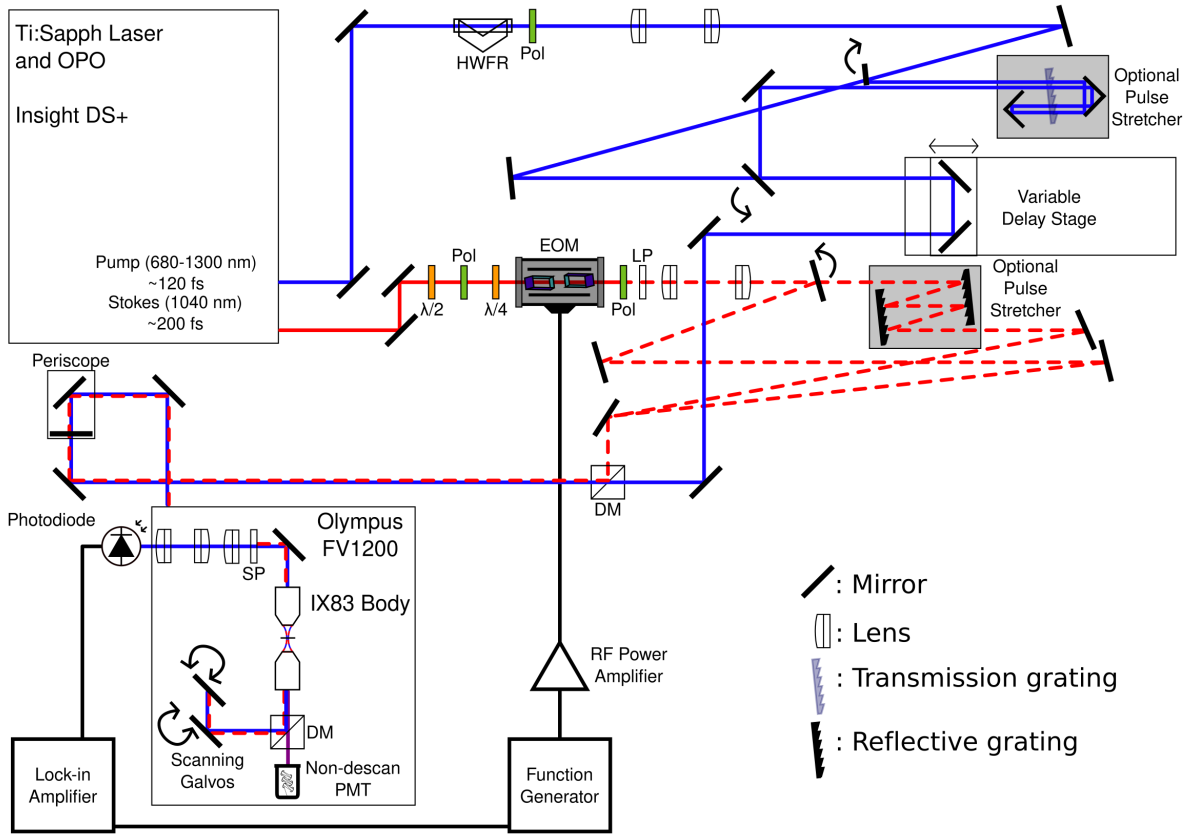
Fundamentally, SRS is a dissipative process. During the interaction, pump photons are converted to Stokes photons, and energy is transferred from light to matter, promoting the molecule to an excited vibrational state. Since no light of a new frequency is created, SRS is often referred to

as a self-heterodyned technique (by spectroscopists), i.e. the signal frequency is mixed with, and at the same frequency as one of the input frequencies. Note that this term should be used with care as heterodyning has a different meaning in the field of quantum optics (Mukamel, 1999). In that field SRS detection would be referred to as homodyne.

The SRS signal can be measured either by the decrease in pump intensity, referred to as stimulated Raman loss (SRL), or as the increase in Stokes intensity, referred to as stimulated Raman gain (SRG) (Cheng, and Xie, 2016). Due to the self-heterodyned (or homodyned) nature of the detection, extracting either the SRL or SRG signal from the background of the input lasers is technologically challenging as the signal frequency is identical to the excitation frequency, in contrast to techniques like CARS or fluorescence. Nonetheless, SRS has been successfully implemented as a microscopy technique and in the past decade has grown into a mature imaging modality. Multiple textbooks and reports have been written on the development of SRS microscopes, and the relevant noise and detection characteristics of them (e.g., Cheng, and Xie, 2016; Cheng, et al., 2021; Manifold, and Fu, 2022). Nonetheless, it is important to cover some of the specifics of the SRS microscope which will be used for the experimental results presented in this dissertation. The remainder of this chapter will be focused on the optical design of the microscope with additional attention paid to two specific aspects: pulse shaping and lock-in detection parameter choice under measurement noise.

## 2.2 Hardware

### 2.2.1 General construction



**Figure 2.1:** Schematic of optical setup.

Solid black lines represent mirrors, black lines with ridges are reflective gratings, with the translucent variant being a transmission grating. Lenses are doublets depicted as unlabeled white components outlined in black. EOM: Electro-optic modulator;  $\lambda/2$ : half waveplate; Pol: polarizer;  $\lambda/4$ : quarter waveplate; LP: long-pass filter; DM: dichroic mirror; SP: short-pass filter.

The optical schematic for the design of the SRS microscope is shown in figure 2.1. A commercial femtosecond laser system, the Insight DS+ from Spectra Physics, is used for both the pump and Stokes beams, operating with a repetition rate of 80 MHz. The Stokes beam is fixed at the fundamental output of the laser, 1040 nm. The pump beam is the output from an internal optical parametric oscillator (OPO), which allows the wavelength of the beam to be tuned between 680 and 1300 nm, although mirrors and filters elsewhere in the optical path limit the usable range to 680-950 nm. This tuning allows for the selection of which vibrational frequency ( $\omega_p - \omega_s = \omega_v$ ) to image at.

The pump beam and Stokes beam average powers are controlled using variable attenuators consisting of a half-wave Fresnel rhomb (pump) or half waveplate (Stokes) and a polarizer. A Fresnel rhomb was chosen for the pump as it provides flatter half-wave rotation over a large wavelength range less expensively than an achromatic half waveplate. The pump beam is passed through a

variable delay stage, i.e. a translation stage mounted retroreflector. The delay stage has 100 nm step precision and repeatability, which is necessary to control temporal coincidence of the two optical pulses. The delay stage is also used for wavenumber tuning when operating in spectral focusing mode (introduced later on). The Stokes beam passes through a quarter waveplate (QWP), an electro-optic modulator, containing a resonant circuit, and then a polarizer. Both beams are expanded using telescopes composed of achromatic doublets so that they ultimately fill the back aperture of the objective to make use of the entire NA of the objective. The beams are then combined on a 950 nm shortpass dichroic mirror (DM), before being passed into the body of the scanning microscope, an Olympus FV1200. A set of galvo mirrors in the microscope is used to raster scan the beams in the X and Y directions, employing a silver mirror for reflecting the beams up to the sample. For CARS, multiphoton fluorescence, and second harmonic imaging, this mirror is replaced by a DM. The frequency-shifted light emitted from these processes is transmitted through the DM, and detected in the epi direction by a non-descanned PMT. The SRS signal is detected in the forward direction as SRL. The Stokes beam is filtered by a shortpass filter and then the pump beam is detected on a large area silicon photodiode the output of which is bandpass filtered and fed through to a lock-in amplifier. A series of lenses after the condenser and before the photodiode are used to reimage the back aperture of the objective onto the detector. The beam arrives at the same point on the back aperture for all the angles scanned by the galvo mirrors. Reimaging this aperture ensures that the beam remains stationary on the detector regardless of the point in the sample that is being imaged.

The intensity loss on the pump beam needs to be separated from the background of the input laser power. In order to achieve this, a modulation and lock-in detection scheme is used. The input intensity of the Stokes beam is modulated from 0 to full amplitude at 10.28 MHz by the combination of QWP, resonant EOM and polarizer. This modulation frequency is chosen for two reasons. First, it is faster than the galvo mirrors are scanning across the sample. Second, MHz modulation frequencies are preferable, as  $1/f$  laser noise typically falls off before 1 MHz (Cheng, et al., 2021). In principle, this allows SRS detection to be shot-noise limited, although for certain optical powers this may not be the case (Slipchenko et al., 2012), as discussed later on.

The result of this modulation process is that, when reaching the detector, the pump beam contains a constant DC background from the input laser intensity, with an AC component at the modulation frequency, corresponding to the SRL signal, because the stimulated Raman process can only occur when both beams are present. This signal is extracted by the lock-in amplifier, and used to construct the image. In this setup, the Stokes beam was modulated in order to detect the SRL signal. In principle, the SRL or SRG signals contain the same information, but practical considerations make detecting the SRL signal simpler, as covered below.

In order to implement this detection scheme the intensity of light reaching the sample needs to be varied at very high frequencies. Optical choppers, which rotate to mechanically block the laser beam are only capable of operation in the kHz regime. Since the image is constructed by raster scanning the laser across the FOV, this would require pixel dwell times on the order of milliseconds, making image acquisition unacceptably slow for biological application. For this reason an EOM was used to modulate the Stokes beam. An EOM makes use of Pockels effect where the refractive index of a material is dependent on the strength of an applied electric field. By applying a voltage across a birefringent crystal exhibiting this effect, we can vary the relative phase delay between the polarization components of any light propagating through it, producing a voltage-dependent waveplate (Kawanishi, 2022). The phase delay is a linear function of the applied voltage, and at a

particular value, we can rotate the polarization of linearly polarized light by 90 degrees. Placing a polarizer at the output of the EOM, the laser beam will then be blocked when the voltage is applied, but allowed to pass otherwise. By varying the applied voltage at MHz frequencies we can therefore modulate the intensity of the light at high speed.

In this design the EOM has been placed on the Stokes line for two reasons. The first is that it is easier to detect the SRL on the pump beam given the wavelengths that are typically employed in our experiments. For most experiments, the wavelength of the pump is tuned between 790 and 900 nm, which falls well within the responsivity range of common Si-based photodiodes. InGaAs detectors, which have better responsivity at 1040 nm, are now more widely available, but remain expensive. Additionally, they are frequently not manufactured in large form factors. As the laser itself is being detected by the photodiode, this has implications for the response time of the photodiode. As the incident power density increases, the photodiode slows down, making detection at high modulation frequencies more challenging (Cheng, and Xie, 2016). Spreading the beam over a large-area photodiode, as are available for Si-based diodes, lowers the power density, reducing this effect.

The second reason to employ the EOM on the Stokes line is that the voltage requirements to induce a given phase delay vary with wavelength. Since the pump beam is broadly tuneable, the correct voltage would need to be calibrated for every output wavelength, complicating the system. It can also be seen that a QWP was used before the EOM. The QWP changes the polarization state of the laser from horizontal to circular. The EOM can then change the polarization state back to horizontal or to vertical by applying a voltage corresponding to a  $\pm\pi/2$  phase delay. This is done because half the voltage is required to induce a  $\pi/2$  vs a  $\pi$  phase delay. It is also easier to purchase or build radio frequency electronics that produce positive and negative voltages than it is to produce larger positive voltages. Even so, construction of high-frequency, high-voltage electronics of any kind is challenging. The original voltage amplifier purchased for this microscope had to be sent back as errors in its construction led to amplification of the mains 60 Hz signal on top of amplification of the driving voltage of the function generator. As a result, the EOM was also replaced with an EOM containing a resonant circuit. This circuit amplifies lower, high-frequency, voltages provided by a small power amplifier driven by an external function generator. Despite these challenges, the advantage of an EOM is that it can in principle achieve 100% modulation depth. Nonetheless, the choice of EOM may be worth reconsidering for future implementations. Unfortunately, as mentioned, optical choppers that mechanically block the beam intermittently, as are commonly used in spectroscopy, can only operate at kHz frequencies, and as such are incompatible with the microscope. However, an AOM may be a more appropriate choice for modulation, even given a slightly lower maximum theoretical modulation depth, due to the lower driving voltages required, as proposed in (Cheng, and Xie, 2016). Such a configuration would also allow for multiple operation modes, such as pulse-picking which can be used for certain types of transient absorption experiments, for example, or with optical amplifiers.

For the coherent Raman processes to take place both the pump and Stokes beams need to arrive at the sample plane with a high degree of spatial and temporal overlap. The spatial overlap is controlled by alignment of the various mirrors in kinematic mounts along the optical path. Temporal overlap is controlled by the variable delay stage on the pump beam, allowing us to add or remove path length. Temporally overlapping in time is required as the path that each beam takes on the table and through the laser is different. While the output of the pump and Stokes beams are synchronized, as the frequency-doubled Stokes beam is used to pump the OPO, an 80 MHz repetition

rate means that there can be up to 6.125 ns of time delay between pulses, equivalent to a relative path difference of over 1.8 m (6.125 lightnanoseconds). The total path through the OPO adds approximately 4 lightnanoseconds to the pump beam path, and so an additional 8 lightnanoseconds is added on that line, in addition to the fine temporal control provided by the delay stage. Both the pump and Stokes beams can optionally pass through two grating based pulse shapers through the use of flip mirrors. These are used for linearly chirping the pulses to perform spectral focusing, covered in depth in the next section. The optical path lengths were designed such that the temporal overlap of the two lasers could be still be controlled using the delay stage regardless of the use of the pulse shapers. I.e. the two beams can be overlapped in time, using the delay stage, if they are both unchirped, both chirped, or if only one of the two is chirped. This allows for the quick transition between multiple imaging modalities.

A parts list for the free-space optics needed to construct the microscope as depicted is provided in the appendix to this chapter.

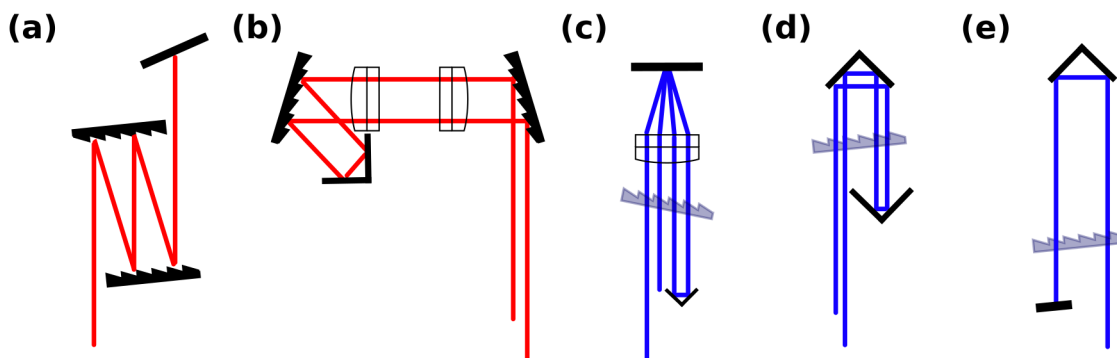
### 2.2.2 Pulse shaping and spectral focusing

In many of the early demonstrations of SRS microscopy, and still continuing on until today, lasers with picosecond pulse durations were used as the excitation sources (Cheng, and Xie, 2016). While shorter pulse durations, on the order of 10s to 100s of femtoseconds, are generally desirable for nonlinear optical interactions, e.g. in multiphoton excited fluorescence, or second harmonic generation microscopies, there is a very good reason to employ longer picoseconds pulses in SRS microscopy. This is because as the duration of a pulse decreases, the spectral bandwidth increases. While a transform-limited 2-7 ps pulse centered at 800 nm will have a spectral bandwidth of 2-7  $\text{cm}^{-1}$  (FWHM), a 100 fs pulse will have a bandwidth closer to 150  $\text{cm}^{-1}$ . In SRS microscopy, the spectral resolution of a measurement will be determined both by the linewidth (homogenous and inhomogenous) of the interrogated transition **and** the bandwidth of the pump and Stokes pulses used as the excitation sources. Given that the linewidth of most Raman-active modes is on the order of 10-15  $\text{cm}^{-1}$  (Lewis, and Edwards, 2001), using a femtosecond pulse duration results in a 10-fold loss of spectral resolution. When tuning the lasers so that the difference (center) frequency matches any given Raman transition, in reality the signal from a 150  $\text{cm}^{-1}$  window centered at that transition will be measured.

At first, this would seem to be a major disadvantage. However, in a number of situations it can be useful. One such scenario is when we are not too concerned about high spectral resolution. In chapter 4 we will introduce deuterated neurotransmitter imaging. The signals arising from deuterated bonds are sufficiently far away from any other Raman transitions that the low spectral resolution provided by a femtosecond laser is not a hinderance. In fact, it is actually advantageous, as the transitions being measured are rather broad compared to the 10-15  $\text{cm}^{-1}$  linewidth of a typical Raman mode. This is due to the overlap of multiple modes with close-lying frequencies. By using femtosecond laser pulses we can simultaneously drive and detect all of these modes, while still being sure that we are far enough away from any others to avoid issues of background signals. The use of femtosecond lasers to improve signal-to-noise in this regime of broad linewidths has previously been investigated and found to indeed be beneficial (Cheng, and Xie, 2016). On the other hand, for typical, spectrally narrow, Raman transitions there is no signal benefit of a femtosecond source over using a picosecond laser source, in addition to providing lower spectral resolution.

This scenario, though, is not enough to warrant the choice of a femtosecond laser over a picosecond laser for SRS imaging in the microscope described above. The use of femtosecond pulse durations, however, facilitates a number of other nonlinear imaging modalities. This allows the SRS microscope, employing femtosecond pulses, to make high quality fluorescence and second harmonic, e.g., measurements with minimal system modifications. Additionally, it is possible to increase the spectral resolution of SRS measurements made with femtosecond pulses, *if* certain pulse shaping strategies are employed. In fact, using shaped femtosecond pulses can lead to an increase in the acquisition speed of a hyperspectral stack. I.e. carefully shaped pulses can be used to more quickly acquire an SRS image at many different wavenumbers for the same FOV. Using picosecond lasers, the center wavelength of one or both of the excitation sources must be tuned each time an image needs to be acquired at a new wavenumber. While top-of-the-line commercial laser systems for microscopy (such as the Insight DS+ used in the microscope) often now include an OPO which can accomplish this quite rapidly, it still takes on the order of multiple seconds to change the wavelength. During the acquisition of many FOVs, at many wavenumbers, this tuning time adds up, and sample motion can become a problem. Additionally, the expensive commercial systems which can accomplish this feat are not available to all researchers.

By employing pulse shaping, femtosecond pulses can be used to avoid needing to tune the laser output over the frequency window dictated by the correlation of the spectra of the pump and Stokes sources. Specifically, the optical frequencies present in a femtosecond pulse can be spread out in time, a process known as chirping. By spreading the chirping the pulses, the relative overlap between the pulses dictates their difference frequency. By changing the overlap one can then change the difference frequency, allowing for the acquisition of images at different wavenumbers as a function of interpulse delay. This strategy of using the interpulse delay between chirped pulses to tune the wavenumber being interrogated is known as spectral focusing (Fu et al., 2013; Upputuri, Gong, and Wang, 2014; Mohseni, Polzer, and Hellerer, 2018). Assuming the acquired chirp is linear, it can be shown that the instantaneous bandwidth of a stretched pulse is equivalent to the bandwidth of an unchirped transform-limited pulse of the same duration (Mohseni, Polzer, and Hellerer, 2018). As a result, the spectral resolution of an SRS microscope using spectral focusing can approach that of a different microscope employing picosecond, instead of femtosecond, duration excitation pulses.





**Figure 2.2:** Schematics of grating-based pulse stretchers

(a) A Treacy style stretcher providing negative dispersion. The four reflections off of the diffraction grating ensure that, when well aligned, there is no spatial chirp. (b) A Martinez style pulse stretcher. The distance between the lenses and the gratings determines whether the stretcher provides positive or negative chirp. A retroreflector is used to offset the beam for uncoupling of the stretched pulse from the input pulse. (c) A folded Martinez style stretcher using a transmission grating. The mirror is placed at the focal distance of the lens, and the distance between the lens and grating determine the stretching factor. (d) A folded Treacy style stretcher using a transmission grating. As with the stretcher in (a) the four passes through the grating ensure that there is no spatial chirp in the stretched output pulses. (e) A Treacy style stretcher with only two passes through the transmission grating. Compared to (e), the retroreflector needs to be twice as far from the grating to achieve the same stretching factor. Unlike the other four stretcher designs, the output pulse of this stretcher has spatial chirp.

There are multiple ways to impart chirp on a pulse (see, e.g., Weiner, 1996). The spectral dispersion provided by glass, or prisms, leads to a frequency-dependent phase delay which delays "bluer" frequencies more than "redder" frequencies. This is known as positive dispersion and is generally viewed as a disadvantage in many nonlinear microscopies. Propagation through the lenses of an imaging system, such as a microscope, for instance, will lead to positive dispersion, and as a result temporal broadening and chirp. This can be a disadvantage for techniques like multiphoton fluorescence microscopy which rely on maintaining the shortest possible pulse duration at the sample; however, introduction of additional glass into the optical path can be used to add the requisite chirp required to perform spectral focusing SRS microscopy.

Alternatively, negative dispersion, where "redder" frequencies are delayed more than "bluer" frequencies, can be applied to a polychromatic laser beam using gratings. Grating-based systems spectrally separate the components of light. As a result, each frequency of light travels a slightly different path. The path length difference is converted into a temporal difference by eventually recombining all of the colors into a single beam using multiple reflections off of, or passes through, a grating(s). For the purposes of spectral focusing, either negative or positive dispersion can be used; however, despite additional alignment sensitivity, we find that grating-based pulse stretchers offer an advantage over glass-based pulse stretchers. This is because the total chirp can more easily be fine-tuned by varying the distance between gratings (or the grating and a mirror), whereas large amounts of glass (10s of cm) would need to be added or removed from the optical path in order to change the dispersion.

A number of grating based pulse stretcher designs are shown in figure 2.2. Figure 2.2a depicts a standard Treacy style pulse stretcher (Treacy, 1969), employing a double-pass through a two-grating system. A single pass through the grating pair would lead to spatial chirp. I.e., the frequencies of light in the laser would be spread out in time as well as space. The second pass through the system doubles the total temporal separation between frequency components while reversing the spatial separation. In this design the chirped pulse duration is directly related to the inter-grating separation, given by the following equation (Knutsen et al., 2004):

$$\Delta = 2 \left( \frac{d(\lambda/g)\tau}{cg(1 - (\lambda/g - \sin(\theta))^2)} \right) \quad (2.1)$$

where  $\Delta$  is the group delay dispersion,  $d$  is the inter-grating distance (measured perpendicular

to both gratings,  $g$  is the groove density of the grating,  $c$  is the speed of light,  $\lambda$  is the center wavelength,  $\tau$  is the spectral bandwidth of the pulse and  $\theta$  is the angle of incidence for the beam on the grating. The factor of two is present as the beam makes a double pass through the grating system. This design was ultimately chosen for the optional pulse stretcher on the Stokes line. One of the gratings is mounted in a stationary mount, while the other is mounted on a micrometer controlled translation stage for fine tuning of the inter-grating separation.

Figure 2.2b depicts a Martinez stretcher (Martinez, 1987), an alternative to the Treacy stretcher. The two lenses in the system have equivalent focal lengths,  $f$ , and are separated by a distance of  $2f$ . I.e. they form a 1:1 telescope. The separation between the lenses and the respective gratings then determines the total dispersion. A double-pass is again used to account for spatial chirp. The Martinez stretcher is capable of adding both positive and negative dispersion to the system depending on whether the distance between the gratings and lenses is less than or greater than  $f$ . The ability to add positive dispersion is important for applications such as chirped pulse amplification, where a Treacy style shaper broadens the pulses prior to amplification, and a Martinez style shaper recompresses them afterwards (Weiner, 1995). For spectral focusing applications, recompression is not a consideration, and so the added complexity of the Martinez stretcher is not needed.

Originally, the pump beam was also stretched using the pulse shaper depicted in figure 2.2a; however, issues of laser-induced damage quickly arose. The original gratings employed for the stretcher were aluminum coated. Aluminum has an interband transition at 800 nm, which is close to the operation wavelength of the pump for most spectral focusing applications. It was initially thought that absorption was occurring due to this transition (Fisher et al., 2001). The end result was that laser-induced indentations would appear in the grating. As the pump beam propagated through the rest of the system interference from components of the beam which had been reflected from different portions of these concave, circular, apertures would lead to the formation of an anular beam profile (as seen in Vyas, Kozawa, and Sato, 2014), in addition to an overall drop in transmission. At the sample-plane, an anular beam focuses to a line in  $z$ , where  $z$  is the direction of propagation. This leads to a drop in peak-power density as well as a loss of  $z$ -resolution. While anular beams can be used for tomography applications, they are detrimental to standard SRS microscopy, and so the aluminum grating system was removed.

Gold-coated reflective gratings were then purchased. Unlike aluminum, gold does not have an interband transition at 800 nm, and, in fact, is the preferred metal for operation in the near infrared. Nonetheless, the laser-induced damage continued to appear, with the same ultimate effects on the beam profile. As gold-coated gratings are often used with amplified laser systems at wavelengths near 800 nm, the damage is hypothesized to arise from a combination of multiphoton processes and the repetition rate of the laser. Amplified laser systems typically operate at kHz repetition rates, as opposed to the 80 MHz of the Insight DS+. This repetition rate, in combination with multiphoton processes that are efficient at 800 nm, but not 1040 nm, could lead to a heat-buildup that ultimately causes damage. At lower duty cycles, the heat can dissipate before detrimental effects appear.

As a result, the reflective gratings were replaced with a transmission grating, which does not suffer from the same damage mechanisms. The first design employed was that depicted in figure 2.2c. This type of design constitutes a folded Martinez-style pulse shaper. A mirror is placed at the focal length of a lens, and then the distance between the grating and lens is varied to change the dispersion. of all pulse shaper designs, this is the most compact; however, the drawback is alignment difficulty. Angular misalignment of the lens is amplified four-fold due to the multiple passes through the system. In addition, as the beam propagates through the lens far from its optical

axis, Seidel aberrations are large.

Overall, the aberrations and alignment sensitivity were found to be too great, and the system in figure 2.2d was ultimately adopted. This constitutes a Treacy-style pulse stretcher, employing a transmission grating, instead of reflective gratings. The retroreflectors are used to offset the beam during each pass through the grating to allow for easier separation of the chirped output pulse from the input pulse. The dispersion of this system is governed by the same equation 2.1 as the pulse shaper in figure 2.2a.

One final note is that adopting the pulse stretcher design depicted in figure 2.2e is also a possibility. As high efficiency diffraction gratings are very expensive, the total transmission through the multi-pass grating-based stretchers presented is fairly low using the gratings employed (cf. parts list). For most applications, there is ample power provided by the Insight DS+ that this does not necessarily affect experiments; however, to increase the total power through the pulse stretcher it is possible to reduce the number of passes through the grating. The total dispersion can remain constant by doubling the distance between the grating and the back retroreflector. The disadvantage of this setup is that it does not remove the spatial chirp acquired; however, the total spatial chirp that results from stretching the pulses to the desired lengths is fairly small. We have found that it does not greatly affect beam quality at the focus, and the slightly elliptical beam is still close enough to circular that all spectral components of the laser can be passed through the back aperture of the objective.

### 2.2.2.1 Characterizing pulses with modified FROG

To characterize the performance of the grating-based pulse stretchers it is necessary to measure the temporal (and spectral) profiles of the unstretched and stretched laser pulses. In order to measure any signal in time the detection system needs to be at least as fast as the temporal variation in the signal, and ideally it is much faster. Laser pulses with durations on the order of 100s of femtoseconds to a few picoseconds are too short to be measured electronically. Even the fastest electronics ultimately have a bandwidth limiting the temporal resolution to 10s-100s of picoseconds. As an alternative, however, the laser pulse itself, or another laser pulse of similar duration, can be used to probe the temporal profile.

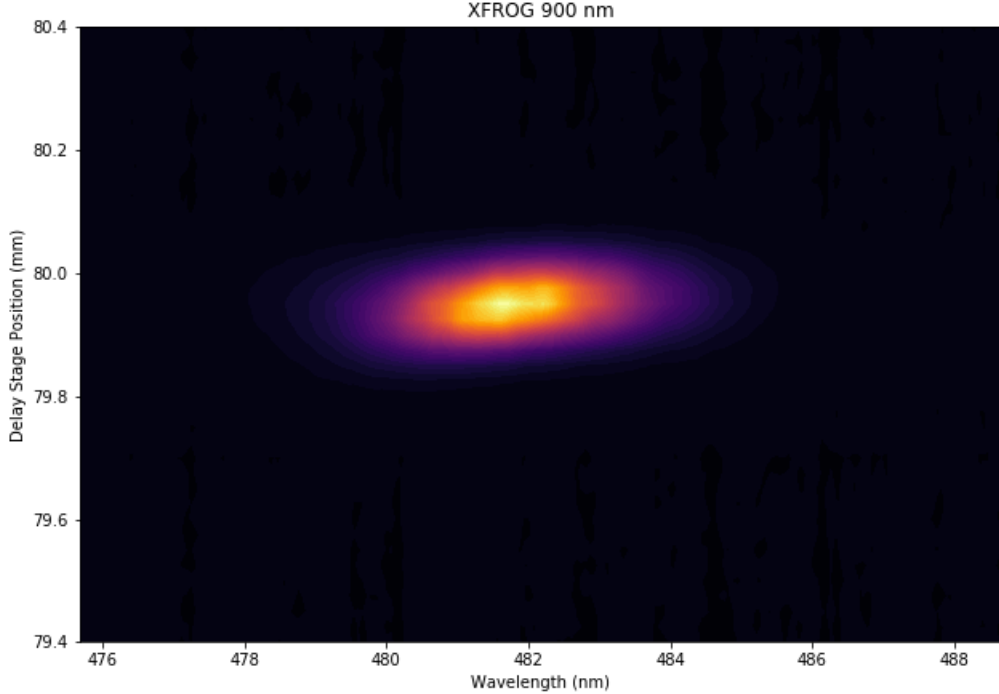
In order to make the requisite measurement, a laser pulse can be split into two halves, and then a signal can be generated and measured that is a function of the overlap between the two pulse duplicates. As the pulse duplicate (or other pulse) has a duration on the order of the duration of the pulse to be measured, the requisite temporal resolution can be achieved, as long as the signal being generated is a function of the temporal overlap between the two. As it turns out, this is exactly the process used in pump-probe nonlinear microscopy. A nonlinear optical signal is measured at the sample as a function of the delay between two optical pulses. If the two pulses are identical duplicates, and the signal is only generated when the two pulses are well overlapped in time, the resultant trace is an autocorrelation of the temporal profile of the pulse. Retrieving the pulse profile from an autocorrelation trace is an inverse problem, and in particular a 1D phase retrieval problem. Unfortunately, the 1D phase retrieval problem does not have a unique solution, meaning that ambiguities can arise (Beinert, and Plonka, 2015). In addition, it is well-known that the use of autocorrelation only measures the coherent part of an optical pulse (Trebino, 2000; Chung and Weiner, 2001). For the well-behaved pulses coming from a high-repetition rate oscillator, this latter comment is less of an issue, as, typically, strange pulse profiles only arise during an optical

amplification process. As a result, taking the coherent part of the pulse as a measure of the total pulse duration is usually sufficiently accurate.

Nonetheless, frequency-resolved optical gating (FROG) (Kane and Trebino, 1991) exists as an alternative to standard autocorrelation, and only requires the addition of a spectrometer to the setup. Instead of measuring a total intensity as a function of the interpulse delay, the spectrum is measured at each delay step. This results in a two-dimensional spectrogram of spectrum vs interpulse delay. From this spectrogram, the pulse amplitude and phase can be retrieved, analogously to with autocorrelation. However, the two-dimensional phase-retrieval problem of FROG has far fewer ambiguities compared to the 1D equivalent autocorrelation problem.

For generation of a FROG signal any nonlinear interaction can be used; however, the mathematics are simpler when using instantaneous processes, such as second harmonic generation, as the material response does not need to be considered, as it would need to be for using a process like SRS (DeLong et al., 1995). To perform a second harmonic FROG (SHG-FROG) measurement, a second harmonic crystal would be placed at the sample plane of the microscope, with a spectrometer placed after the condenser. The interpulse delay between two pulse duplicates would be varied, and the spectrum produced from the crystal would be measured for each delay. It should be noted that the mathematics to describe the FROG process generally makes use of the paraxial approximation, which is no longer valid when using high numerical aperture microscope objectives. Nonetheless, the utility of FROG, and its retrieval algorithms, have been shown to hold in the tight-focusing limit applicable to microscopy (Fittinghoff et al., 1998).

In the case of the SRS microscope, however, there are at least two pulses that need to be measured, the pump and the Stokes. If one of the pulses has already been characterized, by, e.g., an SHG-FROG measurement, it can be used to characterize the other in a process called cross-correlation FROG (XFROG) (Trebino, 2000). The sum-frequency generation nonlinear interaction is often used for this purpose. However, if both pulses are unknown, this defaults to a blind FROG algorithm, the solution to which is known to contain many possible ambiguities (Seifert, Stolz, and Tasche, 2004). This can be intuitively understood by a simple geometric argument. Consider the spectrogram produced from a sum-frequency XFROG measurement of two linearly chirped pulses. The spectrogram is an angled two-dimensional Gaussian (figure 2.3), which can be described by three degrees of freedom, the standard deviation in the major and minor axes, and the angular tilt of the signal. However, four degrees of freedom are needed to describe the two unknown pulses: the phase and amplitude of both pulses.



**Figure 2.3:** Example SFG-XFROG trace

A sum frequency generation XFROG trace between 900 nm pump beam and 1040 nm Stokes beam.

It has previously been demonstrated that by performing a double-blind FROG experiment, i.e., measuring two different FROG traces for the same pulse pair simultaneously, it is possible to solve the phase retrieval problem unambiguously (Wong et al., 2012). The two spectrograms recorded can be, for instance, the sum- and difference-frequency generation signals for the pulse pair.

While this offers an appealing solution to the pulse characterization problem faced in SRS microscopy and spectral focusing, it is difficult to realize experimentally. For instance, while the sum-frequency signal generated from the pump and Stokes beams is in the visible range, the corresponding difference frequency signal is in the mid infrared, which would require an entirely different set of detection optics. Other nonlinear interactions, such as the Kerr effect to perform polarization gating (PG-) FROG, typically require pulse energies that are far greater than those provided by the Insight DS+ (Trebino et al., 1997).

Thus, in order to be able to characterize the pulses without modification of the microscope optical setup purely for that measurement, we propose the following modification to the double-blind FROG algorithm. Two sum-frequency FROG traces are measured between the Stokes beam and the pump beam tuned to two different wavelengths. The Stokes beam is common between the two measurements, so a total of three electric fields need to be retrieved.

To perform the modified double blind FROG measurements, potassium dideuterium phosphate (KDP) salt crystals are placed at the sample plane, and a fiber coupled spectrometer is placed after the condenser lens. The Stokes polarization is rotated perpendicular to the pump polarization to

match the type II phase matching in which FROG has previously been demonstrated to work with microscope objectives (Fittinghoff et al., 1998). Note, however, that in the tight-focusing limit, linearly polarized light acquires polarization in all axes. Furthermore, as the interaction length is so short, the phase-matching bandwidth is nearly infinite. As such, type I phase matching is anticipated to work as well for these measurements as type II. To retrieve the pulses a modified version of the principal component generalized projections algorithm (Kane, 2008) is used, the mathematics of which is described below.

A single measured SFG-XFROG trace is given by:

$$I_{FROG}^{SFG}(\omega, \tau) = \left| \int_{-\infty}^{\infty} E_p(t) E_s(t - \tau) \exp(-i\omega t) dt \right|^2$$

where  $\omega$  is frequency (wavelength) and  $\tau$  is the delay between pulses.  $E_p(t)$  and  $E_s(t)$  are the time domain electric fields describing the pump and Stokes pulses.

We start with initial guesses for  $E_p(t)$  and  $E_s(t)$ . These are chosen to be Gaussian pulses with random phase. Next, consider  $E_p(t)$  and  $E_s(t)$  as vectors of length  $N$ ,  $N$  being the number of delays measured.

$$\begin{aligned} E_p &= [E_p(t_1), E_p(t_2), \dots, E_p(t_N)] \\ E_s &= [E_s(t_1), E_s(t_2), \dots, E_s(t_N)] \end{aligned}$$

Take the outer product  $O$ :

$$O = E_p E_s^T$$

The outer product  $O$  is then row-shifted and the columns are rearranged to obtain a matrix with columns at constant delay and rows at constant time. This is the time domain signal  $E_{sig}(t, \tau)$ . This is done by:

$$\begin{aligned} O_i &\rightarrow O_{i,-i:N-i} \\ E_{sig}(t, \tau)_{:,N/2} &= JO_{:,N/2} \\ E_{sig}(t, \tau)_{:,N/2} &= JO_{:,N/2} \end{aligned}$$

with  $J$  being the exchange operator, reversing the order of the elements in each row.

The columns of the matrix  $E_{sig}(t, \tau)$ , are Fourier transformed to yield  $E_{sig}(\omega, \tau)$ , at which point the square root of the experimental trace is used to replace the magnitude.

$$E'_{sig}(\omega, \tau) = \frac{E_{sig}(\omega, \tau)}{|E_{sig}(\omega, \tau)|} \sqrt{I_{FROG}^{SFG}}$$

$E'_{sig}(\omega, \tau)$  is inverse Fourier transformed back to the time domain yielding  $E'_{sig}(t, \tau)$ . The column and row manipulation is reversed to yield a new outer product  $O'$ . This is only a true outer product of rank 1 if the phase and amplitude are correct. In order to update the guesses for  $E_p$  and  $E_s$  until this is the case we keep only the largest principal component. We take the SVD:

$$O' = USV^T$$

On even iterations of the algorithm:

$$E_p \rightarrow U_{:,1}$$

On odd iterations:

$$E_s \rightarrow V_1^T$$

In this manner each field is optimized iteratively holding the other field constant. Simultaneously, this process is done using the other FROG trace acquired from the pump tuned to a different wavelength; however, the iterations in which the pump and Stokes field guesses are updated are reversed. That is, when the guess for the Stokes field is taken to be the true Stokes field to update one pump field, the other pump field is then taken to be accurate in order to update the Stokes field for that iteration. All three fields are updated until convergence. Two errors are monitored:

$$G = \sqrt{\frac{1}{N^2} \sum_1^N (I_{FROG}^{SFG} - |E_{sig}(\omega, \tau)|^2)^2}$$

$$Z = \sum_1^N |E'_{sig}(t, \tau) - E_{sig}(t, \tau)|^2$$

Convergence is typically reached when the above errors are on the order of  $10^{-3}$ .

Using this procedure the pump and Stokes fields were measured for the pump tuned to 800 nm and 900 nm. The Stokes field has a fixed output of 1040 nm. The pump output at 800 nm is one of the wavelengths typically used for spectral focusing as it allows for coverage of the C-H stretching window. While a rigorous mathematical proof is still outstanding, we believe there are sufficiently many degrees of freedom contained within the spectrograms to fully describe all three fields. Additionally, the pulse duration of the pump beam has independently been verified using the dispersion scan technique (Wnuk et al., 2016). For the unchirped pulses, i.e., without the use of the grating stretchers, the measured pulse widths and spectral bandwidths are shown in table 2.1.

| Wavelength       | Duration (FWHM) | Bandwidth (FWHM) |
|------------------|-----------------|------------------|
| 1040 nm (Stokes) | 317 fs          | 5.7 nm           |
| 900 nm (Pump)    | 179 fs          | 6.7 nm           |
| 800 nm (Pump)    | 181 fs          | 5.5 nm           |

**Table 2.1:** Unstretched pulse parameters

The pulse durations and spectral bandwidths for the fixed Stokes line and two OPO pump outputs without the use of the pulse stretchers. The values were determined by the alternating blind-FROG procedure outlined in the text.

For these same wavelengths, the pulse durations are shown for the current pulse stretcher configurations in table 2.2. Given these durations, the theoretical spectral resolution of a spectral focusing measurement is also given. The spectral bandwidths are omitted as they are similar to those retrieved for the unchirped pulses. Note that the grating stretcher needs to be realigned on the pump line to chirp the beam at 900 nm. This wavelength is not typically used in our spectral focusing experiments.

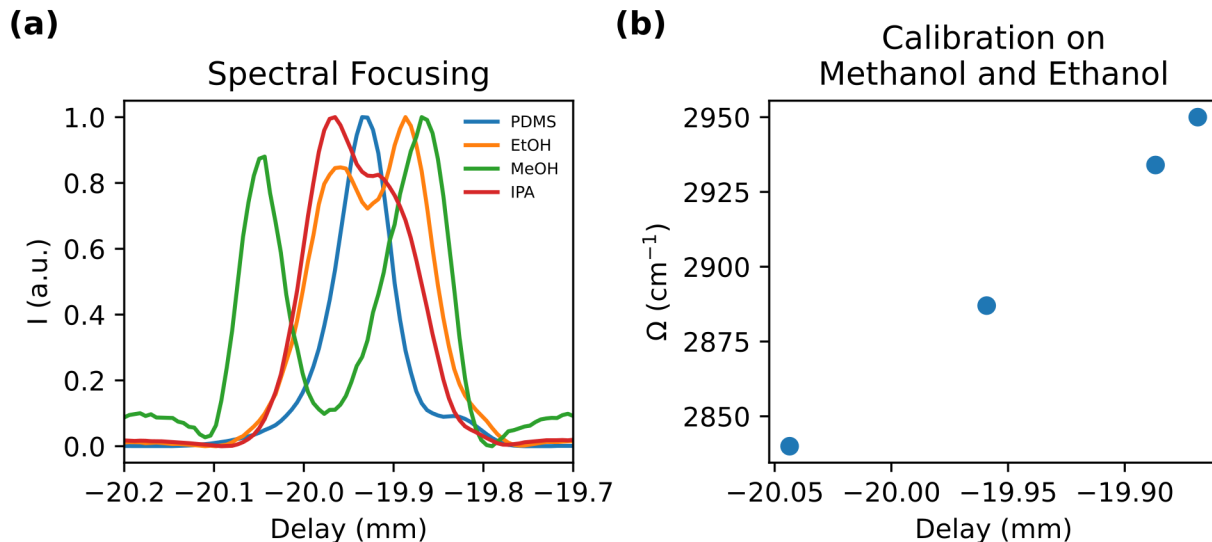
| Wavelength       | Duration (FWHM) | Theoretical Spectral Resolution |
|------------------|-----------------|---------------------------------|
| 1040 nm (Stokes) | 1.7 ps          | 8.6 cm <sup>-1</sup>            |
| 900 nm (Pump)    | 1.5 ps          | 9.8 cm <sup>-1</sup>            |
| 800 nm (Pump)    | 1.9 ps          | 7.7 cm <sup>-1</sup>            |

**Table 2.2:** Stretched pulse parameters

The pulse durations and spectral bandwidths for the fixed Stokes line and two OPO pump outputs with the use of the pulse stretchers. The values were determined by the alternating blind-FROG procedure outlined in the text.

We have thus far been careful to emphasize the linearity of the chirp. This is because all possible dispersion mechanisms for lengthening a pulse impart not only second order dispersion - responsible for linear chirping - but third order dispersion (TOD), and higher as well. For comparatively moderate stretching factors, provided by small lengths of glass, or moderate distances between a grating pair or prisms, TOD is low and the acquired chirp is quasi-linear (Mohseni, Polzer, and Hellerer, 2018). Nonetheless, there is still a limit imposed on the achievable spectral resolution through spectral focusing as larger stretching factors impart greater amounts of TOD (and higher) ultimately degrading resolution. Notably, the range of possible stretching factors can be increased by careful consideration of the materials used. This is possible because while both glass and prisms impart positive TOD, the acquired TOD from a grating system is negative (Mohseni, Polzer, and Hellerer, 2018). This means that it is, in principle, possible to design a system of gratings and/or prisms/glass rods where the TOD cancels to zero, allowing longer linear chirps to be obtained (e.g., Kane and Squier, 1995). Indeed, "grisms" (grating prisms) are often employed where higher order dispersion needs to be carefully controlled, such as in chirped pulse amplification systems (Kane and Squier, 1997). These components are highly specialized, and as a result expensive, and not necessary to achieve spectral resolutions that are reasonable for SRS microscopy. It is still worth remembering, however, that resolution gains are not unlimited.

Using these stretched pulses it is possible to acquire high resolution spectra as depicted in figure 2.4a. A delay stage position, corresponding to the interpulse delay, can then be mapped to a wavenumber to produce a calibration curve as in 2.4b.





**Figure 2.4:** Spectra acquisition using, and calibration of, stretched pulses

(a) Spectra acquired using spectral focusing of PDMS and 3 alcohols: methanol, ethanol and isopropanol. The spectra are all normalized to their maxima to highlight differences in structure. They are plotted as a function of the delay stage position which determines the pulse overlap between the pump and Stokes beams. The pump beam was tuned to 799 nm, while the Stokes is fixed at 1040 nm. (b) The wavenumber calibration of the delay stage position based on the known Raman shifts of the ethanol and methanol peaks. The calibration is highly linear indicating comparatively little higher order dispersion from the grating stretcher system.

### 2.2.3 Detection characteristics and noise model

Finally, we turn our attention to some details of lock-in detection. A lock-in detector performs phase-sensitive detection by multiplying an input signal of interest by a known reference signal, and then passing the result through a low-pass filter. The multiplication of the input signal with the reference waveform should produce a DC output proportional to the strength of the desired component, and so the low-pass filter can remove contributions at frequencies other than the desired one. This procedure, in principle, allows for the detection of very small signals even in the presence of high noise, or other signals, provided that it is employed at a frequency sufficiently far away from the frequency of any of these other contributions. For SRS microscopy, this is accomplished by modulating the Stokes beam at 10.28 MHz, far from the frequencies that dominate the  $1/f$  noise produced by the laser. The input to the lock-in amplifier is also bandpass filtered, with a 5 MHz bandwidth, around this modulation frequency to avoid saturation of the electronics due to the large DC component of the average laser power, and the 80 MHz component from the laser repetition rate.

Note that in the setup described above, the Stokes beam is modulated using an additional function generator. Alternatively, the internal oscillator of the lock-in amplifier could be used to generate the driving waveform. This has the benefit of the lock-in amplifier having an exact copy of the driving waveform to use as a reference signal for the demodulation process. In practice, however, the output connection from the oscillator is adjacent to the input connection for the signal coming from the photodiode on the lock-in amplifier face. We found that this leads to a lot of cross-talk between the output channel and the input channel, which is especially problematic for SRL detection. This is because the SRL signal is  $180^\circ$  out of phase with the driving modulation waveform. When the latter is at its maximum, the Stokes beam output power is zero and so there is no SRL, and vice versa. For this reason an additional function generator was used, which was placed physically far away from the lock-in amplifier to minimize cross-talk. The downside of this setup, however, is that the lock-in amplifier no longer has knowledge of the driving waveform to use as a reference for demodulation. In principle the output of the function generator could be split, with one line providing a reference signal to the lock-in amplifier; however, this results, again, in a lot of cross-talk as the two inputs are in close proximity. The particular lock-in amplifier used in this microscope is capable of determining the phase of the reference waveform to use for demodulation directly from the input signal (given the desired frequency as input); however, this process adds extra noise to the measurement. We found that simply synchronizing the 10 MHz master clocks in the function generator and lock-in amplifier was sufficient to improve the lock-in

amplifiers ability to determine the correct phase to use. The input for clock synchronization is on the back panel of the lock-in amplifier which is physically far enough away that cross-talk is not significant. We found that this combination of clock synchronization with an external function generator provided the best balance of signal to noise.

In addition to generation of the driving modulation waveform, there are multiple lock-in detection parameters to consider. To begin with, for instance, there is the lock-in time constant which is one of the determining factors of the low-pass filters employed by the amplifier. The minimum time constant for a given pixel dwell time (scanning speed) is dictated by the Nyquist frequency. Specifically, the time-constant must be at least a factor of two faster than the dwell time in order to accurately reconstruct the image. However, beyond this minimum, the time constant affects both the noise and settling time of the system, independently of the scanning speed. This means that for a given noise tolerance, and desired signal accuracy, a number of multiples of a time constant must be waited for. This effectively dictates the minimum pixel dwell time required for the most accurate measurements. The trade-off between these two minima must be considered when deciding on the final acquisition parameters.

To better understand the performance of the SRS microscope it will be useful to introduce a simple noise model. Due to the chosen modulation frequency,  $1/f$  is negligible and can be ignored. However, there is still shot noise to consider. Shot, or quantum, noise is Poisson distributed and arises from the uncertainty in the stimulated emission of a photon during lasing. As a result, there is also some uncertainty, i.e., noise, in the photocurrent produced at a photodiode detecting the light from the laser. The shot noise is given by equation 2.2 below

$$\sigma_s = \sqrt{2eI} = \sqrt{2e\mathcal{R}P} \quad (2.2)$$

where  $\sigma_s$  is the shot noise,  $I$  is the induced photo current, and  $e$  is the elementary charge. The photocurrent can be calculated from the product of the photodiode responsivity, denoted  $\mathcal{R}$ , and the incident optical power,  $P$ . The responsivity at 800 nm for the photodiode used in this microscope (Hamamatsu S3590-08), is approximately 0.575 A/W. Note that the shot noise is independent of temperature and is only a function of the incident optical power. I.e., higher optical powers produce greater fluctuations in photocurrent. Nonetheless, the signal-to-noise ratio (SNR) still improves in proportion to  $\sqrt{P}$  as the SRS signal is bilinear in the optical power of both pump and Stokes beams. By varying the power of the undetected beam (the Stokes in our case), it is also possible to increase the signal level moderately, with less effect on the detected noise.

In addition to shot noise, there are other noise sources to consider, such as the lock-in amplifier input noise and thermal noise. The former is approximately  $5nV/\sqrt{Hz}$ , as specified by the lock-in amplifier manufacturer. Thermal noise is a function of temperature, and is independent of the optical power. We will consider a highly simplified model where we consider the thermal (Johnson-Nyquist or J-N) noise of just a single 50 Ohm resistor, such as the one used as the load resistor in the photodiode detection circuit. Note that J-N noise is also independent of the material used to construct the resistor, and is simply a function of the resistance. It is given by equation 2.3 below.

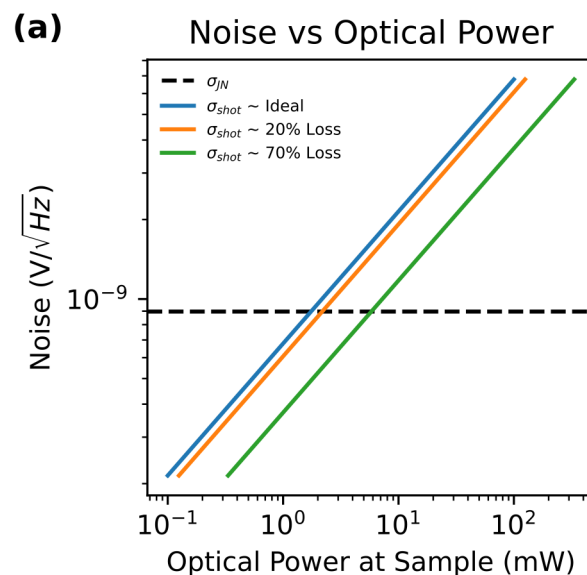
$$\sigma_{JN} = \sqrt{4K_BTR} \quad (2.3)$$

where  $K_B$  is the Boltzmann constant,  $T$  is the temperature, and  $R$  is the resistance. For our single 50 Ohm resistor model, the thermal noise is approximately  $0.9nV/\sqrt{Hz}$ .

Notably, SRS microscopy, and pump-probe microscopy in general, is often assumed to be operating at the shot-noise limit. This is considered a good thing, as this is the most fundamental noise floor a measurement can reach using non-quantum light. Note, however, that SRS microscopy employing squeezed light has been demonstrated which can further improve on this assumption (Casacio et al., 2021).

Nonetheless, if we examine the actual noise characteristics of even our simple model, we see that this situation is not necessarily the case. In figure 2.5a, the shot noise and thermal noise for the introduced model is plotted as a function of optical power **at the sample**. Note, that the optical power at the sample is only equivalent to the optical power at the detector, and thus a reliable measurement to use for quantifying noise, if there is 100% transmission from the sample to the detector (blue curve). In this ideal scenario, shot noise becomes dominating at approximately 0.5 mW of incident pump power. However, if we account for transmission losses along the imaging path (orange and green curves), this is no longer the case. For 80% transmission, which is the estimated transmission of the SRS microscope described, 1 mW excitation power is needed to reach the shot noise limit. For 30% transmission, 5 or 6 mW is needed. This latter transmission level, or even lower, is frequently encountered in applications employing epi-detected SRS (Slipchenko et al., 2012). Furthermore, as this simple model excludes multiple sources of noise, including even the thermal noise from additional resistors (e.g. in the bandpass filter), it becomes clear that operation at the shot noise limit is not a guarantee for excitation powers commonly employed in SRS microscope (5-15 mW), while avoiding the worst effects of phototoxicity in biological samples.

This observation has implications for the manner in which we detect the SRS signal. In particular, we measure the signal from a single point in space at a single wavenumber at a time. In other types of spectroscopy, multiplexing, i.e., measuring signal from multiple points in space or multiple spectral points simultaneously, can be shown to have a SNR advantage, if the signal detection is not shot-noise limited (Fellgett, 1949). At least in the low-power regime, which is necessary for the most photosensitive biological samples, this is a possibility for the SRS microscope described in this chapter. In chapter 5 we will introduce a new type of imaging modality which can leverage this observation while increasing image acquisition speed.



**Figure 2.5:** Thermal and shot noise for SRS detection vs optical power

(a) Thermal noise is plotted as the dashed horizontal line as it is independent of optical power. The noise is calculated using the basic model introduced in the text of a single resistor at 19 °C. Shot noise is calculated as described in the text. Note that the noise is plotted as a function of the optical power at the sample, not the detector. If there are no losses between the sample and the detector, the noise for the model is given by the blue curve. The orange and green curves represent the noise curves given 20% and 70% power losses between the sample and the detector, respectively. Both the x-axis and y-axis are plotted on a log scale.

Placing consideration of the shot-noise limit aside for the moment, the overall detection noise of the system is still determined by a number of lock-in amplifier parameters. In particular the frequency bandwidth, and thus noise, let through a low-pass filter is a function of the time constant and the filter order. If we consider the low pass filter as cascaded RC filters, the order is the number of those filters. The bandwidth let through by a low-pass filter of time constant  $\tau$  and order  $k$  is given by equation 2.4 below.

$$f_{-3db} = \frac{1}{\pi\tau} \sqrt{2^{1/k} - 1} \quad (2.4)$$

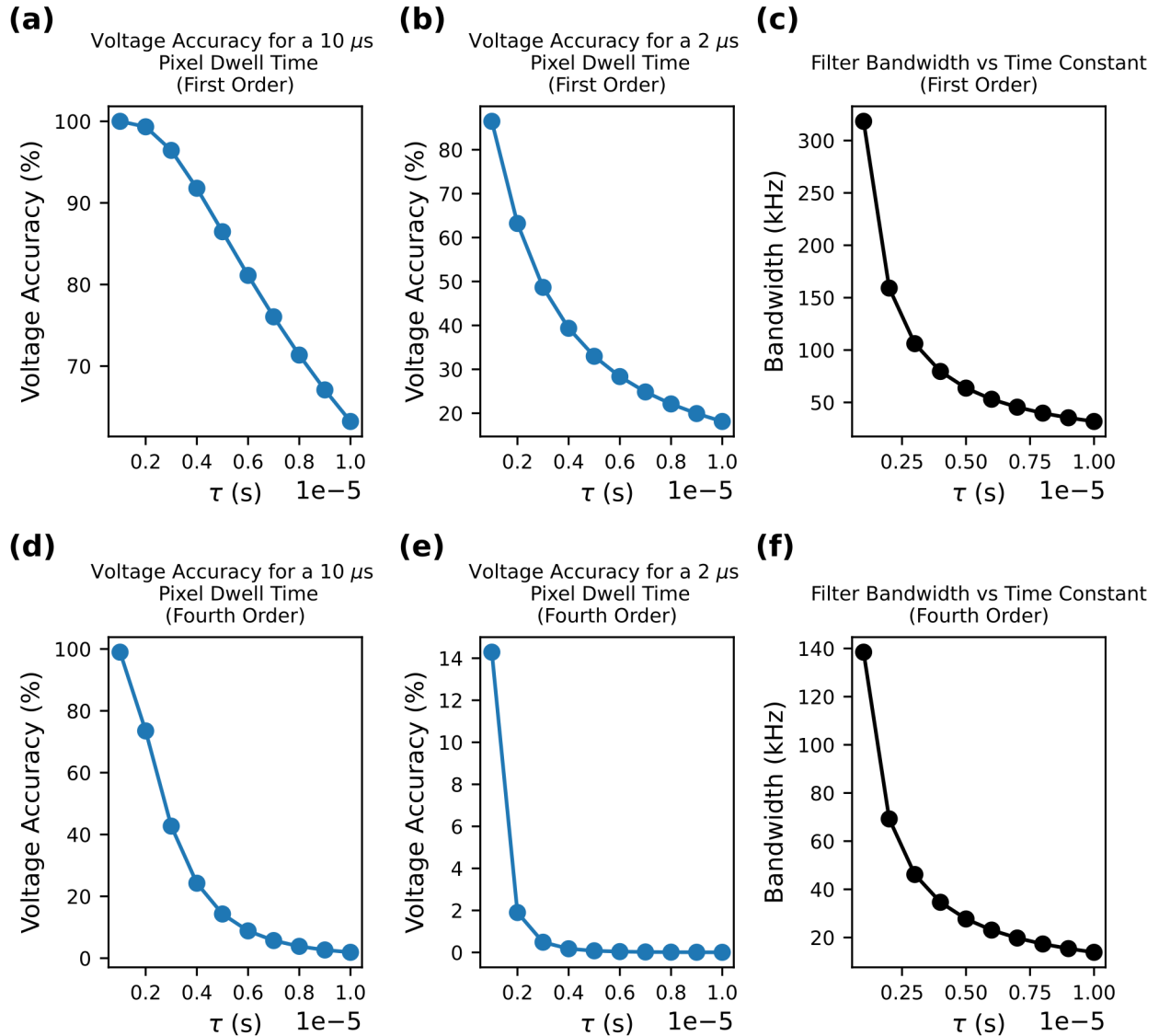
As shown in figure 2.6c and f, for all lock-in amplifier time constants, increasing the filter order from 1 to 4 reduces the total bandwidth of the filter, and thus the amount of additional noise in the measurement. However, this decrease in noise comes at a cost. It takes a longer time to reach a settled voltage for higher order filters. Physically, this is a result of the time it takes to charge the capacitors of the cascaded RC circuit. The end result is then that, for a time constant of  $\tau$  and filter order of  $k$ , the time to reach 99% of the final voltage value is given by equation 2.5.

$$e^{-t/\tau} \sum_{k=0}^{n-1} \frac{(t/\tau)^k}{k!} = 0.01 \quad (2.5)$$

Alternatively, this time can be converted to a voltage accuracy metric,  $g$ , i.e., a percentage of the final voltage achieved by waiting for a given time  $t$ .

$$g = 1 - e^{-t/\tau} \sum_{j=0}^{n-1} \frac{(t/\tau)^j}{j!} \quad (2.6)$$

Using this equation it is possible to calculate the voltage accuracy of measurements made using a certain lock-in time constant and filter order for a given pixel dwell time. In figure 2.6a and d the voltage accuracy is plotted as a function of time constant for a 10  $\mu$ s pixel dwell time using a first order and fourth order filter, respectively. In figure 2.6b and e, the analogous plots are presented for a 2  $\mu$ s pixel dwell time. Perhaps unsurprisingly, in the plots of figure 2.6b and e it is clear that using a lock-in time constant that is longer than the pixel dwell time leads to inaccurate results. Note, however, that this is not accounting for the change in signal as the beam has moved to a new location. I.e., this is a depiction of the anticipated accuracy of the signal measured at a single point.



**Figure 2.6:** Detection noise and voltage accuracy for a variety of lock-in amplifier parameter choices

(a-b) The voltage accuracy of a first order filter is plotted vs the lock-in time constant assuming a 10  $\mu$ s and 2  $\mu$ s pixel dwell time, respectively. (c) The noise bandwidth of a first order filter is plotted vs the lock-in time constant. (d-e) As with (a-b) the voltage accuracy is plotted assuming a 10  $\mu$ s and 2  $\mu$ s pixel dwell time, respectively; however, these plots correspond to a fourth order low-pass filter. (f) The noise bandwidth as a function of lock-in time constant is plotted for a fourth order low-pass filter.

More surprising, however, is that, particularly for fourth order filters, which are the default operating mode of the lock-in amplifier used in the described microscope, pixel dwell times that are 5 or more times longer than the lock-in time constant are necessary to accurately measure the signal. For shorter pixel dwell times, the low pass filter is simply not able to respond fast enough to the rapid changes in signal. The implication of this observation is that there are inaccuracies

in most SRS microscopy measurements being made as the lock-in time constant is often set to a value that fulfills the Nyquist sampling criterion, but ignores the settling time of the filter used.

There is of course a tradeoff between accuracy (and SNR), and acquisition time. Biological microscopy is often in the difficult position where it needs both high SNR measurements and fast acquisition speeds. This is technologically challenging to implement. As previously noted above in the discussion on shot noise, in chapter 5 a new type of imaging modality will be introduced which can potentially increase acquisition speeds by 5 or 10 fold. An alternative use of this imaging modality may be to maintain current acquisition speeds in favor of using longer pixel dwell times to increase the accuracy of the SRS measurements being made.

## 2.3 Software

Operation of the SRS microscope requires the coordination of multiple components. These include the scanning mirrors, sample stage, the optical delay stage, the laser, and the lock-in amplifier, among others not covered in depth. The most rapid and seamless operation of a hypothetical microscope could be achieved through the all electronic coordination of these components, through the use of, for instance, a FPGA(s) and/or TTL triggering. The limits of what can be achieved using such a setup are well exemplified by a recent publication using SRS microscopy for rapid cell sorting (Nao et al., 2020). Nonetheless, that setup required the development of an extensive amount of custom electronics, and laser systems. The alternative system we have developed, and described in this chapter, makes use of commercially available products which can simplify some aspects of this coordination.

Even so, a fair amount of custom software is required to efficiently coordinate all the components. Not all of the electronics were designed to accept TTL triggering for performing actions, as such coordination is accomplished via a central computer running software which communicates with all the components, for the most part via serial communication. Software was developed in Python, C++ and Cython (for interfacing the two languages), in order to facilitate this communication. An additional graphical user interface (GUI) was included to allow full control of the laser, delay line, lock-in amplifier, and some additional motors. Unfortunately, the scanning microscope used is closed source, and does not have an application programming interface (API), so the proprietary Olympus software is still required in order to start the scanning process; however, image reconstruction and control of the other aspects of a SRS experiment can be monitored from the central GUI. The documentation for the software has been provided as an appendix to this dissertation. It is also available from the source repository. Note that the code is under constant development, and so the documentation is subject to change. For the most up-to-date version, refer to the repository.

## 2.4 References

Beinert, R., and Plonka, G. (2015) Ambiguities in one-dimensional discrete phase retrieval from Fourier magnitudes”, *Journal of Fourier Analysis and Applications*, Vol. 21 No. 6, pp. 1169-1198.

Casacio, C. A., Madsen, L. S., Terrasson, A., Waleed, M., Barnscheidt, K., Hage, B., Taylor,

M. A., Bowen, W. P. (2021) “Quantum-enhanced nonlinear microscopy”, *Nature*, Vol. 594, pp. 201-206.

Cheng, J-X., and Xie, X. S. eds (2016) “Coherent Raman scattering microscopy”, CRC press.

Cheng, J-X., Min, W., Ozeki, Y., and Polli, D. eds (2021) “Stimulated Raman scattering microscopy”, Elsevier.

Chung, J-H., and Weiner, A. (2001) “Ambiguity of ultrashort pulse shapes retrieved from the intensity autocorrelation and the power spectrum”, *IEEE J. Sel. Top. Quantum Electronics*, Vol. 7 No. 4, pp. 656-666.

DeLong, K. W., Ladera, C. L., Trebino, R., Kohler, B., and Wilson, K. R. (1995) “Ultrashort-pulse measurement using noninstantaneous nonlinearities: Raman effects in frequency-resolved optical gating”, *Optics Letters*, Vol. 20 No. 5, pp. 486-488.

Fellgett, P. B. (1949) “On the ultimate sensitivity and practical performance of radiation detectors”, *Journal of the Optical Society of America*, Vol. 39 No. 11, pp. 970-976.

Fisher, D., Fraenkel, M., Henis, Z., Moshe, E., and Eliezer, S. (2001) “Interband and intraband (Drude) contributions to femtosecond laser absorption in aluminum”, *Physical Review E*, Vol. 65 No. 1.

Fittinghoff, D. N., Squier, J. A., Barty, C. P. J., Sweetser, J. N., Trebino, R., and Muller, M. (1998) “Collinear type II second-harmonic generation frequency-resolved optical gating for use with high-numerical-aperture objectives”, *Optics Letters*, Vol. 23 No. 13, pp. 1046-1048.

Freudiger, C. W., Min, W., Saar, B. G., Lu, S., Holtom, G. R., Chengwei, H., Tsai, J. C., Kang, J. X., and Xie, X. S. (2008), “Label-free biomedical imaging with high sensitivity by stimulated Raman scattering microscopy”, Vol. 322 No. 5909, pp. 1857-1861.

Fu, D., Holtom, G., Freudiger, C., Zhang, X., and Xie, X. S. (2013) “Hyperspectral Imaging with Stimulated Raman Scattering by Chirped Femtosecond Lasers”, *The Journal of Physical Chemistry B*, Vol. 117, pp. 4634–4640.

Kane, D. J., and Trebino, R. (1991) “Measuring the intensity and phase of a femtosecond pulse using spectrally resolved self-diffraction”, OSA Annual Meeting, Optica Publishing Group.

Kane, D. J. (2008) “Principal components generalized projections: a review”, *Journal of the Optical Society of America B*, Vol. 25 No. 6, pp. A120-A132.

Kane, S., and Squier, J. (1995) “Grating compensation of third-order material dispersion in the normal dispersion regime: Sub-100-fs chirped-pulse amplification using a fiber stretcher and grating-pair compressor”, *IEEE Journal of Quantum Electronics*, Vol. 31 No. 11, pp. 2052-2057.

Kane, S., and Squier, J. (1997) “Grism-pair stretcher-compressor system for simultaneous second- and third-order dispersion compensation in chirped-pulse amplification”, *Journal of the Optical Society of America B*, Vol. 14 No. 3, pp. 661-665.

Kawanishi, T. (2022) “Electro-optic modulation for photonic networks”.

Knutsen, K. P., Johnson, J. C., Miller, A. E., Petersen, P. B., Saykally, R. J. (2004) “High spectral resolution multiplex CARS spectroscopy using chirped pulses”, *Chemical Physics Letters*, Vol. 387, pp. 436–441.

Lewis, I. R., and Edwards, H. (2001) “Handbook of Raman spectroscopy: from the research laboratory to the process line”, CRC press.

Manifold, B., and Fu, D. (2022) “Quantitative stimulated Raman scattering microscopy: promises and pitfalls”, *Annual Review of Analytical Chemistry*, Vol. 15.

Martinez, O. (1987) “3000 times grating compressor with positive group velocity dispersion: Application to fiber compensation in 1.3-1.6  $\mu\text{m}$  region”, *IEEE Journal of Quantum Electronics*, Vol. 23 No. 1, pp. 59-64.

McCamant, D. W., Kukura, P., and Mathies R. A. (2003) “Femtosecond broadband stimulated Raman: A new approach for high-performance vibrational spectroscopy” *Appl. Spectrosc.*, Vol. 57, pp. 1317-1323.

McCreery, R. L. (2005) “Raman spectroscopy for chemical analysis”, John Wiley & Sons.

Messerschmidt, R. G., and Harthcock, M. A. (1988) “Infrared microspectroscopy. Theory and applications”..

Mohseni, M., Polzer, C., and Hellerer, T. (2018) “Resolution of spectral focusing in coherent Raman imaging”, *Optics Express*, Vol. 26 No. 8.

Mukamel, S. (199) “Principles of nonlinear optical spectroscopy (No. 6)”, Oxford University Press on Demand.

Nao, N., Iino, T., Isozaki, A., Yamagishi, M., Kitahama, Y., Sakuma, S., Suzuki, Y., Tezuka, H., Oikawa, M., Arai, F., Asai, T., Deng, D., Fukuzawa, H., Hase, M., Hasunuma, T., Hayakawa, T., Hiraki, K., Kotaro, H., Hoshino, Y., Inaba, M., Inoue, Y., Ito, T., Kajikawa, M., Karakawa, H., Kasai, Y., Kato, Y., Kobayashi, H., Lei, C., Matsusaka, S., Mikami, H., Nakagawa, A., Numata, K., Ota, T., Sekiya, T., Shiba, K., Shirasaki, Y., Suzuki, N., Tanaka, S., Ueno, S., Watarai, H., Yamano, T., Yazawa, M., Yonamine, Y., Di Carlo, D., Hosokawa, Y., Uemura S., Sugimura, T., Ozeki, Y., and Goda, K. (2020), “Raman image-activated cell sorting”, *Nature Communications*, Vol. 11.

Seifert, B., Stolz, H., Tasche, M. (2004) “Nontrivial ambiguities for blind frequency-resolved op-



tical gating and the problem of uniqueness”, *Journal of the Optical Society of America B*, Vol. 21 No. 5, pp. 1089-1097.

Slipchenko, M. N., Oglesbee, R. A., Zhang, D., Wu, W., and Cheng, J. X. (2012) “Heterodyne detected nonlinear optical imaging in a lock-in free manner”, *Journal of biophotonics*, Vol. 5 No 10., pp. 801-807.

Stuart, B. H. (2004) “Infrared spectroscopy: fundamentals and applications”, John Wiley & Sons.

Trebino, R., DeLong, K. W., Fittinghoff, D. N., Sweetser, J. N., Krumbugel, M. A., Richman, B. A. (1997) “Measuring ultrashort laser pulses in the time-frequency domain using frequency-resolved optical gating”, *Rev. Sci. Instrum.*, Vol 68 No. 9.

Trebino, R. eds (2000) “Frequency-resolved optical gating: the measurement of ultrashort laser pulses”, Springer Science and Business Media.

Treacy, E. (1969) “Optical pulse compression with diffraction gratings” *IEEE Journal of Quantum Electronics*, Vol. 5 No. 9, pp. 454-458.

Turrell, G. and Corset, J. eds (1996) “Raman microscopy: developments and applications”, Academic Press.

Upputuri, P. K., Gong, L., and Wang, H. (2014) “Chirped time-resolved CARS microscopy with square-pulse excitation”, *Optics Express*, Vol. 22 No. 8, pp. 153–159.

Vyas, S., Kozawa, Y., Sato, S. (2014) “Generation of a vector doughnut beam from an internal mirror He-Ne laser”, *Optics Letters*, Vol. 39 No. 7, pp. 2080-2082.

Weiner, A. M. (1995) “Femtosecond optical pulse shaping and processing”, *Progress in Quantum Electronics*, Vol. 19 No. 3, pp. 161-237.

Wnuk, P., Fuest, H., Neuhaus, M., Loetscher, L., Zherebtsov, S., Riedle, E., Major, Zs., and Kling, M. F. (2016) “Discrete dispersion scanning as a simple method for broadband femtosecond pulse characterization”, *Optics Express*, Vol. 24 No. 16, pp. 18551-18558.

Wong, T. C., Ratner, J., Chauhan, V., Cohen, J., Vaughan, P. M., Xu, L., Consoli, A., Trebino, R. (2012) “Simultaneously measuring two ultrashort laser pulses on a single-shot using double-blind frequency-resolved optical gating”, *Journal of the Optical Society of America B*, Vol. 29 No. 6, pp. 1237-1244.

## 2.5 Appendix: Parts list

Below is a complete parts list for the free-space optics and optomechanics necessary to build the setup in figure 2.1. Part numbers, except where otherwise noted, are from Thorlabs and are

provided as a reference; however, many of these items can be obtained from multiple suppliers.

| <b>Description</b>                              | <b>Quantity</b> | <b>Part Number</b> |
|---|-----------------|--------------------|
| Flip mount adapter                              | 3               | FM-90              |
| 3", 1/2" diameter, posts                        | 18              | TR3                |
| 2", 1/2" diameter, posts                        | 3               | TR2                |
| 3" post holder                                  | 18              | PH3                |
| 2" post holder                                  | 3               | PH2                |
| 1.25" diameter pedestal                         | 21              | BE1                |
| Clamping fork                                   | 21              | CF175              |
| Breadboard for detection optics                 | 1               | MB1218             |
| 1.5" diameter post for breadboard (L = 14")     | 4               | P14                |
| 1.5" diameter post for breadboard (L = 4")      | 4               | P4                 |
| 1" diameter post for periscope (L = 12")        | 1               | RS12               |
| 1" diameter post for periscope (L = 2")         | 1               | RS2                |
| 1" post C-clamp                                 | 1               | RM1A               |
| Lens tube                                       | 1               | SM1L30             |
| Lens tube                                       | 1               | SM1CPL             |
| Lens tube                                       | 1               | SM1S30             |
| Right angle cage for elliptical optics          | 1               | KCB1E              |
| Rotation mount                                  | 3               | RSP1               |
| Motorized rotation mount and controller         | 1               | KPRM1E             |
| Manual rotation platform                        | 2               | QRP02              |
| Kinematic mirror mount                          | 20              | KM100              |
| Kinematic prism mount                           | 2               | KM100              |
| Polarizer mount                                 | 3               | SM1PM10            |
| 1/2" translation stage                          | 3               | MT1                |
| Top unit of periscope                           | 1               | RS99T              |
| 45° mount for 1" optics                         | 1               | H45                |
| Kinematic grating mount                         | 1               | KGM60              |
| Kinematic mount for 2" square optics            | 1               | KM200S             |
| 1" diameter protected silver mirrors            | 21              | PF10-03-P01        |
| 1" D-shaped protected silver mirror             | 1               | PFD10-03-P01       |
| 1"x1" Hollow roof prism protected silver mirror | 2               | HRS1015-P01        |
| 1000 nm shortpass dichroic mirror               | 1               | DMSP1000           |
| 1" protected silver elliptical mirror           | 1               | PFE10-P01          |
| 750 nm longpass filter                          | 1               | FELH0750           |
| 1000 nm shortpass filter                        | 1               | FESH1000           |

*continued on next page*

|  |   |                                      |
|--|---|--------------------------------------|
| Half wave Fresnel rhomb  | 1 | FR600HM                              |
| Glan laser polarizer   | 3 | GL10-B                               |
| Half waveplate (1053 nm)   | 1 | WPH10ME-1053                         |
| Quarter waveplate  | 1 | DMSP950                              |
| Near IR 1" achromatic lens kit   | 1 | LSB08-B                              |
| 300 grooves/mm, 31.7° groove angle,<br>transmission grating, 50 mm x 50 mm | 1 | GTI50-03A                            |
| Ruled reflective diffraction grating,<br>1 $\mu$ m blaze, 50 mm x 50 mm    | 2 | GR50-0310                            |
| 200 mm optical delay line  | 1 | DL200<br>*Newport, out of production |
| Resonant free-space<br>electro-optic modulator (10.28 MHz)                 | 1 | EO-AM-R-20-C1                        |
| Compact five-axis stage (EOM Alignment)                                    | 1 | PY005                                |
| Slotted mounting base for 5-axis stage                                     | 1 | PY005A1                              |
| Power amplifier  | 1 | ZHL-32A+<br>*Mini-circuits           |
| Large area Si Photodiode   | 1 | S35090-08<br>*Hamamatsu              |
| 50 MHz lock-in amplifier   | 1 | HF2LI<br>*Zurich Instruments         |

**Table 2.4:** Optics and optomechanics for the SRS microscope

# Chapter 3

## Image processing and information quantification

\*Some of the results in this chapter have been previously published and have been adapted for inclusion here.

### 3.1 Introduction

Optical microscopy in all its varieties has a long history. Since Robert Hooke first published *Micrographia*, in the 17th century (Hooke 1635), scientists have been pushing the limits of the technique. This has led to a proliferation of different imaging modalities all with different strengths and drawbacks. This plethora of choices greatly increases the range of questions that can be asked, and answered, using microscopy.

Nonetheless, there remain many open questions with regards to best practices. The introduction of new techniques leads to new possibilities; however, there are also potential pitfalls. Old knowledge can be erroneously overlooked, sometimes simply because, over centuries, the textbooks written on microscopy and the tips and tricks of the trade have become so extensive that it is easy to miss certain parts of them. This can paradoxically lead to the latest state-of-the-art techniques not making use of some of the most basic, established, best practices.

Additionally, while it is often said that an image is worth a thousand words, quantifying exactly what one says is challenging. The entire field of computer vision is dedicated to the task. In some industries this may be for the purpose of improving models of vision for the next generation of self-driving cars. In the basic sciences, it may be to extract data from an image to generate results with statistical power. Neither of these tasks is straightforward.

Sometimes the most straightforward questions to answer qualitatively are the most difficult to quantify. Such is the case when discussing information in an image. At a glance, it may be possible to say that one image conveys more information to you than another. However, why exactly this is the case is not easy to articulate. Even so, it is often of fundamental importance to understand the information content of an image. This may be for evaluating changes in an optical path or discerning true signal from background signal from noise.

In this chapter we will delve into some techniques to address a number of these topics. We begin with the introduction (or reintroduction) of some strategies for image processing and feature

extraction, with suggestions on their applications to SRS microscopy. The final section of the chapter introduces a metric which allows us to quantify the information, in the common meaning of the word, contained within an image.

## 3.2 Results

### 3.2.1 The importance of preprocessing

Early on in the history of microscopy it became clear that the manner in which you illuminate a sample could affect the images observed. Even without a full understanding of geometric optics, or the optical transfer function, microscopists were noticing that introducing apertures or changing the illumination source or angle were having effects. By the end of the 19th century, as light sources and microscopes were becoming more standardized, Koehler introduced a new type of illumination to evenly illuminate the sample without also imaging the light source itself (Koehler, 1894; Evenett, 1983).

As more advanced techniques, such as fluorescence microscopy, began to become the workhorses of the life sciences, further adjustments needed to be made. When making quantitative measurements of fluorescence intensities, it is of great importance that the FOV illumination is considered in the analysis. Beginning with epi-fluorescence microscopy, in particular, microscopists realized the need for flatfield corrections (Siebert, Boone, and Lindfors, 1998). I.e. using an image of a homogenous field of the dye (or other fluorophore) you could determine the illumination profile of your imaging system. This could then be used to correct your image to account for any inhomogeneity in the illumination. Very commonly, for instance, the corners of a FOV will receive less light, due to vignetting from circular optics, or apertures, or any of numerous other possible causes. While not ideal, provided you have access to a flatfield image, this deviation from a truly even illumination can be accounted for so that the intensity of voxels in the corner of an image can be directly compared to the intensity in the center. This is accomplished through the simple equation 3.1 below:

$$\tilde{I} = \frac{\bar{B}I}{B} \quad (3.1)$$

where  $I$  is the image,  $B$  is the flatfield image,  $\tilde{I}$  is the corrected image and  $\bar{\cdot}$  denotes the mean. Unfortunately, it is not always possible to easily obtain a flatfield image; however, in this case it is still possible to perform a correction by generating a "pseudo-flatfield" image. One such way to do so is through low-pass filtering of any image acquired with the imaging system. Ideally one would use an image that contains as uniform a distribution of signal across the FOV as possible, although in principle any image works. Low-pass filtering retains the lower spatial frequencies that contribute to the signal in the image. The zero frequency term approximates the intensity distribution in the absence of higher order structure, and thus the illumination profile. One way of performing this operation is through convolution with a Gaussian filter produced with a very large radius, e.g., half the length of the image (in pixels/voxels), as indicated below in equation 3.2.

$$B = G_{\sigma=200} \otimes I \quad (3.2)$$

where  $\otimes$  denotes convolution.

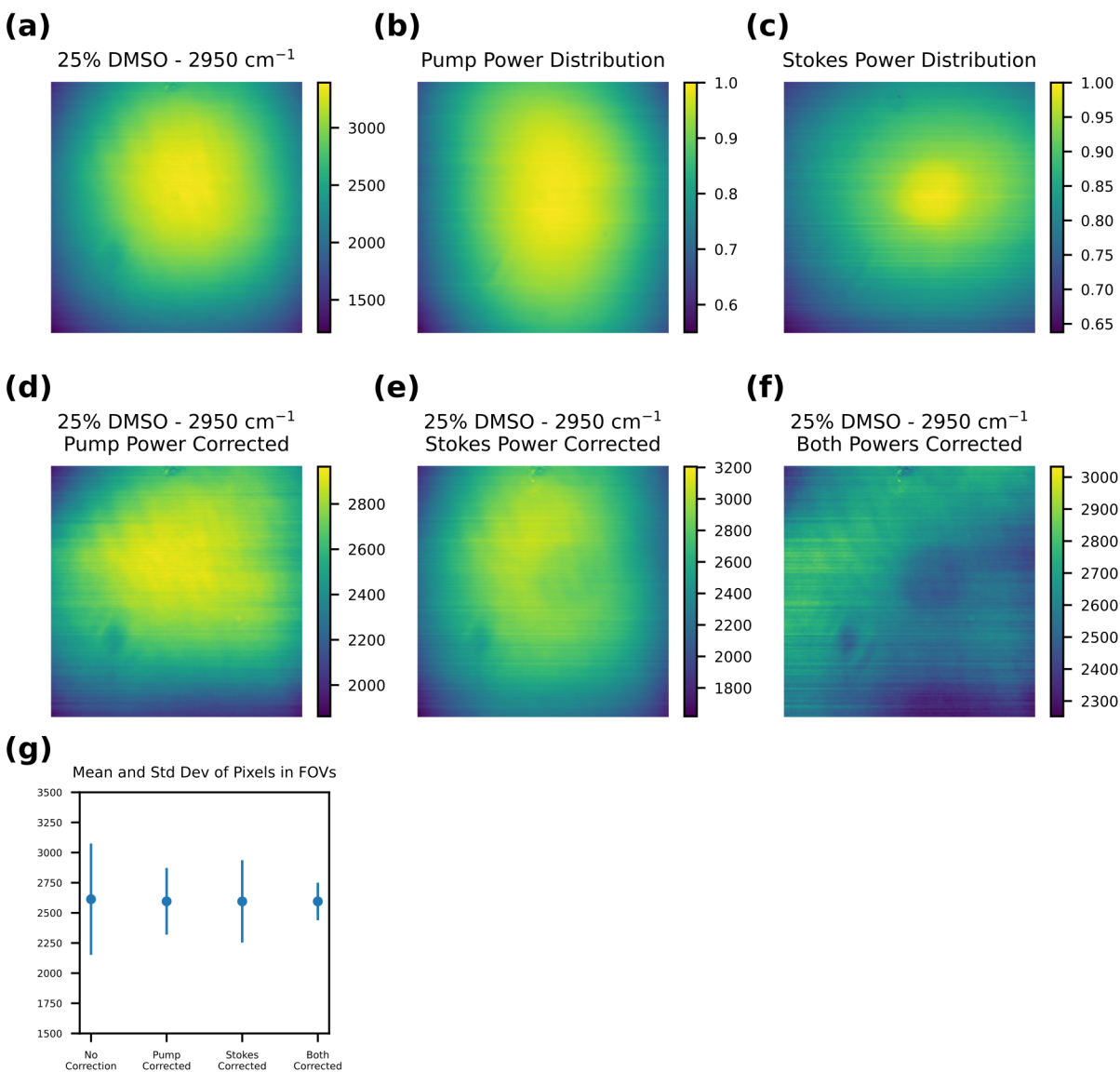
While this procedure is now commonplace in fluorescence microscopy, it is surprisingly not a routine preprocessing procedure in SRS microscopy. Very few studies perform this step; however, it is even more important in SRS, and pump-probe microscopy generally as there are multiple beams upon which the signal depends.

As a recent study shows, in order to perform the most accurate absolute concentration quantifications using SRS it is a necessary step (Oh et al., 2022). In fact, in the ideal scenario (barring perfectly even sample illumination) the image will be independently corrected for three major factors:

1. The pump beam illumination of the sample.
2. The Stokes beam illumination of the sample.
3. Any variation in the pump beam transmission from the sample to the detector.

The first two factors are necessary to correct for because the signal at the sample is generated by both the pump and Stokes beams. The Stokes beam is eventually filtered out in the microscope described in detail in chapter 2. As such, the characteristics of its propagation through the rest of the imaging system after the sample have no effect on the detected signal. The third factor is necessary to correct for as the pump beam is actually detected, and its propagation through the imaging system may have a spatial dependence. I.e., as it is scanned through different angles it may be blocked to a lesser or greater degree by any apertures in the system.

The first and third factor are simple to measure simultaneously, as the raw, 0 Hz output from the detector can be measured as a function of spatial position of the pump within a FOV (figure 3.1b). Factor 2 is rather more difficult to account for in isolation. As an approximate solution, the power distribution can be measured at the detector, analogously to the pump, after removal of the short pass filter on the detection arm. While this also includes effects from differences in the propagation of the beam after the sample, the use of high NA collection optics, as employed in the microscope, should limit the variation in transmitted light as a function of scanned angle. The measured distribution of the Stokes beam using this procedure is shown in figure 3.1c.



**Figure 3.1:** Demonstration of the importance of flatfield corrections

(a) Uncorrected SRS image acquired at  $2950\text{ cm}^{-1}$  of a homogenous field of 25% DMSO. The image is displayed at the maximum contrast which includes all the recorded intensity values. (b-c) The power distributions of the pump and Stokes beams across the field of view, respectively. Each power distribution is normalized to its maximum. (d-e) The image in (a) corrected for the uneven pump power distribution (d), uneven Stokes power distribution (e), or both distributions (f). Correction for the power distributions allows for finer details to become clear. As with (a), all images are shown to maximize the contrast while displaying the entire range of values. (g) The mean and standard deviation of the images in (a), (d), (e), and (f). A large percentage of the variance in the field of view could be explained by the power distributions of the excitation beams.

As an illustration of the necessity of flatfield correction, a nominal flatfield image is acquired of a 25% mixture of DMSO and water at a wavenumber of  $2950\text{ cm}^{-1}$ . The image is presented in figure 3.1a. While this image on its own could be used as a flatfield image for further image correction, we perform a correction on it using the measured pump and Stokes; power distributions.

The results of correcting for just the pump power distribution, just the Stokes' power distribution, or both are shown in figure 3.1d-f. Note that all images are displayed at the maximum contrast that includes all values within the FOV (i.e., does not clip the distribution). Interestingly, there is additional structure in apparent in figure 3.1f. This likely arises from imperfections in the glass surface the DMSO was imaged on as well as small amounts of evaporation or motion cause by the excitation lasers. In figure 3.1g, the mean and standard deviation of the pixels in each of the presented FOVs is plotted. As expected, a large amount of the variation present in the uncorrected FOV can be attributed to the power distributions of the illumination lasers. Nonetheless, even in the nominal "flatfield" image imperfections remain. Given this observation, we propose the following options for flatfield correction in order of preference:

1. Flatfield correction based on the measured power distributions of the pump and Stokes beam.
2. Flatfield correction using a measured flatfield image.
3. Pseudoflatfield correction using a low-pass filtered image.

Images presented in the remainder of this dissertation have all undergone one of these three preprocessing corrections.

### **3.2.2 Cellular and sub-cellular segmentation**

One of the most basic, and arguably most important, questions that can be asked about an image is where are the boundaries between objects? When quantifying metabolic composition of cells, as SRS microscopy is often used to do, it is important to have an objective methodology for defining objects. What constitutes an object depends on the nature of the sample under investigation. For tissue samples, the objects of interest may be cells and their boundaries. When studying single cells, it may be the subcellular structures, such as organelles. Often, traditional techniques for defining these objects begin with contrast enhancement, and gradient based edge finding algorithms. One simple approach is to use thresholding. This can be done in an automated way using an algorithm such as Otsu's method, (Otsu, 1979) or balanced histogram thresholding (dos Anjos and Shahbazkia, 2008). These types of thresholding algorithms tend to separate out objects. On the other hand, ad hoc thresholds can be determined for background/foreground separation. Different gradient operators, such as the Sobel (Pattern, 1973), or Canny (Canny, 1986) can provide information on sharp line boundaries.

Using the images produced by these thresholding and gradient operations, object boundaries can often be defined using a variety of techniques, e.g., a watershed-based algorithm, (Lantuejoul and Beucher, 1979) which imagines filling basins from minima in the images and draws boundaries where the watersheds meet. Implementations of these techniques can be found in all major programming languages, and are also included in many widely available image analysis software suites, like Fiji/ImageJ. (Schindelin et al., 2012)

In recent years, machine learning approaches have begun to become more prominent. In particular, convolutional neural networks (CNN) have undergone rapid development in the past decade (Gu et al., 2018). While once the subject of theoretical computer vision work, they are now being applied widely in the analysis of scientific images. This includes the analysis of SRS micrographs,



with numerous studies being published in the past five years (Manifold et al., 2019; Zhang et al., 2020; Abdolghader et al., 2021; Feizpour et al., 2021; Zhang et al., 2021).

Despite the growing appreciation for machine-learning based models, particularly for SRS microscopy it can be experimentally challenging to acquire the data needed to successfully implement one. This is because accurate machine-learning models require extensive amounts of training data. Even with the signal increases provided by SRS over spontaneous Raman, it can be a tedious, and sometimes infeasible, proposition to acquire this data, given the time investment required. Sometimes data needs to be extracted from a small number of images which would be insufficient to train a CNN on. Perhaps the sample itself is limited, or there are time constraints on obtaining the results.

For these situations, more traditional computer vision techniques need to be employed to extract quantitative data from images. We have found that the following algorithms perform well in providing accurate cell-boundary, or subcellular, segmentations of real SRS data. These algorithms are used for the data extraction in the work presented in chapter 4, and are presented in detail here.

---

### *Algorithm - Cell segmentation*

---

1. Flatfield or pseudoflatfield correct the image  $I$ .
2. Local histogram equalization using  $n \times n$  pixel grid size.  $n$  should be chosen based on the size of the average cell (in pixels) given the FOV size. The tile of the grid should be able to fully encompass the cell. Default value is  $n = 50$
3. Adaptive thresholding using a Gaussian-weighted sum of the neighborhood values. Pixel neighborhood (block) size is larger than the histogram grid size. A size of 200 pixels is the default value.
4. Morphological opening using a small radius structuring element. This is set to around 3x3 pixels to remove noise while maintaining the shape of cells.
5. Morphological closing to remove any holes remaining in the image.
6. Calculation of the Laplacian of the image.
7. Border following algorithm for contour tracing after the algorithm presented in (Suzuki and Be, 1986)
8. Remove internally lying contours. The algorithm of the prior step is often implemented to provide results in a hierarchical manner.

---

### *Algorithm - Subcellular segmentation*

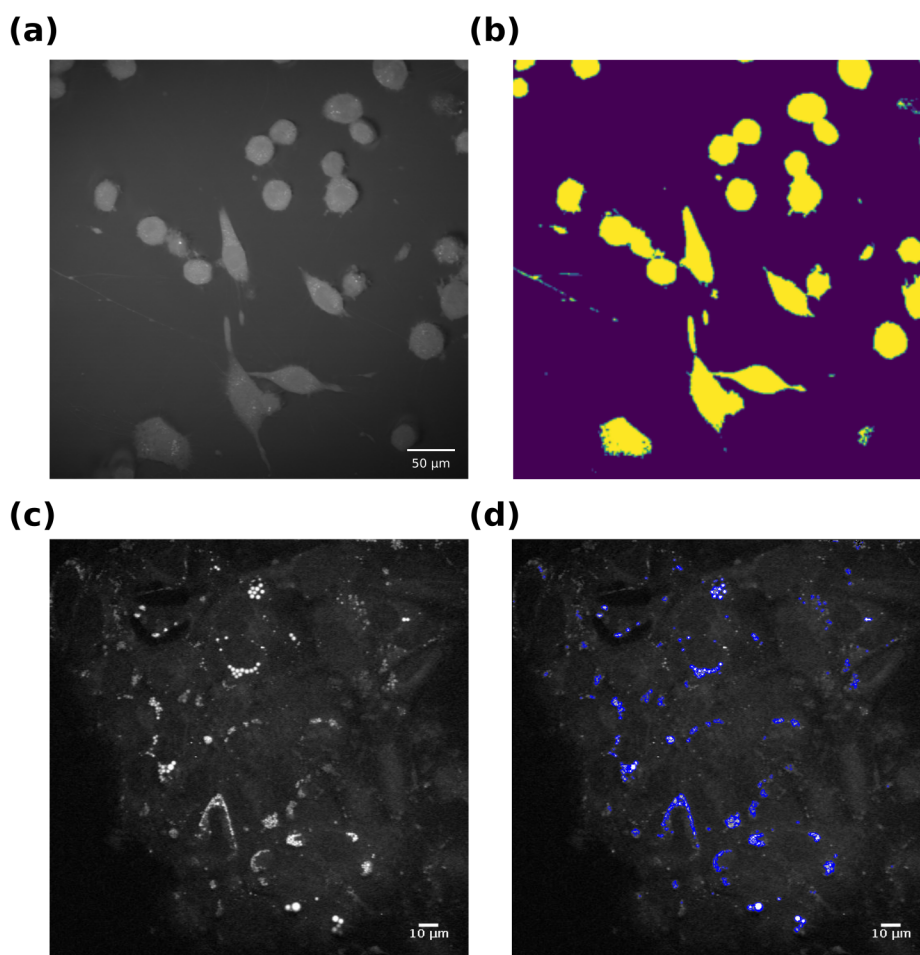
---

1. Flatfield or pseudoflatfield correct the image  $I$ .
2. Local histogram equalization using  $n \times n$  pixel grid size.  $n$  should be chosen based on the size of the average cell (in pixels) given the FOV size. The tile of the grid should be able to fully encompass the cell. Default value is  $n = 50$
3. Tophat filtering using a structuring element on the size order of 8x8 pixels.
4. Local histogram equalization with a fine grid size. Default value is  $n = 5$
5. Tophat filter with a structuring element on the order of 2x2 filters. Subtract the result from the previous step.

6. Border following algorithm for contour tracing (Suzuki and Be, 1986).
7. Remove outermost contours to capture the smaller internal features.

Contour finding:

An example of the usage of the first algorithm is presented in chapter 4, figure 4.4a, and reproduced below in figure 3.2a-b. The cellular segmentation was used to facilitate the quantification of intracellular neurotransmitter concentrations, discussed in greater detail in the corresponding section. Figure 3.2c-d presents a second example for the segmentation of lipid droplets. Both the number and integrated intensity could be quantified in this manner. This approach was used for validation in our publication (Munoz et al., 2022). A chief benefit of these two algorithms, is that they are capable of operating on, and extracting features of varying size, from images with different SNRs. With minimal modification, they have been successfully used to quantify different metabolic targets in different cell lines.



**Figure 3.2:** Cellular and sub-cellular segmentation of cells

**(a-b)** Original SRS image at  $2950\text{ cm}^{-1}$  and the cellular masks produced from it for PC12 cells, cultured as described in chapter 4. **(c-d)** The original CARS image at  $2950\text{ cm}^{-1}$  induced pluripotent stem cells (iPSCs), and the sub-cellular segmentation of internal lipid droplets within it.

### 3.2.3 Multiscale entropy

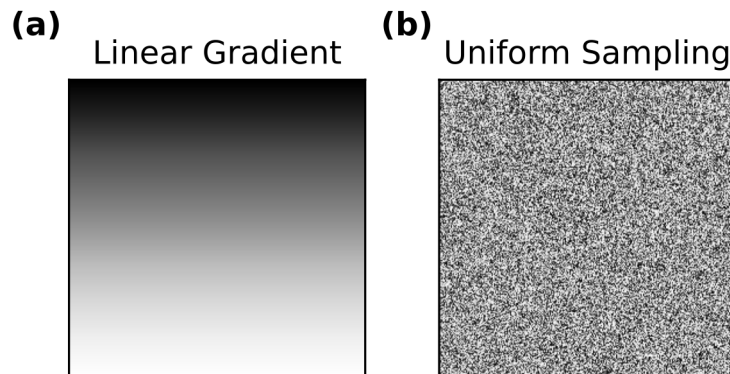
In the preceding sections, attention was paid to the structure of the signal in an image and how to extract information from one. However, we have yet to introduce a way to objectively evaluate the quantity of information contained in an image. In the field of information theory, information is often computed through the use of Shannon's entropy (MacKay, 2003), calculated using the equation below.

$$-\sum_{x,y} P(I(x,y)) \log_2(P(I(x,y))) \quad (3.3)$$

where the summation runs over the pixels,  $x$  and  $y$ , of the image  $I$ , and  $P$  denotes the probability.

This provides a measure of the uncertainty or "surprisingness" of a signal. However, entropy as formulated has a number of shortcomings when applied to images. To begin with, uncertainty does not coincide well with our colloquial usage of the term information. An 8-bit image contains maximal entropy if every pixel value,  $[0, 255]$ , is equally likely. I.e. an image of uniformly distributed noise is maximally entropic as each pixel provides new information (figure 3.3b). However, when observing this image, no information would be conveyed.

Another potential shortcoming is that the entropy calculated for any given image might not be reflective of the information depth and dynamic range for the system as a whole. For example, any given image from a set of many may be composed of pixels over a relatively small intensity distribution, while the set as a whole covers a much more extensive range. The entropy as calculated for any image may be lower than that calculated over the set. In this scenario, comparing the information content across images, or between sets then becomes challenging.



**Figure 3.3:** Two 8-bit images with 8 bits of entropy

**(a)** A 256x256 8-bit image of a linear gradient of values from 0 to 255. Each row of the image has a single value which is incremented by one for each subsequent row. E.g the first row has a value of zero, the second a value of one, the third a value of two and so on. **(b)** A 256x256 8-bit image with random sampling from a uniform distribution  $[0, 255]$ . Both images have an entropy of 8 as each value from 0 to 255 is equally probable. The colorscale runs from black representing zero to white representing one.

Furthermore, entropy as defined above does not account for structure, simply the raw intensity distributions measured. This can result in somewhat unintuitive results. For example, the aforementioned 8-bit image with random sampling from a uniform distribution,  $[0, 255]$ , will have an entropy of 8, as each value is equally likely to occur (figure 3.3b). A 255x255 image of a linear gradient where each row has the value corresponding to its index minus 1 (i.e. row 1 has a value of zero across the row, row 2 has a value of one, and so on) also has an entropy of 8 (figure 3.3a). Nonetheless, these two images clearly convey different

information. In the strictest sense of entropy, only the first image should have an entropy of 8; however, in terms of structural content the linear gradient clearly contains more 'information.'

A measure of information/structural content, however, can be of much use in SRS microscopy, and microscopy in general. In particular, it can sometimes be difficult to discern by eye whether a modification to the beam path has been beneficial. Certain characteristics of a system are measured, such as the noise spectrum of a signal; however, there is no general consensus on how to measure the properties of the final completed image, which is, of course, the object ultimately of interest.

Additionally, in the case of pump-probe microscopy, a number of background sources of signal (i.e. not noise) often contribute. The electrical detection may be operating at the shot noise limit; however, these additional sources do not register as noise, but rather signal. The largest determinant of these background signal sources in pump-probe microscopy tends to be intensity fluctuations due to the optical Kerr effect. The Kerr effect is a nonlinear optical effect which produces an intensity dependent refractive index of a material. This means that when two high peak power lasers are incident on a sample, each electric fields can modify the refractive index of the sample for the other, a process known as cross-phase modulation (XPM). The refractive index changes due to XPM can manifest as intensity changes when the small refractive index deviations lead to clipping of the beam. This results in a modulation-frequency dependent intensity change in the detected pump beam. I.e., exactly the signal we are hoping to detect in an SRS experiment.

As the predominant reason for their presence is the clipping of the beam on apertures, the easiest way to alleviate these signals is by using collection optics with a numerical aperture larger than the excitation optics (Berto et al., 2014); however, it is not always possible to completely eliminate all obstructions. The detected signals are on the level of hundreds of nanovolts to microvolts over a tens of millivolts DC background. To produce signals at these levels, clipping may often not be seen by eye, or even with an IR viewer.

As such, it is of general interest to have a metric to quantify information content, and in particular, one that is comparable across varying datasets. Ideally, the best metric would be invariant to the intensity of acquired signal; however, such invariance is practically infeasible due to the SNR dependence on intensity in a SRS measurement.

For a measure of structural information as opposed to uncertainty we propose another metric referred to as the multi-scale second order entropy (MSSoE). This metric is calculated using the second-order probability  $Q(P(\cdot))$ , i.e., the probability of the probability (Baron, 1987). Furthermore, we consider that information in an image can occur over multiple length scales, and spatial frequency distributions. E.g. in an image of a cell, high-frequency information can arise from the sub-cellular distribution of organelles, and how easy they are to discern, while lower-frequency information comes from the overall background signal of the cell. In order to include information from all of these spatial scales, the second-order entropy is computed for each level of a Laplacian pyramid  $\Delta\tilde{I}_n$ , and then summed across levels, as proposed by Rakshit and Mishra (Subrata and Mishra, 2006). To construct the pyramid, a scaling factor of 2 is used to a final layer size of 16x16. The final MSSoE equation is given by:

$$MSSoE = - \sum_n \sum_{x,y} Q(P(\Delta\tilde{I}_n \log_2(Q(P(\Delta\tilde{I}_n(x,y)))))) \quad (3.4)$$

Note that, other multiscale approaches can be used to, such as the discrete complex wavelet transform (Lina and Gagnon, 1995). These approaches, however, incur additional computational overhead, and we found that the Laplacian pyramid performs well for the quantification of information in real SRS images.

While the metric on its own is useful for comparison of images, it is also helpful to visualize which features and areas of an image most strongly contribute to the metric. In order to do so, two dimensional entropy maps,  $E_n(P(\Delta\tilde{I}_n))$ , can be constructed. The entropy map is computed beginning with the final layer of the pyramid, and successively upscaling by 2 through bilinear interpolation. The maps produced at each layer of the pyramid are summed together. As a mechanism to suppress noise, the final entropy map

(produced from the sum) is normalized to the first order entropy map calculated over the first layer. Taking  $U$  to be the operator performing the upscaling operation, and the summation occurring over decreasing  $n$ , the final entropy map is given by the following equation:

$$\frac{\sum_n^1 E_{n-1}(P(\Delta\tilde{I}_{n-1})) + U(E_n(P(\Delta\tilde{I}_n)))}{E(\Delta\tilde{I}_0)^{0.5}} \quad (3.5)$$

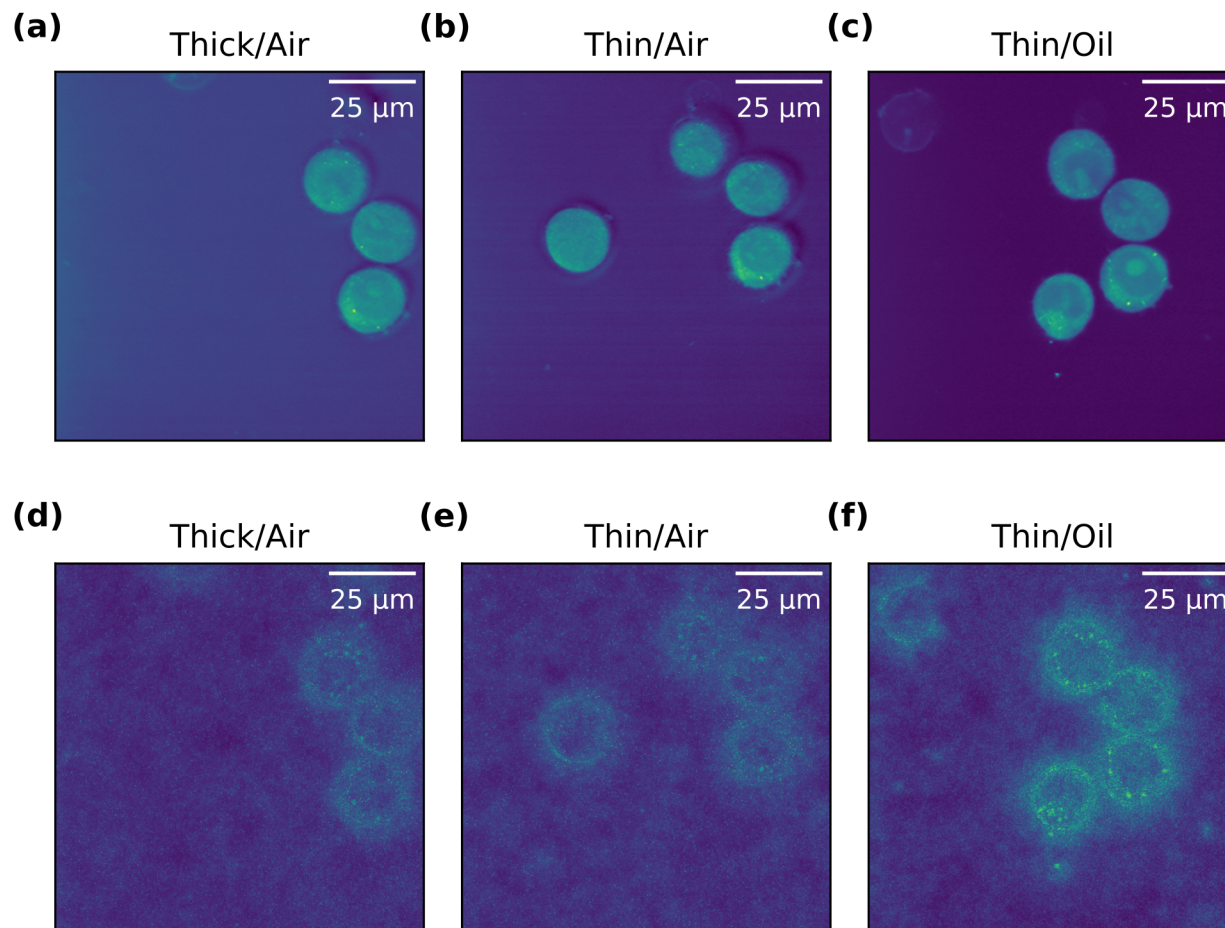
We first applied this metric in our publication Kim et al. (Kim et al., 2021). As mentioned previously, the biggest contributor to background signal in SRS is the Kerr effect in the form of XPM. In order to minimize the problem we use a high numerical aperture collection condenser; however, since the excitation objective typically employed on our SRS microscope is a 1.2 NA water immersion objective, the only condenser with a larger NA is a 1.4 NA oil immersion condenser. This makes sample mounting difficult as the sample must be enclosed on both sides in order to accommodate the two immersion fluids. Furthermore, due to the very high NA of the optics, there is very little space to work, as such it is challenging to manipulate a sample once it is on the microscope.

For many applications, however, it is desirable to watch a change in a detected SRS signal from a sample in response to some stimulus. For example in chapter 4, we will look at the response of differentiated neuron-like cells to potassium stimulation to monitor neurotransmitter release. Given the geometric constraints of the imaging system, one possible solution to allow for sample manipulation that is compatible with the microscope is the use of microfluidics.

Even so, typical microfluidic devices are thick relatively thick compared to the total working distance provided by the objective and condenser optics. The geometry requires a total sample thickness less than 1.6 mm. In Kim et al., a new microfluidic fabrication protocol was proposed which produces "paper-thin" microfluidic devices of a few hundred microns thickness. The key aspect of this device is that it employs multiple layers, which allows for the precise manipulation of cells or reagents using fluidic circuits and valves.

In order to evaluate the compatibility of the microfluidic devices, produced with this fabrication protocol, with SRS imaging, we performed experiments to measure the information content of images produced using the device under a number of conditions. HEK cells were grown in  $D_2O$  containing medium, or normal growth medium, and imaged at two wavenumbers,  $2950\text{ cm}^{-1}$  and  $2135\text{ cm}^{-1}$ . Cells grown in normal growth medium should have a signal at  $2950\text{ cm}^{-1}$  but not at  $2135\text{ cm}^{-1}$ , as the former corresponds to C-H stretching modes, while the latter to C-D stretching modes. On the other hand, cells grown in  $D_2O$  containing medium should have measurable SRS signals in both wavenumber channels, as the  $D_2O$  is incorporated into C-D bonds in biomolecules.

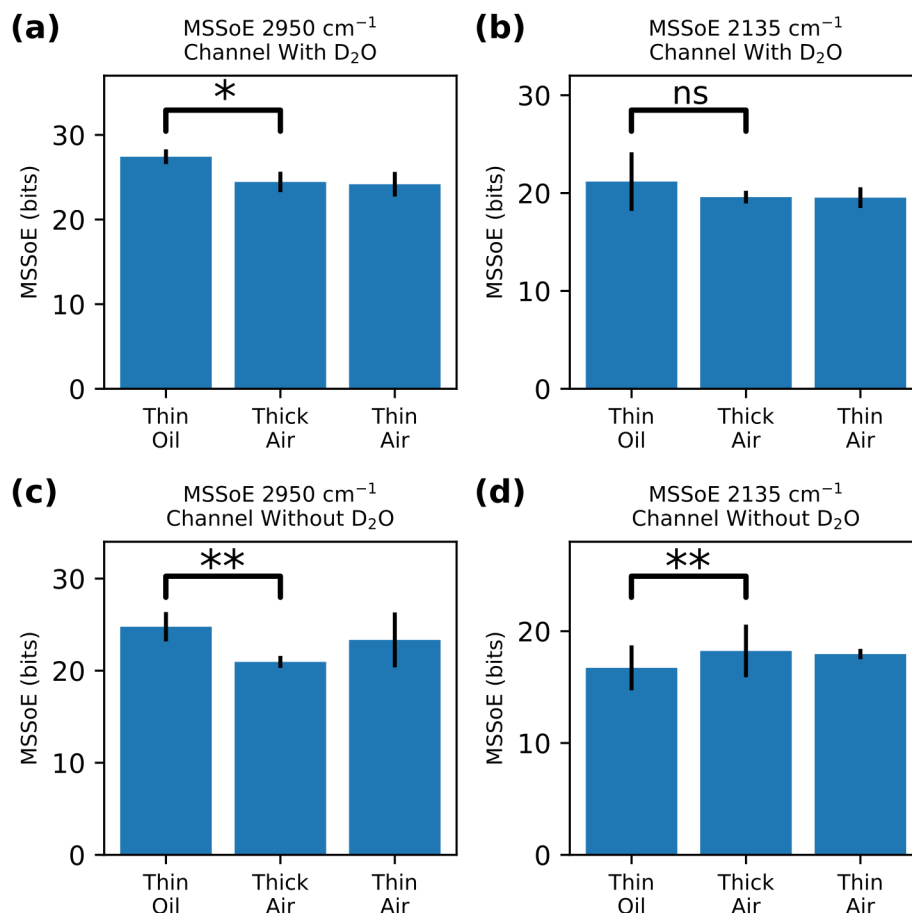
In order to test this hypothesis, the same cells were imaged in three microfluidic device/condenser combinations. The first combination employed the new thin device and the oil condenser (the Thin/Oil condition). This is the combination which corresponds to the optimal imaging conditions for SRS microscopy. The second combination was using the thin device and a long working distance air condenser (the Thin/Air condition). The numerical aperture of the air condenser is lower than that of the excitation objective, so signals at  $2135\text{ cm}^{-1}$  are expected to become apparent, even for cells that were not grown in  $D_2O$ . Finally, to emulate the effect of the extra PDMS material that would be present in a conventionally-fabricated microfluidic device, an extra block of PDMS, approximately 1 cm thick, was placed on top of the thin device, and the cells were reimaged under the air condenser (the Thick/Air condition).



**Figure 3.4:** SRS and MSSoE images of HEK cells incubated with  $D_2O$  acquired using three microfluidic device/condenser combinations

**(a-c)** SRS images of HEK cells at  $2950\text{ cm}^{-1}$  for three microfluidic chip/condenser combinations. The cells were incubated with  $D_2O$  for 20 hours prior to image acquisition. The three cells along the right side are the same three cells in all FOVs. Due to some sample movement between acquisitions a fourth cell appeared and is the same cell in (b) and (c), but is absent from the image in (a). All images are displayed using the same color scale. **(d-e)** The multiscale second order entropy images corresponding to the images in (a-c). All images are displayed using the same color scale. The brighter green-yellow color is indicative of greater information content in that region of the image.

Example images acquired at  $2950\text{ cm}^{-1}$  are displayed in figure 3.4a-c, for cells grown in  $D_2O$ . There was some sample motion during the transition between the three imaging conditions; however, the three cells on the rightmost side of the image are the same between all FOVs. Images acquired in the Thin/Oil condition are the clearest, with the most distinguishable internal structure. The extra background present in the Thick/Air condition arises from XPM occurring in the media itself. These qualitative observations are confirmed by the MSSoE images presented in figure 3.4d-f, both large-scale spatial features, such as the entire cell body, and smaller-scale features such as lipid droplets, contribute to the metric.



**Figure 3.5:** Calculated MSSoE of images acquired of HEK cells using three microfluidic chip/condenser combinations

(a-b) MSSoE values of images of HEK cells incubated with D<sub>2</sub>O at 2950 cm<sup>-1</sup> (a) and 2135 cm<sup>-1</sup> (b) for each of the three microfluidic chip/condenser combinations. At first it may be surprising that the difference between the oil/thin and air/thick conditions at 2135 cm<sup>-1</sup> is not significant for D<sub>2</sub>O incubated cells; however, this anticipated as XPM in the air/thick condition generates additional contrast increasing the overall information content of the images. This extra information, however, is non-chemically specific and not related to the Raman signal of interest. (c-d) MSSoE values of images of HEK cells which were not incubated with D<sub>2</sub>O at 2950 cm<sup>-1</sup> (c) and 2135 cm<sup>-1</sup>.

\*\* One-sided Mann-Whitney U = 0, n = 5, significance level = 0.005

\* One-sided Mann-Whitney U = 0, n = 3, significance level = 0.05

ns: Not Significant

The calculated MSSoE values for cells imaged under the three condenser/device conditions are presented in figure 3.5. Panel a and b show the metric results for cells grown in D<sub>2</sub>O imaged at 2950 cm<sup>-1</sup> and 2135 cm<sup>-1</sup>, respectively. Panels c and d show the analogous results for cells grown in normal growth medium without D<sub>2</sub>O. As anticipated, images acquired using the Thin/Oil combination contained the most information in the 2950 cm<sup>-1</sup> channel, regardless of the presence of D<sub>2</sub>O in the growth medium, as the background due to XPM was minimized. This reduction of the background signal increased the total in-

formation as features of the cells could more readily be distinguished. For cells grown without D<sub>2</sub>O in the growth medium, imaging at 2135 cm<sup>-1</sup> with the Thin/Oil combination produced images with significantly less information content, as anticipated. The low profile of the new device allowed for XPM signals to be minimized, leading to a reduction of the total signal, and thus information content, when no SRS interaction was expected to be present.

For cells incubated with D<sub>2</sub>O, in the 2135 cm<sup>-1</sup> channel there was no observed significant difference in the information for the cells measured using the Oil/Thin combination versus the Air/Thick combination. At first this result may seem counterintuitive; as the other conditions illustrate, the oil condenser and thin chip presumably provides a greater amount of information. Nonetheless, this actually reveals something of fundamental importance. Regardless of the relative intensity of the signals arising from the Raman interaction, or the Kerr effect, at the structural level of the image, the latter contributes an amount of information on roughly the same order as the former. This means that unaccounted for, the signals arising from XPM can greatly alter our perception and interpretation of the acquired data.

### 3.3 Discussion

In this chapter, a number of computational tools and best practices have been presented for the analysis and interpretation of SRS images. While simple techniques such as flatfield corrections have been around for a long time, their use in SRS microscopy is not commonplace. Yet, given the complexity of the measurement, compared to the simplicity of the procedures, they should be. SRS measurements are challenging to make and every effort should be made to streamline and regularize the analysis of the resultant data.

While algorithms for feature extraction are abundant, their successful application to one set of problems is no guarantee of successful application to another. The origins of signals and backgrounds in SRS microscopy are distinct from those in fluorescence microscopy, for instance, and different approaches often need to be taken with this in mind. Nonetheless, image analysis in the field of SRS microscopy still often oscillates between two extremes of manual intervention and quantification, and advanced machine learning techniques. The former is both time intensive and not easily transferable to new datasets. The latter, on the other hand, is not applicable to a vast number of SRS microscopy experiments currently being performed due to an insufficient or unavailable training data. Simple algorithms that perform well across a variety of sample types commonly encountered in SRS microscopy help to fill this gap while new methods are continually developed.

Finally, in addition to understanding the fundamental noise characteristics of any detection system, it is helpful to have a way to characterize the performance based on the final end product - the image. This is not always simple to achieve, as the sources of noise and background signal can vary widely. This becomes increasingly true as the complexity of the imaging technique increases, and ever more possibilities need to be considered. The method described in this chapter, coupled with an understanding of possible signal sources in SRS (or pump-probe) microscopy, can be used to guide both routine maintenance and experimental design. It can be used to characterize the degree of alignment of the microscopy system being used for investigation by checking for the generation of additional signals due to XPM. In the same manner, it can help to address concerns about the origin of signals, and confirm the minimization of background in actual experimental data.



## 3.4 Materials and Methods

### 3.4.1 HEK293T cell culture

Human embryonic Kidney (HEK293T) cells (UC Berkeley cell culture facility) were cultured in Dulbecco's Modified Eagle Medium (DMEM) with L-glutamine and without pyruvate, including 10% fetal bovine serum (FBS), and 1% penicillin/streptomycin. Cells were passaged upon achieving between 70-100% confluency. Cells were imaged at 50-70% confluency. Cells were transferred to and imaged in the Live Cell Imaging Solution (Fisher Sci).

To deuterium tag cells, the D<sub>2</sub>O (Sigma) DMEM media was made by mixing 70% D<sub>2</sub>O (Sigma) with 30% deionized H<sub>2</sub>O to dissolve DMEM powder (Sigma), 10% fetal bovine serum (FBS), 1% penicillin/streptomycin, and 3.7 g/L sodium bicarbonate (Sigma). After pH adjusting to 7.4, the media was sterilized with a 0.2  $\mu$ m filter. The cell media was replaced with the D<sub>2</sub>O media at 50-70% cell confluency then incubated for 20-24 hours. The cells were transferred to and imaged in the Live Cell Imaging Solution with 70% D<sub>2</sub>O.

### 3.4.2 Undifferentiated induced-pluripotent stem cells (iPSCs)

iPSCs were cultured as described in (Munoz et al., 2022). Briefly, the cells were cultured in mTESR1 media supplemented with the Y-27632 2HCl inhibitor.

### 3.4.3 SRS imaging

The synchronized dual output of a femtosecond oscillator/OPO (Insight DS+, Spectra-Physics) provided the pump (tunable OPO output), and Stokes beams (fundamental at 1040 nm) for the experiment. The power of both beams was controlled using two variable attenuators consisting of a half-wave Fresnel rhomb (for the pump), or half-wave plate (for the Stokes), and a polarizer. The pump beam was tuned to 796 nm for CH<sub>2</sub>/CH<sub>3</sub> imaging, and 851 nm for off-resonance imaging, corresponding to Raman shifts of 2950 cm<sup>-1</sup> and 2135 cm<sup>-1</sup>. The actual spectral bandwidth was 160 cm<sup>-1</sup> due to the short pulse duration of 150 fs. The output from the Stokes beam was then intensity modulated at 10.28 MHz using a quarter waveplate (Thorlabs), a resonant electro-optic modulator (EOM; Thorlabs) and a Glan-laser polarizer (Thorlabs). The resonant EOM was driven by a function generator (33120A, Hewlett Packard), and an additional power amplifier (ZHL-32A+, Minicircuits). The pump beam path length was controlled by an optical delay line (FCL200, Newport) to ensure coincident arrival of the pulses of the two beams. The beams were then combined on a 1000 nm short-pass dichroic mirror (Thorlabs), and fed into the scan head of a confocal scanning microscope (FV1200, Olympus). The beams were delivered to the sample by a near infrared optimized 60 $\times$  water immersion objective with a numerical aperture (NA) of 1.2 (UPLSAPO60XWIR, Olympus). After the sample, the lasers were collected by a 1.4 NA oil condenser (CSC1003, Thorlabs; or equivalently D-CUO, Nikon) or 0.8 NA air condenser. The Stokes beam was then filtered by a 1000 nm shortpass filter (Thorlabs). A series of relay lenses delivered the pump beam to a photodiode (S3590-08, Hamamatsu), reverse biased at 61.425 V (Keysight). The signal from the photodiode was bandpass filtered and fed into a lock-in amplifier (HF2LI, Zurich Instruments) for image formation. For HEK293T cells, the power of the pump and Stokes beams was set to 30 mW at the sample, and the images were acquired at a size of 512  $\times$  512 pixels using a pixel dwell time of 2  $\mu$ s and a lock-in time constant of 1  $\mu$ s. For iPSCs, a pixel dwell time of 10  $\mu$ s and a lock-in time constant of 3  $\mu$ s were used. The power of both the pump and Stokes beam was set to 10 mW.

## 3.5 References

- Abdolghader, P., Ridsdale, A., Grammatikopoulos, T., Resch, G., Legare, F., Stolow, A., Pegoraro, A. F., and Tamblyn, I. (2021) “Unsupervised hyperspectral stimulated Raman microscopy image enhancement: denoising and segmentation via one-shot deep learning”, *Optics Express*, Vol. 29 No. 21, pp. 34205-34219.
- Baron, J. (1987) “Second-order probabilities and belief functions”, *Theory and Decision*, Vol. 23, pp. 25-36.
- Berto, P., Andresen, E.R. and Rigneault, H. (2014), “Background-free stimulated raman spectroscopy and microscopy”, *Physical Review Letters*, Vol. 112 No. 5, pp. 1–5.
- Canny, J. F. (1986) “A computational approach to edge detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 679-698.
- Dos Anjos, A., and Shahbazkia, H. R. (2008) “Bi-level image thresholding – A fast method”, *Proceedings of the First International Conference on Bio-inspired Systems and Signal Processing*, pp. 70-76.
- Evennett, P. E. T. E. R. (1983) “Kohler illumination: a simple interpretation”, *Proc Royal Microsc Sc*, Vol. 28 No. 4, pp. 189-192.
- Feizpour, A., Marstrand, T., Bastholm, L., Eirefelt, S., and Evans, C. L. (2021) “Label-free quantification of pharmacokinetics in skin with stimulated Raman scattering microscopy and deep learning”, *Journal of Investigative Dermatology*, Vol. 141 No. 2, pp. 395-403.
- Gu, J., Zhenhua, W., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., and Tsuhan, C. (2018) “Recent advances in convolutional neural networks”, *Pattern Recognition*, Vol. 77, pp. 352-377.
- Hooke, Robert (1635-1703) “Micrographia, or, some physiological descriptions of minute bodies made by magnifying glasses: with observations and inquiries thereupon”, London: printed for James Allestry.
- Kim, S., Dorlhiac, G., Chaves, R. C., Zalavadia, M., Streets, A. (2021) “Paper-thin multilayer microfluidic devices with integrated valves”, *Lab on a Chip*, Vol. 21, pp. 1287-1298.
- Koehler, A. (1894) “New method of illumination for photomicrographical purposes”, *Journal of the Royal Microscopical Society*, Vol. 14, pp. 261-262.
- Lantuejoul, C., and Beucher, S. (1979) “Use of watersheds in contour detection”, *Proc. Int’l Workshop on Imaging Processing, Real-Time Edge and Motion Detection/Estimation*.
- Lina, J-M., and Gagnon, L. (1995) “Image enhancement with symmetric Daubechies wavelets” in “Wavelet applications in signal and image processing III”, Vol. 2569, pp. 196-207. SPIE.
- Manifold, B., Thomas, E., Francis, A. T., Hill, A. H., and Fu, D. (2019) “Denoising of stimulated Raman scattering microscopy images via deep learning”, *Biomedical Optics Express*, Vol. 10 No. 8, pp. 3860-3874.
- Masia, F., Glen, A., and Stephens, P. (2013) “Quantitative chemical imaging and unsupervised analysis using hyperspectral coherent anti-Stokes Raman scattering microscopy”, *Analytical Chemistry*, Vol. 85 No.

22, pp. 10820-10828.

MacKay, D. J. C. (2003) "Information theory, inference, and learning Algorithms", Cambridge University Press.

Munoz, A., Theusch, E., Kuang, Y-L., Nalula, G., Peaslee, C., Dorlhiac, G., Landry, M. P., Streets, A., Krauss, R. M., Iribarren, C., Mattis, A. N., and Medina, M. W. (2022) "Undifferentiated induced pluripotent stem cells as a genetic model for nonalcoholic fatty liver disease", *Cellular and Molecular Gastroenterology and Hepatology*, Vol. 14 No. 5, pp. 1174-1176.

Oh, S., Lee, C.H, Yang, W., Li, A., Mukherjee, A., Basan, M., Ran, C., Yin, W., Tabin, C. J., Fu, D., Xie, X. S., and Kirschner, M. W. (2022), "Protein and lipid mass concentration measurement in tissues by stimulated Raman scattering microscopy", *Proceedings of the National Academy of Sciences*, Vol. 119 No. 17.

Otsu, N. (1979) "A threshold selection method from gray-level histograms." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-9 No. 1, pp. 62-66.

Pattern, D. R. H. (1973) "Classification and scene analysis", John Wiley & Sons.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J-Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., and Cardona, A. (2012) "Fiji: an open-source platform for biological-image analysis", *Nature Methods*, Vol. 9, pp. 676-682.

Siebert, J. A., Boone, J. M., and Lindfors, K. K. (1998) "Flat-field correction technique for digital detectors", in "Medical Imaging 1998: Physics for Medical Imaging", Vol. 3336, pp 348-354, SPIE.

Subrata, R., and Mishra, A. (2006) "Estimation of structural information content in images", In: Narayanan, P. J., Nayar, S. K., Shum, H.Y, eds, *Computer Vision - ACCV 2006. ACCV 2006 Lecture Notes in Computer Science*, Vol. 3851. Springer, Berlin, Heidelberg.

Suzuki, S., and Be, K. A. (1985) "Topological structural analysis of digitized binary images by border following", *Computer Vision, Graphics, and Image Processing*, Vol. 30 No. 1, pp. 32-46.

Varinli, H., Osmond-McLeod, M. J., Molly, P. L., and Vallotton, P. (2015) "LipID-QuanT: A novel method to quantify lipid accumulation in live cells", *Journal of Lipid Research*, Vol. 56 No. 11, pp. 2206-2216.

Zhang, J., Zhao, J., Lin, H., Tan, Y., and Cheng, J-X. (2020) "High-speed chemical imaging by dense-net learning of femtosecond stimulated Raman scattering", *J. Phys. Chem. Lett.*, Vol. 11 No. 20, pp. 8573-8578.

Zhang, Q., Yun, K. K., Wang, H., Yoon, S. W., Lu, F., and Won, D. (2021) "Automatic cell counting from stimulated Raman imaging using deep learning", *PLOS ONE*, Vol. 16 No. 7.

# Chapter 4

## Neurotransmitter Isotopologue Imaging

\*This chapter is adapted from previously published work. An additional appendix has been included with results beyond the scope of the original report.

**Leveraging isotopologues as a general strategy to image neurotransmitters with vibrational microscopy.**

Gabriel F. Dorlhiac<sup>1</sup>, Markita P. Landry<sup>1,2,3,4,5\*</sup>, Aaron Streets<sup>1,4,5,6,7\*</sup>

<sup>1</sup>Biophysics Graduate Group, University of California Berkeley, Berkeley, CA, 94720, USA

<sup>2</sup>Department of Chemical and Biomolecular Engineering, University of California, Berkeley, Berkeley, CA, 94720, USA

<sup>3</sup>Innovative Genomics Institute, University of California Berkeley, Berkeley, CA, 94720, USA

<sup>4</sup>California Institute for Quantitative Biosciences, QB3, University of California, Berkeley, Berkeley, CA, 94720, USA

<sup>5</sup>Chan-Zuckerberg Biohub, San Francisco, CA, 94158, USA

<sup>6</sup>Department of Bioengineering, University of California, Berkeley, Berkeley, CA, 94720, USA

<sup>7</sup>UC Berkeley-UC San Francisco Graduate Program in Bioengineering, University of California, Berkeley, Berkeley, CA, 94720, USA

### 4.1 Abstract

Chemical neurotransmission is central to neurotypical brain function but also implicated in a variety of psychiatric and neurodegenerative diseases. The release dynamics of neurotransmitters is correlated with but distinct from electrical signal propagation. It is therefore necessary to track neurotransmitter modulation separately from electrical activity. Here, we present a new approach for imaging deuterated neurotransmitter molecules with vibrational microscopy in the cell-silent window. Using stimulated Raman scattering microscopy, we perform direct imaging of the neurotransmitters dopamine and GABA in PC12 chromaffin cells, and primary hippocampal neurons, respectively, based on their C-D vibrational frequencies. We find that isotope labelling does not perturb neurochemical activity. We further show that stimulation of neurotransmitter release results in an overall 20-50% signal reduction, with the ability to observe inter- and intracellular variation

in vesicular release potential. Taken together, our data suggest that neurotransmitter isotopologues can serve as a generic, non-perturbative, method to image neurotransmitters.

## 4.2 Introduction

Information flow in the nervous system is mediated by both electrical and chemical transmission processes, an understanding of both of which is necessary to study neurotypical activity, and any deviations which can result in neuropathology. Neuronal stimulation results in intracellular electrical signal propagation, in the form of action potentials, on the millisecond timescale. Interneuronal communication, however, is most often mediated by small molecule, i.e. neurotransmitter, release at the neuronal synapse. Canonically these neurotransmitters are locally confined to the sub-micron sized synapse, leading to rapid communication shortly following action potential arrival. One class of neurotransmitters, known as neuromodulators, are thought to exhibit additional extrasynaptic diffusive action, which may modulate broader neuronal networks on the order of microns and seconds. Frequently, the release of these compounds is overestimated relative to action potential arrival, as in the case of dopamine, where estimates of the correlation between action potential activity and dopamine release are as low as 6-10% (Grienberger and Konnerth, 2012; Südhof, 2004). Furthermore, abnormal regulation of neuromodulator activity is associated with a wide variety of neuropathologies, including schizophrenia (Kesby et al., 2018), epilepsy (Starr, 1996), Parkinson's (Warren et al., 2017), and Huntington's disease (Chen et al., 2013), and addiction (Nutt et al., 2015), among others. The study of chemical neurotransmission is, therefore, of fundamental importance to our understanding of neuropathologies.

Owing to the high spatiotemporal resolution of fluorescence imaging, its adaptability to image whole organs and organisms, and the broad abundance of fluorescence-based probes to image neurons and neuronal activity, fluorescence-based techniques have been widely adopted to study the brain and the process of neurotransmission. However, fluorescence-based approaches face a major obstacle when imaging neurochemical communication, in that neurotransmitters are small molecules that do not intrinsically fluoresce in the visible range. Therefore, several approaches have been taken to enable fluorescence-based measurements of neurotransmitter activity, including synthesis of fluorescent chemical analogues, the use of genetically encoded receptors co-expressed with fluorescent proteins with affinity for the neurotransmitter of interest, and readout of neuron activity by fluorescent reporters as a proxy for neurotransmitter release. For instance, monitoring of vesicular fusion can be accomplished through introduction of exogenous amphipathic FM dyes (Betz et al., 1992), or by using the acidic vesicular environment to quench genetically encoded pHluorin sensors (Sankaranarayanan et al., 2000), which report on fusion when exposed to the neutral synaptic environment. Yet neither technique reports on the neurotransmitter populations themselves inside the vesicles, nor whether they are released. Other genetically encoded fluorescent receptors report binding of the neurotransmitter itself. For example, cell-based neurotransmitter fluorescent engineered reporters (CNiFERS), linking neurotransmitter-binding G-protein coupled receptors to fluorescent readout, have been used to study a number of different neurotransmitters (Arroyo et al., 2016; Lacin et al., 2016). This technique, however, requires the transplantation of reporter cells into neuronal tissue, exhibiting low spatiotemporal resolution relative to neurochemical signaling. To increase spatiotemporal sensitivity of neurotransmitter imaging, GRABDA was developed by engineering dopamine receptors with circularly permuted eGFP, to be expressed in cell

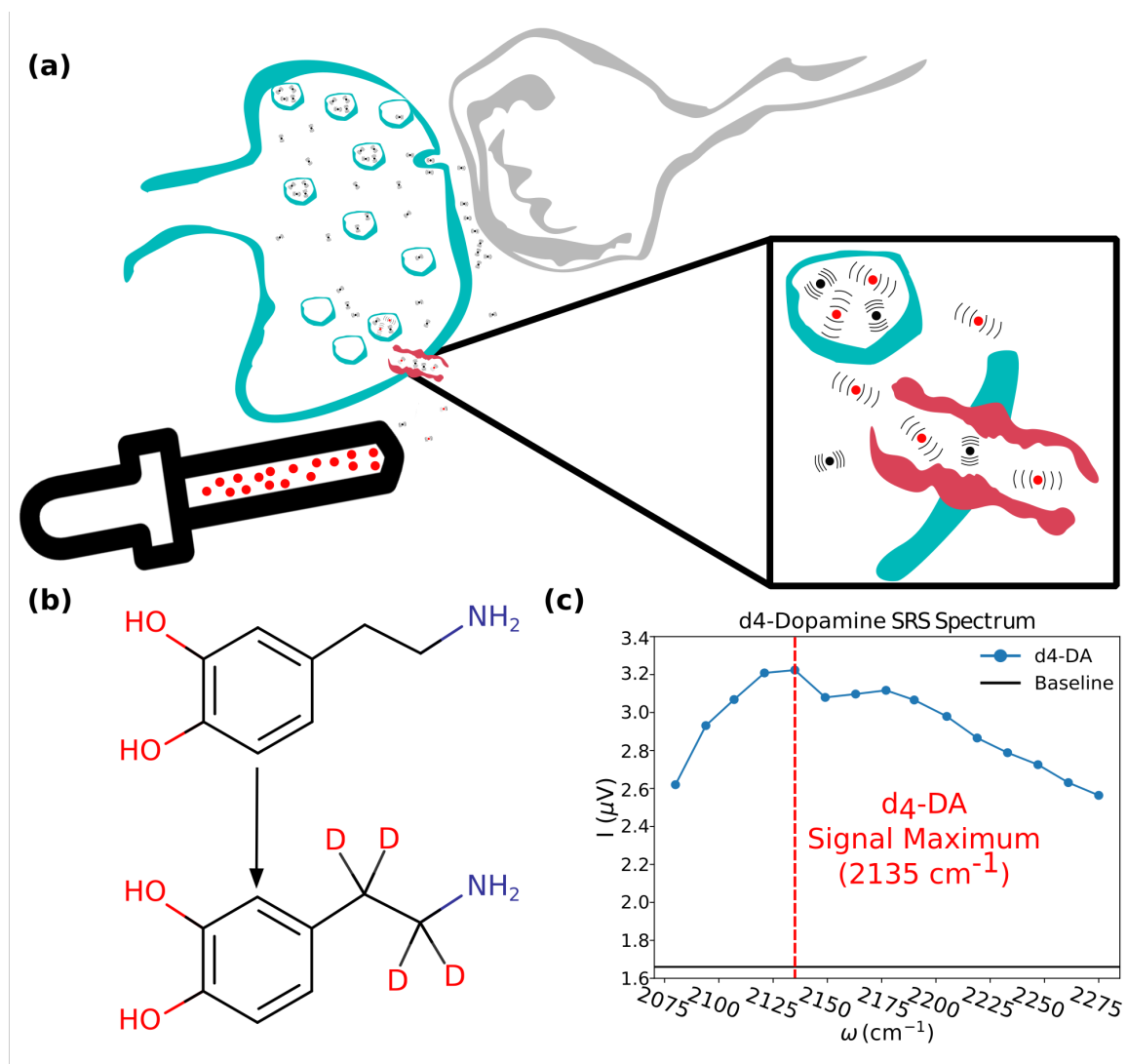
membranes, and is used to image exocytosed volume transmission of dopamine (Sun et al., 2018). Patriarchi et al concurrently produced a suite of genetic encoded sensors, operating on similar principles, called dLight (Patriarchi et al., 2018). Both probes require extensive protein engineering, which is not easily translated to other neurotransmitters or non-model organisms, to-date lacks single release-site sensitivity, and can only measure neurotransmitters post-exocytosis. Another class of sensors based on near-infrared fluorescent nanoparticles (Kruss et al., 2017) can be localized to the extracellular space in the brain and produce fluorescence changes upon neurotransmitter binding, allowing imaging of synaptic-scale neurotransmitter volume transmission in acute brain slices (Beyene et al., 2019). These sensors have recently been used to image dopamine release from somatodendritic processes and to characterize dopamine release from individual synapses (Bulumulla et al., 2022). All aforementioned fluorescence-based probes face similar research and development challenges, requiring new approaches to their design, development, and deployment with each new neurotransmitter target. Furthermore, their use is limited to the measurement of extracellular analytes, making it difficult to study neurotransmitters in the intracellular environment.

To address the challenge of intracellular neurotransmitter imaging, direct chemical modification of neurotransmitters has been used to produce visible fluorescence, allowing tracking of the molecule itself. False neurotransmitters analogs have been under development for many years as artificial alternatives to their endogenous counterparts. To this end, fluorescent false neurotransmitters (FFN) enable intracellular tracking and subsequent release of the neuromodulators dopamine (Pereira et al., 2016; Rodriguez et al., 2013) and serotonin (Henke et al., 2018). However, relatively large structural changes to the native neurotransmitter are necessary to produce a visibly fluorescent analog, requiring extensive validation and complex syntheses. Furthermore, it remains unclear to what extent the structural modifications present in FFNs affects their biodistribution and function. Several investigations have therefore used the natural autofluorescence of certain neurotransmitters, such as dopamine and serotonin (Balaji et al., 2005; Maity and Maiti, 2018; Sarkar et al., 2014). While this approach provides a direct readout of the compound, the use of UV fluorescence limits penetration depth and precludes application to other non-intrinsically autofluorescent neurotransmitters.

Fluorescence, however, is not the only physical property that can be used to gain chemical specificity. Molecular vibrations are specific to the types of atoms and bonds in a molecule and can be probed through several different optical interactions, many of which are well suited to implementation as imaging techniques. The most common probes of molecular vibrations are infrared (IR) and Raman spectroscopy, although many others exist including surface-enhanced Raman scattering (SERS), photoacoustic imaging, and nonlinear techniques such as coherent anti-Stokes Raman scattering (CARS), and stimulated Raman scattering (SRS). However, vibrational imaging faces its own difficulty in isolating the vibrational signature of a single molecule of interest in situ in a biological system. A typical vibrational spectrum of a cell contains a substantial amount of information about cellular composition but, even with high spectral resolution, is too complex to deconvolve into individual molecular contributions. To date, only a single neurotransmitter, acetylcholine, has been imaged using vibrational microscopy in the frog neuromuscular junction (Fu et al., 2017). The lack of subsequent development of vibrational probes for neuroscience illustrates the challenge in specifically imaging a single type of molecule in a complex cellular context.

Nonetheless, in the animal systems of interest, a gap appears in the vibrational spectrum from 1800 - 2800  $cm^{-1}$ , often called the cell-silent window. This corresponds to the frequency range

of alkyne and nitrile vibrations, functional groups that are largely absent from biomolecules in animals. Alkyne labeling has therefore been used to study a number of processes including imaging of newly synthesized DNA by incorporation of 5-ethynyl-deoxyuridine (EdU) (Wei et al., 2014; Yamakoshi et al., 2011), to study mitochondria (Yamakoshi et al., 2015), membrane proteins (Hu et al., 2018), and drug uptake (Wei et al., 2014; Zhao et al., 2017). Such a modification was recently proposed to study dopamine; however, imaging based on the alkyne vibration has not yet been demonstrated (Nuriya et al., 2021). An alternative strategy to the incorporation of alkyne functional groups is to use isotope labeling, as the frequency of a molecular vibration is inversely proportional to the reduced mass of the atoms involved. Increasing the mass of the involved atoms will therefore reduce the vibrational frequency, for example by employing one of two common isotopic substitutions used in mass spectrometry and nuclear magnetic resonance (NMR):  $^1\text{H}$  to  $^2\text{H}$  (D) or  $^{12}\text{C}$  to  $^{13}\text{C}$ . These substitutions move the C-H vibrational modes that are normally at 2800-3000  $\text{cm}^{-1}$  into the cell-silent window. Isotope labeled compounds have been used in conjugation with vibrational imaging to study lipid and cholesterol dynamics (Alfonso-García et al., 2015), proteome degradation, and Huntington protein aggregation (Shen et al., 2014), in addition to being used as a general readout of metabolic activity (Shi et al., 2018). Furthermore, due to the application of isotope labeling in mass spectrometry and NMR, many neurotransmitters are already commercially available in isotope-labeled form as analytical standards. This commercial availability provides a rapid source of neurotransmitters with vibrational signatures that can be easily separated from the cellular background, obviating the need for special syntheses, or genetic engineering for their implementation in neuroscience research. We hypothesized that when introduced into neurobiological systems or cell cultures, these compounds would be internalized by endogenous neurotransmitter transporters (figure 4.1a). Once internalized, vesicular populations of the neurotransmitter of interest can then be monitored under native conditions using vibrational microscopy.



**Figure 4.1:** Overview of isotopologue imaging with vibrational microscopy.

(a) Deuterated neurotransmitters are incubated with the sample of interest. Endogenous neurotransmitter transporters uptake the deuterated isotopologues, facilitating their packaging into vesicles and allowing their visualization using vibrational microscopy techniques such as stimulated Raman scattering (SRS). (b) The commercially available, four deuterium substituted dopamine isotopologue used for tracking dopamine uptake and release. (c) The SRS spectrum of the dopamine isotopologue.

In this report, we demonstrate the application of SRS microscopy with this labeling strategy to image two such commercially-available deuterated neurotransmitters, d<sub>4</sub>-dopamine (d<sub>4</sub>-DA) and d<sub>6</sub>- $\gamma$ -aminobutyric acid (d<sub>6</sub>-GABA), in cultured PC12 cells and primary hippocampal neurons, respectively. SRS is a coherent 3rd order nonlinear process which makes use of two pulsed lasers, and is implemented in a point-scanning configuration for microscopy, like with multiphoton excited fluorescence microscopy. The two laser frequencies are chosen such that the difference in their frequency is resonant with a vibrational mode of interest. As such, for any single image only a single spectral channel is acquired, in contrast to confocal spontaneous Raman microscopy.



However, through coherent driving of the vibration, signal enhancements of up to 108 over spontaneous Raman scattering can be obtained (33, 34). We show that leveraging this signal enhancement allows for visualization of  $d_4$ -DA and  $d_6$ -GABA neurotransmitter isotopologues at speeds comparable to confocal fluorescence microscopy, with probe structures that are functionally identical to their native counterparts.

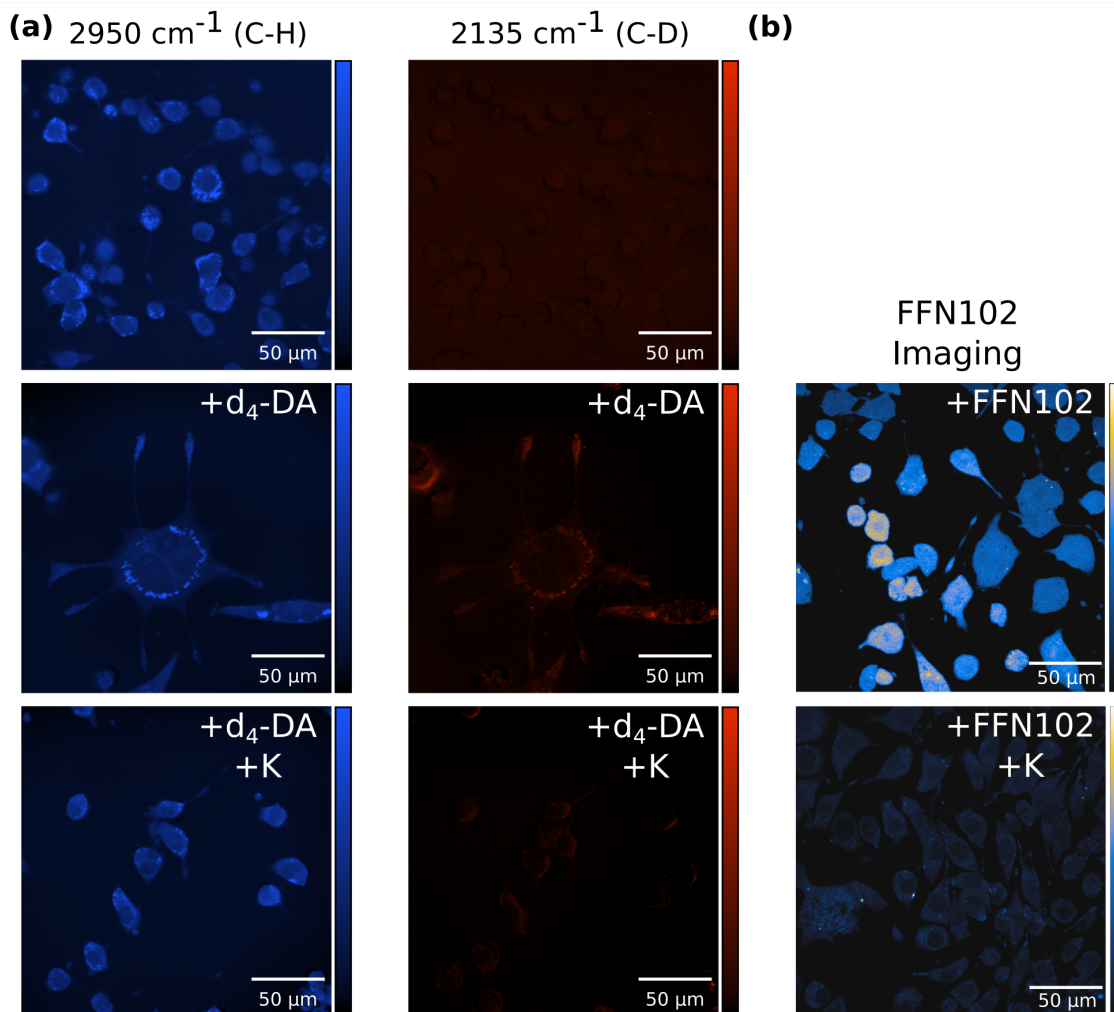
## 4.3 Results

### 4.3.1 SRS microscopy allows observation of uptake and release of $d_4$ -DA

As a proof of principle, we investigated the capability to image the uptake and intracellular distribution of the deuterated dopamine neurotransmitter,  $d_4$ -DA, in cultured PC12 cells. The PC12 cell line, derived from rat chromaffin cells in the 1970s, is a popular in vitro culture system for studying catecholamines and neurite development (Westerink and Ewing, 2008). Figure 4.1b shows the hydrogen-to-deuterium substitutions made for this particular isotopologue, while its corresponding SRS spectrum acquired in the cell-silent window is displayed in panel c. We first tested the ability to detect uptake of the labeled compound  $d_4$ -DA, by adding it to a final concentration of  $50 \mu\text{M}$  to the cell culture medium. The PC12 cells were incubated for one hour, after which they were washed and then imaged in PBS.

To verify uptake, images were acquired at  $2135 \text{ cm}^{-1}$ , the maximum of the  $d_4$ -DA SRS spectrum in the cell-silent window (figure 4.1c), and compared to the images of cells which were not incubated with  $d_4$ -DA. To provide a reference for the location and structure of the entire cell, images were also acquired at  $2950 \text{ cm}^{-1}$ , corresponding to  $\text{CH}_3$  stretches, a signal largely dominated by proteins.  $d_4$ -DA internalization was observed for the  $d_4$ -DA-incubated cells, largely localized to the cell periphery, and in punctate structures, with clear nuclear exclusion (figure 4.2a, middle row; figure 4.9). Under identical imaging and culture conditions, unincubated cells displayed no such signal, consistent with detection of a signal specific to the  $d_4$ -DA neurotransmitter (figure 4.2a, top row; Figure 4.10). We observe a significant difference between the average intracellular  $d_4$ -DA signal intensity in incubated vs unincubated cells (figure 4.3a;  $p < 0.001$ ,  $\Delta \bar{i} 15$ ).

The observed subcellular distribution of  $d_4$ -DA reveals signal along the cell periphery and in distinct puncta as predicted, suggesting a biologically-relevant internalization mechanism, with the isotopologue confined to intracellular vesicles. Comparatively little to no signal is observed outside these bright puncta, consistent with a much lower concentration of neurotransmitter present in the cytoplasm, estimated to be  $0.5 - 5 \mu\text{M}$  (Jones et al., 1999; Olefirowicz and Ewing, 1990), compared to  $110 - 190 \text{ mM}$  in the vesicles (Chen et al., 1994; Wightman et al., 1991). Furthermore, we observe the expected lack of  $d_4$ -DA internalization into the nucleus. To test whether this  $d_4$ -DA signal did arise from an active vesicular population of  $d_4$ -DA, we incubated PC12 cells with  $50 \mu\text{M}$   $d_4$ -DA and next stimulated  $d_4$ -DA release with  $50 \text{ mM}$  potassium, and imaged cells two minutes post-stimulation. In line with the expected response to potassium stimulation, there was an observable decrease in the measured  $d_4$ -DA signal for the population of cells imaged post stimulation (figure 4.2a, bottom row; figure 4.11), which is significant compared to the unstimulated cells (figure 4.2a;  $p < 0.001$ ,  $\Delta \bar{i} 0.35$ ). This 22% decrease is commensurate with prior reports that quantal release of dopamine in PC12 cells is fractional.

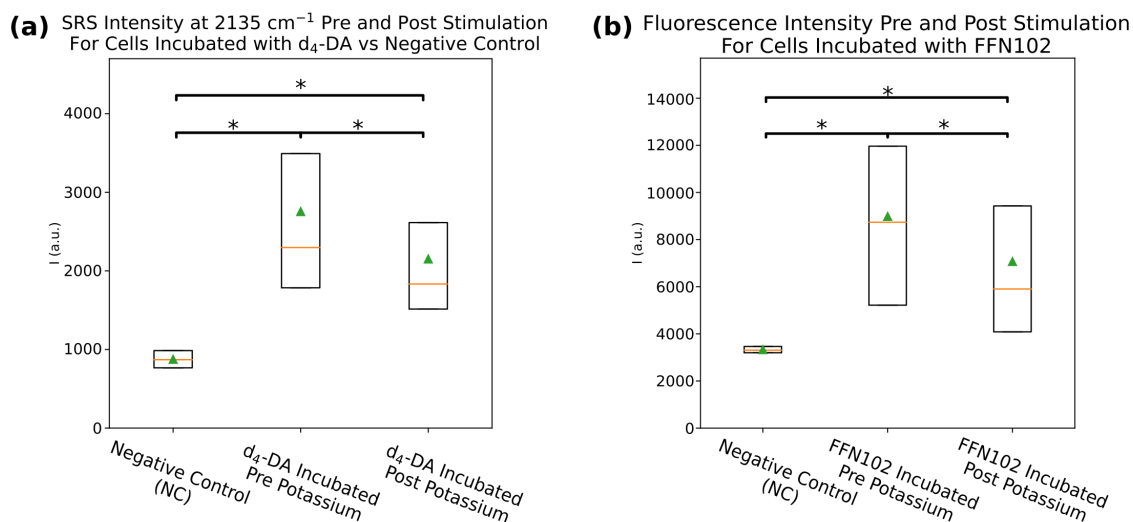


**Figure 4.2:** SRS and two-photon fluorescence imaging of a deuterated dopamine isotopologue and a false fluorescent analogue in PC12 cells.

(a) SRS images of PC12 cells without incubation with the deuterated dopamine isotopologue (top), with incubation with the dopamine isotopologue (middle), and with incubation with the dopamine isotopologue, but two minutes post stimulation to release neurotransmitter with 50 mM potassium (bottom). The left column displays images at  $2950\text{ cm}^{-1}$ , the C-H stretching frequency range, broadly corresponding to protein signal and allowing cell localization. The right column displays images at  $2135\text{ cm}^{-1}$ , the C-D stretching frequency range, corresponding to the signal originating from the deuterated dopamine compound. (b) Two-photon excited fluorescence images of PC12 cells incubated with a false fluorescent neurotransmitter, FFN102 (top), and cells incubated with FFN102 imaged two minutes post stimulation to release neurotransmitter with 50 mM potassium.

We next compared the performance of a dopamine FFN, FFN102, in the above detailed potassium stimulation experiments. PC12 cells were incubated with  $10\text{ }\mu\text{M}$  FFN102. A strong signal was visible within cells incubated with FFN102 when imaged by two-photon excited fluorescence (TPEF) microscopy. (figure 4.2b, middle row). Notably, there was stronger FFN102 signal from the entire cell and unlike  $\text{d}_4\text{-DA}$ -labeled cells, some FFN signal was observed in the nucleus

suggesting non-specific internalization mechanisms of the dopamine FFN analogue. FFN102-incubated cells imaged post potassium stimulation also displayed a lower overall signal relative to non-potassium stimulated controls (figure 4.2b, bottom row). This 21% decrease for FFN-incubated cells post stimulation was observed to be significant over a large sample of cells (figure 4.3b;  $p < 0.001$ ,  $\Delta > 0.48$ ) and is similar to the 22% signal change observed when cells are studied with  $d_4$ -DA. The two distributions for  $d_4$ -DA and FFN102, in fact, show very similar trends, and suggest  $d_4$ -DA can report on relevant dopamine uptake and release in cultured PC12 cells.



**Figure 4.3:** Cell-averaged SRS or fluorescence intensity distributions for cells incubated with deuterated dopamine analogue, or FFN102.

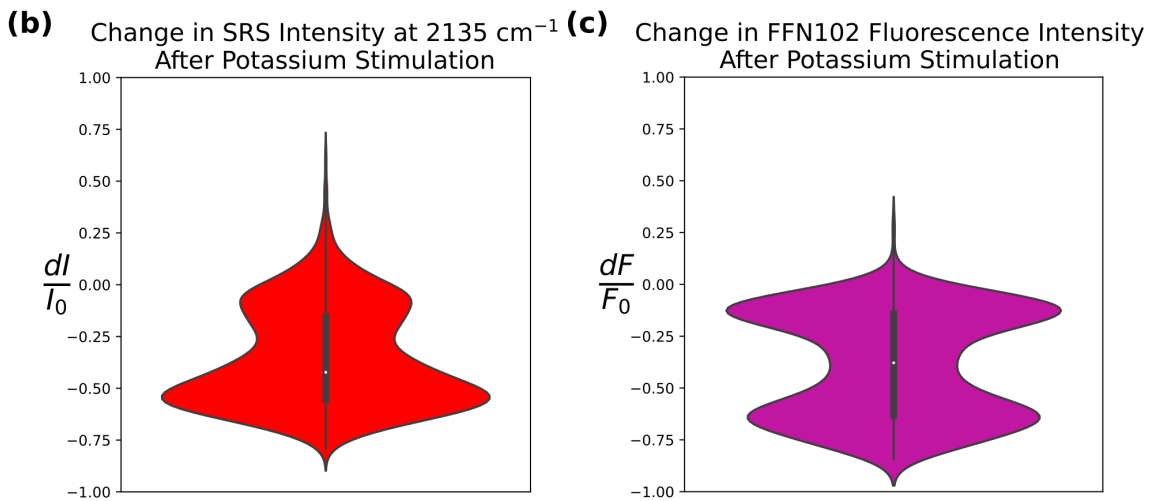
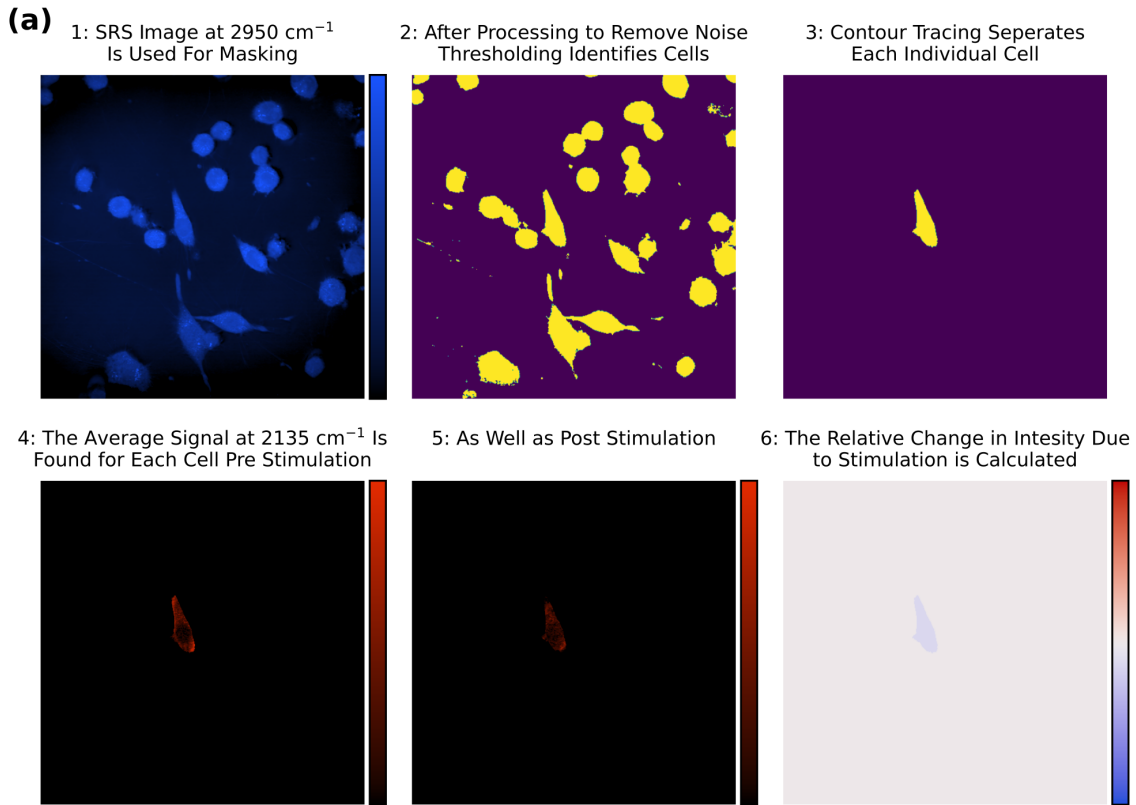
**(a)** The cell-averaged SRS intensity for cells that were not incubated with deuterated dopamine (left column), cells that were incubated with deuterated dopamine (middle column), and cells that were incubated with deuterated dopamine, stimulated to release neurotransmitter with 50 mM potassium, and imaged two minutes later (right column). \*  $p < 0.001$ ,  $\Delta > 15$  (incubated vs negative control),  $\Delta > 0.35$  (incubated vs incubated post stimulation).  $N(\text{negative control}) = 1273$ ,  $N(\text{Incubated, pre-stimulation}) = 1152$ ,  $N(\text{Incubated, post-stimulation}) = 1145$ . **(b)** The cell-averaged two-photon excited fluorescence intensity for cells that were not incubated with FFN102 (left column), cells that were incubated with FFN102 (middle column), and cells that were incubated with FFN102, stimulated to release neurotransmitter with 50 mM potassium, and imaged two minutes later (right column). \* $p < 0.001$ ,  $\Delta > 0.48$  (incubated vs incubated post-stimulation).  $N(\text{Negative control}) = 1802$ ,  $N(\text{Incubated, pre-stimulation}) = 1904$ ,  $N(\text{Incubated, post-stimulation}) = 2920$ .

### 4.3.2 PC12 cells display cell-to-cell heterogeneity of $d_4$ -DA uptake and release

We next sought to measure stimulated release in individual cells that were incubated with  $d_4$ -DA, or FFN102. While population level measurements have relevance for several important biological questions, they often obscure cell-to-cell heterogeneity. It is, therefore, frequently of interest to

take repeat measurements on the same individual cells. Single-cell measurements are of particular relevance for neurons, where measuring the effect of stimulation and the relevant kinetics of neurotransmitter release are of fundamental importance to understand neuro-communication. Furthermore, in PC12 cells, it is known that multiple vesicular populations (Westerink et al., 2000) can exist heterogeneously across individual cells, producing cell-to-cell variability in dopamine concentration that is often not apparent when making bulk population level measurements (Westerink and Ewing, 2008).

To measure  $d_4$ -DA concentration within the same PC12 cells pre- and post-stimulation, we built a simple microfluidic flow cell, with inlets for the injection of stimulating potassium solution. Cells were grown inside these devices, composed of coverslips on the top and bottom, and PDMS around the sides. Using these flow cells, we first incubated PC12 cells with  $50 \mu\text{M}$   $d_4$ -DA, as in prior experiments. Prior to imaging, the cells were washed and imaging was performed in clean culture medium. We first acquired images at  $2950 \text{ cm}^{-1}$ , to localize cell boundaries, and serve as a mask to identify and separate cells computationally during image analysis. We then acquired images at  $2135 \text{ cm}^{-1}$  to image intracellular  $d_4$ -DA before and after stimulation by exposure to a  $50 \text{ mM}$  potassium solution (Figure 4a). We next quantified the relative change in  $d_4$ -DA intensity, i.e. the change in  $d_4$ -DA intensity normalized to the initial intensity, in response to potassium stimulation on a single-cell basis. This analysis revealed a bimodal distribution for PC12 cells incubated with  $d_4$ -DA, with one cell population exhibiting a strong 64% release of  $d_4$ -DA in response to potassium stimulation, and another population exhibiting a weaker 15% release of  $d_4$ -DA (figure 4.4b). This bimodal distribution is similar to the distribution observed when the experiment is repeated for PC12 cells incubated with FFN102 (figure 4.4c). We next examined whether a cell's initial  $d_4$ -DA intensity, putatively correlated to the intracellular dopamine concentration, is correlated to its subsequent release magnitude by comparing the relative change in  $d_4$ -DA intensity due to potassium stimulation as a function of the initial pre-stimulation  $d_4$ -DA intensity. We observe a general trend where cells showing higher  $d_4$ -DA intracellular signals show greater release – i.e. higher change in  $d_4$ -DA intensity in response to potassium stimulation in both distinct populations. Interestingly, while we also observe these two populations for FFN102-incubated cells, a third, high initial intensity, but weakly-releasing, cell population is also present in FFN102-incubated cells. (figure 4.8) This result may potentially point to an additional, non-specific, internalization mechanism of the FFN102 FFN that is not active for  $d_4$ -DA uptake.

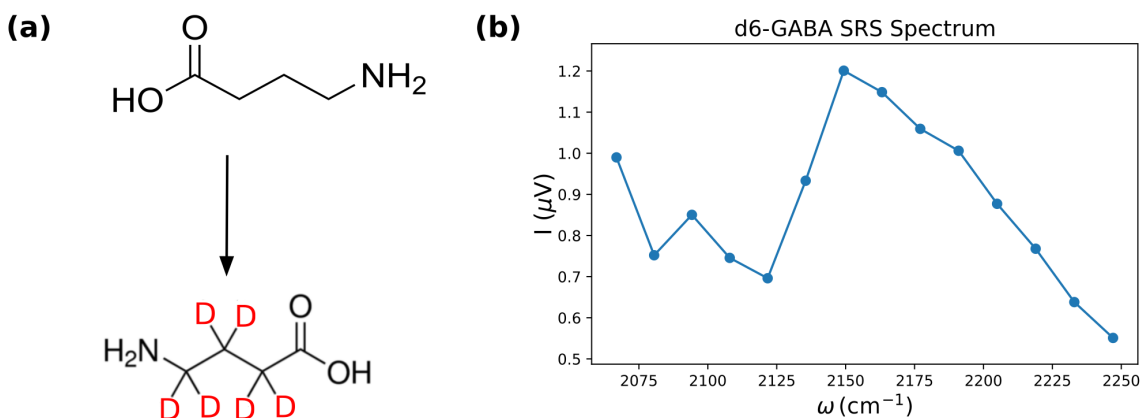


**Figure 4.4:** SRS and two-photon excited fluorescence imaging on the same single cells pre- and post-potassium stimulation.

**(a)** Workflow for calculation of relative SRS intensity changes due to potassium stimulation. 1. A single image is acquired in the  $2950\text{ cm}^{-1}$  (C-H) channel, broadly corresponding to protein distribution, which is used as a mask. 2. A mask is created from this image using adaptive thresholding and binarization. 3. Contour tracing allows single cells to be retrieved from the mask. 4. A single image is acquired in the  $2135\text{ cm}^{-1}$  (C-D) channel, corresponding to the deuterated dopamine signal. The average signal is determined for each masked cell. 5. The cells are stimulated to release neurotransmitter with 50 mM potassium solution, and a second image in the  $2135\text{ cm}^{-1}$  channel is acquired two minutes later. 6. The relative change in SRS signal is calculated for each masked cell. **(b)** The distribution of relative SRS intensity change in the  $2135\text{ cm}^{-1}$  channel on a cell-averaged level due to stimulation with 50 mM potassium. **(c)** The relative change in FFN102 fluorescence intensity due to stimulation with 50 mM potassium.

Using these flow cells, we first incubated PC12 cells with  $50\text{ }\mu\text{M}$   $\text{d}_4\text{-DA}$ , as previously. Prior to imaging the cells were again rinsed; however, imaging was performed in clean culture medium as opposed to PBS. We next acquired an image at  $2950\text{ cm}^{-1}$ , to localize cell boundaries, and which served as a mask to identify and separate cells computationally. Subsequently an image was acquired in the  $2135\text{ cm}^{-1}$  to quantify  $\text{d}_4\text{-DA}$  concentration prior to stimulation. Next the cells were stimulated to release the isotopologue by exposure to a 50 mM potassium solution, before a second image was acquired at  $2135\text{ cm}^{-1}$  to quantify  $\text{d}_4\text{-DA}$  concentration post stimulation (figure 4.4a). We next quantified the relative change in  $\text{d}_4\text{-DA}$  intensity, i.e. the change in  $\text{d}_4\text{-DA}$  intensity normalized to the initial intensity, in response to potassium stimulation. This analysis revealed a bimodal distribution for PC12 cells incubated with  $\text{d}_4\text{-DA}$ , with one population exhibiting a strong 64% release of  $\text{d}_4\text{-DA}$  in response to potassium stimulation, and another population exhibiting a weaker 15% release of  $\text{d}_4\text{-DA}$  (figure 4.4b). This bimodal distribution is similar to the distribution observed when the experiment is repeated for PC12 cells incubated with FFN102. We next compared the relative change in intensity due to potassium stimulation versus the initial pre-stimulation intensity in  $\text{d}_4\text{-DA}$  incubated cells (figure 4.8). We observed a general trend where the greater the initial pre-stimulation intensity, the greater the relative change in intensity in response to potassium stimulation, which was split roughly into two populations. Interestingly, while we observe these two populations for FFN102-incubated cells, a third, high initial intensity, but weakly-responsive, population is also present in FFN102-incubated cell. This result may potentially point to an additional, non-specific, internalization mechanism of the FFN, that is not active for  $\text{d}_4\text{-DA}$  uptake.

### 4.3.3 d<sub>6</sub>-GABA uptake is also observable with SRS microscopy



**Figure 4.5:** A deuterated GABA isotopologue and its spectrum.

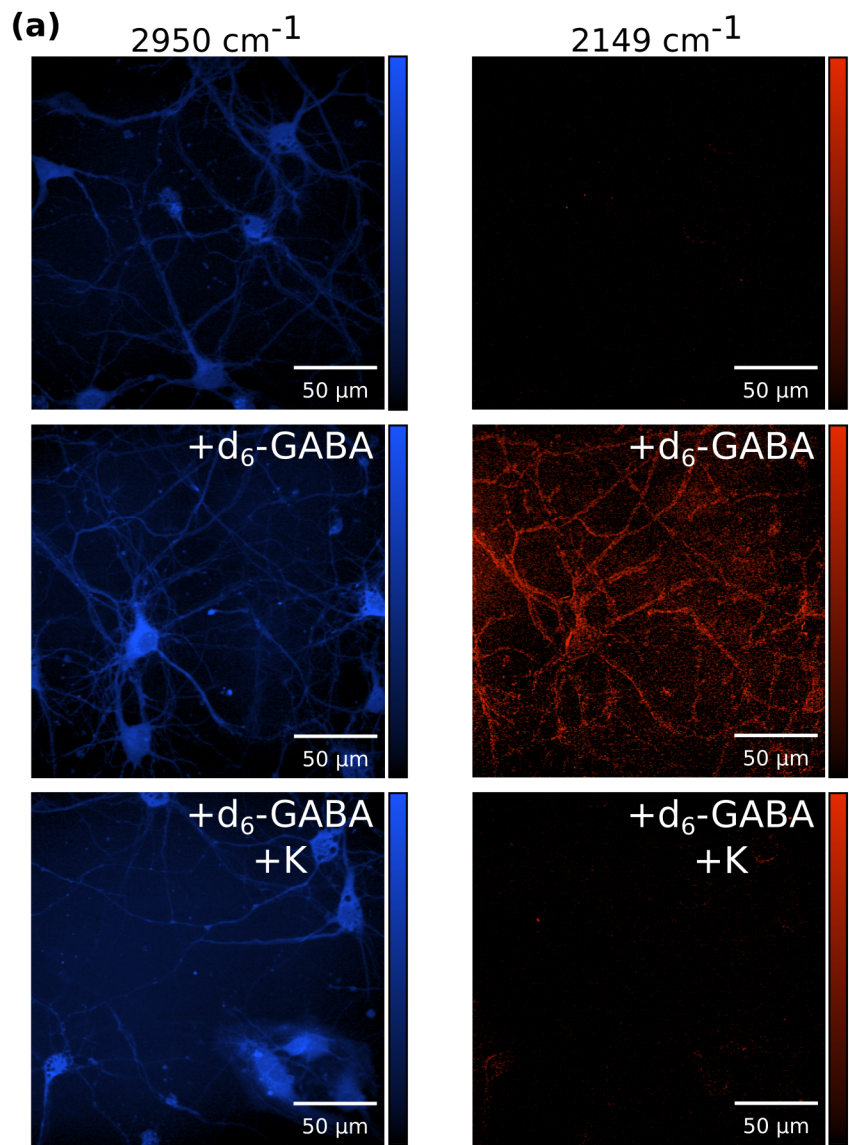
(a) The H to D substitutions for the commercially available GABA isotopologue used in this study. (b) The corresponding SRS spectrum for the deuterated GABA isotopologue in the cell-silent window.

We next assessed whether the strategy of neurotransmitter deuteration combined with SRS microscopy could be applied more broadly to different commercially-available deuterated neurotransmitters. GABA plays a crucial role in the nervous system as the main inhibitory neurotransmitter and is consequently centrally implicated in many neurochemical processes throughout the brain. Furthermore, while dopamine as a neuromodulator exhibits volume transmission-based activity, GABA is thought not to, representing a class of neurotransmitter putatively confined to the neuronal synapse. While recent advances have been made for imaging GABA with exogenous fluorophores in FRET-based biosensors (Lecat-Guillet et al., 2017; Masharina et al., 2012), and a genetically-encoded GABA biosensor (Marvin et al., 2019), comparatively few tools have been developed for GABA imaging. To test the capability of SRS microscopy to image the uptake and stimulated release of a deuterated GABA isotopologue we used dissociated fetal hippocampal neurons from rat.

The isotopologue used, d<sub>6</sub>-GABA, was fully substituted for deuterium at all available carbon bonded hydrogens (figure 4.5a). Fetal hippocampal neurons, dissociated from day 19 Sprague Dawley rat embryos, were incubated with 50  $\mu\text{M}$  d<sub>6</sub>-GABA prior to imaging. d<sub>6</sub>-GABA has a different spectral signature than d<sub>4</sub>-DA in the cell-silent window, so images were acquired at the d<sub>6</sub>-GABA maximum of 2149  $\text{cm}^{-1}$  (figure 4.5b) for its visualization. Images were again acquired in the 2950  $\text{cm}^{-1}$  channel to provide contrast of the cell body.

Hippocampal neurons displayed a strong signal at 2149  $\text{cm}^{-1}$  when incubated with the deuterated neurotransmitter (figure 4.6, middle row). In contrast to d<sub>4</sub>-DA signal which was largely confined to cellular puncta, d<sub>6</sub>-GABA signal was visible across the entire cell. This diffuse d<sub>6</sub>-GABA biodistribution may be explained by the higher cytoplasmic concentration of GABA, compared to dopamine, the former of which is maintained in the 5 mM range (Rothman et al., 1993; Wu et al., 2007). As expected, d<sub>6</sub>-GABA signal was absent from images acquired from hippocampal neurons not incubated with d<sub>6</sub>-GABA but otherwise cultured under identical conditions (figure 4.6,

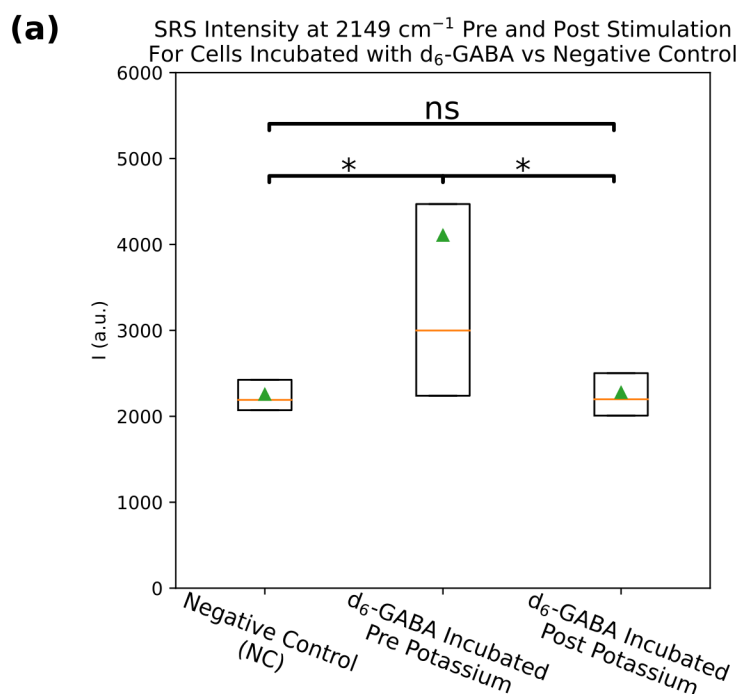
top row). Across many neurons, we observe a significant separation between the average signal intensity in  $d_6$ -GABA incubated vs unincubated cells (figure 4.7;  $p < 0.001$ ,  $\zeta = 5.58$ ), confirming uptake of  $d_6$ -GABA in hippocampal neurons and confirming the selectivity of SRS for imaging  $d_6$ -GABA at  $2149\text{ cm}^{-1}$ . In contrast to the case of PC12 cells, however, a population of cells do not uptake  $d_6$ -GABA, reflected in an overlap between the incubated and unincubated distributions. This large variation in  $d_6$ -GABA uptake by hippocampal neurons is likely due to the many neuronal subtypes that exist in the hippocampus where glutamatergic neurons may be more prevalent than GABAergic neurons (Zhu et al., 2017). Evidence also suggests that glutamine and GABA may coexist and may be co-released in certain neurons, which could affect relative GABA uptake (Shabel et al., 2014; Zhu et al., 2017). For non-GABAergic, or neurons containing a mixture of multiple neurotransmitters, no or decreased uptake of  $d_6$ -GABA is expected.





**Figure 4.6:** SRS imaging of a deuterated GABA isotopologue primary hippocampal neurons. (a) SRS images of primary hippocampal neurons without incubation with the deuterated GABA isotopologue (top), with incubation with the GABA isotopologue (middle), and with incubation with the GABA isotopologue, but two minutes post stimulation to release neurotransmitter with 50 mM potassium (bottom). The left column displays images at  $2950\text{ cm}^{-1}$ , the C-H stretching frequency range, broadly corresponding to protein signal and allowing cell localization. The right column displays images at  $2149\text{ cm}^{-1}$ , the C-D stretching frequency range, corresponding to the signal originating from the deuterated GABA compound.

To examine whether the signal measured at  $2149\text{ cm}^{-1}$  did arise from active internalization of d6-GABA, we repeated the stimulated release experiments by incubating neurons with  $50\text{ }\mu\text{M}$  d6-GABA for one hour then stimulating with 50 mM potassium for 2 minutes. Images were acquired in PBS after first washing away the potassium and d6-GABA containing medium. Neurons imaged after potassium stimulation showed little to no observable signal at  $2149\text{ cm}^{-1}$  (Figure 5c, bottom row). This 55% decrease in d-6-GABA signal following potassium stimulation was found to be significant across many cells (figure 4.7;  $p < 0.001$ ,  $\Delta > 0.65$ ). Furthermore, we compared d6-GABA signal in post-potassium stimulated cells, and neurons not incubated with d6-GABA. We found no significant difference in signal between d6-GABA neurons imaged post potassium stimulation, and unincubated neurons, suggesting successful and near complete release of internalized d6-GABA. Together these data suggest the ability to visualize uptake and release of a GABA isotopologue within cultured primary neurons. To the best of our knowledge, these results represent the first optical micrographs of the neurotransmitter GABA itself within a neurobiological system



**Figure 4.7:** Cell-averaged SRS intensity distributions for cells incubated with deuterated GABA isotopologue.

(a) The cell-averaged SRS intensity for cells that were not incubated with deuterated GABA (left column), cells that were incubated with deuterated GABA (middle column), and cells that were incubated with deuterated GABA, stimulated to release neurotransmitter with 50 mM potassium, and imaged two minutes later (right column). \*  $p < 0.001$ ,  $\Delta > 5.58$  (incubated vs negative control),  $\Delta > 0.65$  (incubated vs incubated post stimulation).  $N(\text{negative control}) = 1840$ ,  $N(\text{Incubated, pre-stimulation}) = 2110$ ,  $N(\text{Incubated, post-stimulation}) = 2537$ .

## 4.4 Discussion

Isotopes have long a history of use in neuroscience. Some of the first techniques developed to measure concentrations of neurotransmitters were based on radioactive isotopes (Bowdler et al., 1983; Lieberman et al., 1969). Positron emission tomography (PET) continues to make frequent use of radiolabels to study the proteins of the nervous system (Chen et al., 2020; Stehouwer and Goodman, 2009). Powerful mass spectrometry imaging methods use deuterated analogues as calibration standards for quantitative spatial mapping of neurotransmitters (Fernandes et al., 2016; Shariatgorji et al., 2014; Zhu et al., 2011). In this report, we extend the use of isotopes in neuroscience and demonstrate that by combining the use of commercially-available isotopologues with vibrational microscopy, we can image the biodistribution, and stimulated release, of deuterated dopamine and GABA isotopologues in cultured cells and primary neurons with sub-cellular resolution. Our results suggest that commercially-available deuterated neurotransmitters and neuromodulators could be applied to a broad range of studies of neurotransmitter uptake and release kinetics in neuronal cells, tissue, and potentially in living animals. Recent developments in the field of vibrational microscopy also provide opportunities to further enrich the information this technique can provide through integration of orthogonal methodologies that could simultaneously measure neurotransmitter release kinetics and electrical activity or to follow uptake and release of multiple neurotransmitters simultaneously.

SRS microscopy, for instance, has been shown to be sensitive to membrane potential (Liu et al., 2015), a property recently exploited to track neuron depolarization in response to action potentials (Lee et al., 2017). In combination with isotopologue imaging, SRS offers the potential for a single optical system to image both neurochemical and electrical information simultaneously. The narrow linewidths of vibrational transitions ( $10\text{s cm}^{-1}$ ), compared to electronic transitions ( $100\text{s cm}^{-1}$ ), raise the additional possibility of simultaneous observation of multiple neurotransmitters. Acquisition of hyperspectral image stacks, i.e. multiple images of the same field of view at many wavenumbers, allows for the spectral unmixing of the contribution of distinct neurotransmitter isotopologues. A multiplexed imaging strategy could help answer questions surrounding release and uptake of multiple neurotransmitters in single neurons, as in the case of GABA and glutamate, two neurotransmitters that canonically have opposing activity (Shabel et al., 2014; Zhu et al., 2017).

The current study has demonstrated isotopologue imaging in cells and primary neurons with transmission-based SRS imaging. To extend the use of neurotransmitter isotopologues to tissue samples, or even live animals, an epi-SRS microscope could be used, with a geometry analogous to more familiar fluorescence-based microscopes. The use of epi-SRS has previously been demonstrated to work in brain tissue (Bae et al., 2018; Ji et al., 2013). In fact, SRS imaging is rather

unique in that regard, as signal collection in the epi-direction increases with increasing sample thickness. For samples more than 2 mm thick, it has been found to be preferable to image in this back scattering geometry (Hill et al., 2020). Additionally, other vibrational imaging techniques are not subject to the same constraints. For instance, CARS, another coherent Raman technique, is frequently implemented in either a transmission or epi geometry, and confocal spontaneous Raman and SERS microscopy are almost exclusively implemented in epi geometries. As such, there are numerous opportunities to adopt existing Raman-based imaging techniques for isotopologue imaging across a broad range of neurobiological studies of neurotransmission.

Similarly, exclusively intracellular populations of dopamine and GABA were measured using our current imaging setup, since neither spontaneous Raman nor SRS microscopy are likely to achieve the sensitivity required to observe the lower concentrations of neurotransmitter present extracellularly. For neuromodulators like dopamine, for which the study of volume transmission is of interest, their extracellular concentrations can vary widely but are often present at < 20 nM concentrations extracellularly (Watson et al., 2006). Isotopologue imaging for intracellular studies combined with existing probes to image volume transmission would present a powerful opportunity to track neurotransmitter uptake, release, and volume transmission. As such, the current combination of neurotransmitter isotopologues and SRS microscopy is best implemented to study intracellular neurotransmitter populations and can serve as an orthogonal measure, complementing other tools available to study neuromodulator volume transmission. These include direct extracellular concentration sensors, such as those based on carbon nanotubes or genetically engineered proteins, or techniques like magnetic resonance spectroscopic imaging (MRSI), which is used to study neurotransmitter concentrations under a variety of paradigms. While powerful, MRSI is unable to distinguish intracellular and extracellular contributions to the signal (Spurny et al., 2020), due to lower spatiotemporal resolution, a void which our technique may help to fill when questions of mechanism or attribution arise. Additionally, signal enhancement techniques may also be used to further increase the sensitivity of vibrational microscopy. Recent work suggests that surface enhanced SRS (SESRS) can be used to image at the single molecule level as demonstrated with bacterial cells (Zong et al., 2019). Similar observations have been made for surface enhanced CARS (SECARS) (Koo et al., 2005; Steuwe et al., 2011; Zhang et al., 2014). Furthermore, a recent study (Zong et al., 2021) describes a widefield implementation of SECARS which could extend that level of sensitivity to the millisecond temporal resolution of more conventional widefield microscopies. While the use of plasmonic nanoparticles, or culturing on plasmonic nanostructure surfaces, introduces an additional level of complexity, the SERS field has frequently demonstrated its applicability to and utility in neuroimaging (Diaz et al., 2014; Nicolson et al., 2019), even in intraoperative/clinical settings (Jiang et al., 2019; Karabeber et al., 2014).

In this report we have presented a practical strategy to image neurotransmitters under native biological conditions using vibrational microscopy. This strategy generalizes to any class of neurotransmitter, many of which are already commercially available in isotope-labeled form, opening new avenues for fundamental investigations into the biology of neurotransmitters. The existing and long-standing use of deuterated neurochemicals in clinical practice further substantiates the biocompatibility of their use and motivates the use of our approach in non-model organisms and human brain tissues. Notably, many commercially-available isotopologues, such as GABA as demonstrated here, are neurotransmitters or neuromodulators that currently lack cellular-scale imaging tools. While this work represents the first demonstration of the use of isotopologues and vibrational microscopy in this context, it makes use of a technique, i.e., isotope labeling, which is

broadly familiar to both the microscopy and neuroscience communities. Furthermore, a growing demand for new tools to study the nervous system has led to increasing adoption of vibrational imaging methods in neuroscience, which can be easily extended to the study of neurotransmitter uptake and release as demonstrated herein. Taken together, our results suggest isotope labeling of neurotransmitters, combined with SRS microscopy, represents a strategy with the potential for wide adoption to further the understanding of neurotransmission in the brain.

## **4.5 Materials and Methods**

### **4.5.1 PC12 cell culture**

PC12 rat chromaffin cells (UC Berkeley cell culture facility) were continuously cultured in RPMI-1640 medium with L-glutamine, containing 10% goat serum, 10% FBS, and 1% penicillin/streptomycin. For d<sub>4</sub>-DA imaging, cells were passaged onto 25 mm diameter round coverslips. For FFN102 imaging, cells were passaged into 24 well-plates. Upon achieving 50-70% confluency on the coverslips, the media was exchanged to a serum deprived DMEM media, containing 0.5% FBS, 1% penicillin/streptomycin, and 100 ng/mL NGF- $\beta$ . The coverslips were imaged 5 days post media exchange. The cells were maintained in an incubator at 37 °C and 5% CO<sub>2</sub>.

### **4.5.2 Flow cell construction and PC12 cell culture on device**

SRS imaging of d<sub>4</sub>-DA in cells introduces a set of strict geometric constraints for sample placement on the microscope. This is due to the fact that the SRS signal is, generally, collected in a transmission geometry, requiring the sample be placed between short working-distance objective and condenser lenses. SRS microscopy can also be subject to non-specific signals arising from the optical Kerr effect in the form of cross-phase modulation (XPM). The most straight-forward way to reduce or eliminate this effect is to use a collecting lens with a numerical aperture equal to or larger than that of the objective lens (Berto et al., 2014), necessitating a 1.4 NA oil immersion condenser lens, for use with the 1.2 NA water immersion objective used in this study. In order to make measurements of d<sub>4</sub>-DA signal intensity on the same PC12 cells both before and after potassium stimulation, we constructed a simple flow cell. The flow cell was constructed using two 48 x 60 mm coverslips (Gold Seal, Thermo scientific). RTV 615 PDMS was first mixed in a 10:1 ratio of monomer to cross-linker. PEEK tubing (1/32"x0.010, IDEX Health and Science) was cut into lengths of 2" to serve as inlets into the device, and to control the height of the device. Two lengths of tubing were placed in the center of one short edge of the coverslip, while another two were placed along the entire length of each long edge. The PDMS mixture was then applied using a brush along the two long edges and the short edge with tube inlets. The PDMS was applied in a thickness equal to the height of the PEEK tubing extending up from the coverslip. The second coverslip was then placed on top, enclosing the device. The device was baked for 10 minutes at 70 °C, then flushed with ethanol and further sterilized under UV-C light for 2 minutes. The final height of the devices was 1.1 mm. PC12 cells were introduced to the device by extracting 2 mL of trypanized cells from the passaging culture using a syringe, sterilized as above using ethanol and UV-C light, and then affixing the syringe to one of the available inlets. Cells were allowed to grow

for 2 days before the medium was exchanged for the serum deprived medium. Cells were grown for a further 5 days before imaging, the same as those on coverslips. The devices were maintained within separate sterile petri dishes in the same incubator until imaging.

### **4.5.3 Primary hippocampal neuron dissociation and culture**

Neurons were dissociated from extracted hippocampi of day 19 embryos of Sprague Dawley rats, as previously described (Ortiz et al., 2021). Briefly, cells were dissociated for 15 minutes from excised hippocampi using a trypsinated buffer. The neurons were then plated onto 25 mm coverslips and incubated at 37 °C in MEM medium with 5% FBS, 2% B-27, 2% 1M dextrose and 1% GlutaMax. One day after dissociation, half the medium was exchanged for neurobasal medium with 2% B-27 supplement and 1% GlutaMax, to prevent proliferation of glial cells. Coverslips were then imaged using SRS microscopy two weeks post dissociation.

### **4.5.4 SRS microscope optical setup**

The optical setup has been described previously (Chapter 2; Kim et al., 2021). Briefly, the synchronized dual output from a commercial femtosecond laser/OPO system is used for the pump and Stokes beams. The Stokes beam was fixed at 1040 nm, while the pump was tunable. For imaging at 2950  $\text{cm}^{-1}$ , 2149  $\text{cm}^{-1}$ , and 2135  $\text{cm}^{-1}$ , the pump was tuned to 796 nm, 850 nm, and 851 nm respectively. The Stokes beam was intensity modulated at 10.28 MHz using a quarter waveplate (Thorlabs), a resonant electro-optic modulator (Thorlabs), driven by a function generator (Hewlett Packard) and power amplifier (Minicircuits), and a polarizer (Thorlabs). The pump and Stokes were combined colinearly using a 1000 nm short pass dichroic mirror. Temporal coincidence of the pulses was controlled using an optical delay line (Newport). A commercial scanning microscope (FV1200, Olympus) was used to raster scan the beams across the sample. A 60x 1.2 NA water immersion objective (Olympus) was used for excitation, and a 1.4 NA oil immersion condenser (Thorlabs) was used to collect the transmitted light. The Stokes beam was filtered by a 1000 nm shortpass filter, before the pump was detected on a photodiode (Hamamatsu) reverse biased at 61.425. The stimulated Raman loss signal was extracted from the photodiode signal using a lock-in amplifier (Zurich Instruments).

### **4.5.5 Deuterated neurotransmitter imaging**

PC12 cells or primary neurons cultured on coverslips were incubated for one hour at 37 °C with a final media concentration of 50  $\mu\text{M}$   $\text{d}_4$ -DA or  $\text{d}_6$ -GABA, respectively. After one hour the coverslips were washed three times with DPBS, before being placed on 25 mm diameter, low-profile 600  $\mu\text{m}$  culture dish (Grace coverwells), filled with DPBS. The edges of the culture dish contain an adherent which fixed the coverslips in place. PC12 cells were then imaged at 2950  $\text{cm}^{-1}$  and 2135  $\text{cm}^{-1}$  using the SRS microscope. The pump and Stokes power were both set to 20 mW. Images were acquired at a size of 512 x 512 pixels, using a 10  $\mu\text{s}$  pixel dwell time and a lock-in constant of 3  $\mu\text{s}$ . Neurons were imaged using a pump and Stokes power of 15 mW, and at 2950  $\text{cm}^{-1}$  and 2149  $\text{cm}^{-1}$ , but otherwise identical imaging conditions. For the negative control samples for both PC12 cells and primary neurons, the media was exchanged for fresh media not containing the deuterated compound. The cells were then left to rest for one hour before being

washed three times with DPBS and prepared identically to the cells incubated with the deuterated isotopologues, and imaged using the same conditions. For the post stimulation condition PC12 cells and primary neurons were incubated with 50  $\mu\text{M}$   $\text{d}_4$ -DA or  $\text{d}_6$ -GABA, respectively. After a one hour incubation, the media was exchanged for media containing 50 mM potassium. After two minutes, the coverslips were washed twice with DPBS, and prepared and imaged analogously to the previously described conditions.

PC12 cells which were cultured on the flow cell devices were incubated with 50  $\mu\text{M}$   $\text{d}_4$ -DA by first emptying the device of media through the open side and then introducing new media with the deuterated compound. After one hour the device was then washed twice with DPBS, before being filled with fresh media. In contrast to the PC12 cells cultured on coverslips, those cultured on the device were imaged in media not DPBS, in order to enable stimulated release of the  $\text{d}_4$ -DA while on the microscope. The devices were then imaged with the same settings as those used for the cells cultured on coverslips. One image was taken at 2950  $\text{cm}^{-1}$ , then a pre stimulation image at 2135  $\text{cm}^{-1}$ , an image at 2135  $\text{cm}^{-1}$  following influx of media not containing supplemental potassium, and finally a post-stimulation image 2 minutes after an influx of 50 mM potassium. This set of images was also repeated for PC12 cells cultured on devices which were not incubated with  $\text{d}_4$ -DA.

#### **4.5.6 Two-photon excited fluorescence imaging of FFN102**

For imaging of FFN fluorescence in PC12 cells, the cells were first incubated to a final media concentration of 10  $\mu\text{M}$  FFN102 for one hour at 37  $^\circ\text{C}$ . After one hour, samples were washed three times with DPBS. For imaging, the pump wavelength on the SRS microscope was tuned to 800 nm for two-photon excitation of FFN102, with the power set to 20 mW. Images were acquired in the epi-direction at a size of 512 x 512 pixels, using a 10  $\mu\text{s}$  pixel dwell time and non-descan PMT available on the Olympus microscope. For post potassium stimulation measurements, FFN102 was either imaged in cells 2 minutes after exposure to potassium in adjacent wells on the well plate that had not been previously imaged, or for measurements on the same cells, 2 minutes after exposure to potassium in a well that had been previously imaged.

#### **4.5.7 Image processing**

All statistics reported are calculated over the intensity distribution of the average signal in individual cells. Statistical significance was tested using a Welch's T-test assuming unequal variance. Effect sizes are displayed as Glass'  $\Delta$ . For calculations involving SRS images, all images in both the 2950  $\text{cm}^{-1}$  and 2135  $\text{cm}^{-1}$  or 2149  $\text{cm}^{-1}$  are pseudo-flatfield corrected using a replica of the respective image, convolved with Gaussian kernel of radius 200 pixels, as the flatfield. The 2950  $\text{cm}^{-1}$  channel images are then used for thresholding and segmentation. First a Gaussian blur of radius 5 pixels is applied, followed by a contrast limited adaptive histogram equilization with radius 15. Otsu's method is then applied to produce a binary thresholded image. A hierarchical contour tracing algorithm is then used to determine the outlines of the thresholded image. Internal contours, arising from organelles or low-intensity regions within a cell, are discarded, leaving the highest level outlines of the cell. Any contours with a total length of less than 100 pixels are also discarded. Each of the remaining contours is used to determine a mask over which the signal inside the contour is averaged in the corresponding 2135  $\text{cm}^{-1}$  or 2149  $\text{cm}^{-1}$  channel. Beyond

flatfield corrections the channels corresponding to deuterated compounds undergo no additional processing.

Fluorescence images of PC12 cells incubated with FFN102 are processed similarly; however, as the signal is strong enough to see cell boundaries in the fluorescence channel, no additional channel is needed to serve as the mask. Instead, after flatfield correction, a second copy of each image is used for masking, and undergoes the same processing as the SRS images acquired in the  $2950\text{ cm}^{-1}$ . The discovered contours are then used as masks on the unprocessed copy.

## 4.6 Acknowledgements

We would like to acknowledge Professor Evan Miller and his students Pavel Klier and Deshka Lynn Niell for providing primary hippocampal cells, and advice on maintenance of their culture conditions.

## 4.7 Funding

We acknowledge support of a Burroughs Wellcome Fund Career Award at the Scientific Interface (CASI) (to M.P.L.), a Dreyfus foundation award (to M.P.L.), the Philomathia foundation (to M.P.L.), an NIH MIRA award (to M.P.L.), an NIH R03 award (to M.P.L.), an NSF CAREER award (to M.P.L.), an NSF CBET award (to M.P.L.), an NSF CGEM award (to M.P.L.), a CZI imaging award (to M.P.L.), a Sloan Foundation Award (to M.P.L.), a USDA BBT EAGER award (to M.P.L.), a Moore Foundation Award (to M.P.L.), an NSF CAREER Award (to M.P.L.), a DOE office of Science grant with award number DE-SC0020366 (to M.P.L.), an NSF award with award number 1845623 (to A.S.). M.P.L. is a Chan Zuckerberg Biohub investigator, a Hellen Wills Neuroscience Institute Investigator, and an IGI Investigator. A.S. is a Chan Zuckerberg Biohub investigator, and a Pew Biomedical scholar. Work at the Molecular Foundry was supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Co-corresponding authors contributed equally and may modify the ordering of listed corresponding authors when disseminating this work.

## 4.8 References

Alfonso-García, A., Pfisterer, S.G., Riezman, H., Ikonen, E. and Potma, E.O. (2015), “D38-cholesterol as a Raman active probe for imaging intracellular cholesterol storage”, *Journal of Biomedical Optics*, Vol. 21 No. 6, p. 061003.

Arroyo, D.A., Kirkby, L.A. and Feller, M.B. (2016), “Retinal Waves Modulate an Intraretinal Circuit of Intrinsically Photosensitive Retinal Ganglion Cells”, *Journal of Neuroscience*, Vol. 36 No. 26, pp. 6892–6905.

Bae, K., Zheng, W., Lin, K., Lim, S.W., Chong, Y.K., Tang, C., King, N.K., et al. (2018), “Epi-Detected Hyperspectral Stimulated Raman Scattering Microscopy for Label-Free Molecular Subtyping of Glioblastomas”, *Analytical Chemistry*, Vol. 90 No. 17, pp. 10249–10255.

Balaji, J., Desai, R., Kaushalya, S.K., Eaton, M.J. and Maiti, S. (2005), “Quantitative measurement of serotonin synthesis and sequestration in individual live neuronal cells”, *Journal of Neurochemistry*, Vol. 95, pp. 1217–1226.

Berto, P., Andresen, E.R. and Rigneault, H. (2014), “Background-free stimulated raman spectroscopy and microscopy”, *Physical Review Letters*, Vol. 112 No. 5, pp. 1–5.

Betz, W.J., Mao, F. and Bewick, G.S. (1992), “Activity-dependent fluorescent staining and destaining of living vertebrate motor nerve terminals.”, *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, Vol. 12, pp. 363–375.

Beyene, A.G., Delevich, K., Travis, J., Donnell, D.B., Piekarski, D.J., Lin, W.C., Thomas, A.W., et al. (2019), “Imaging striatal dopamine release using a nongenetically encoded near infrared fluorescent catecholamine nanosensor”, *Science Advances*, Vol. 5.

Bowdler, J.M., Green, A.R., Minchin, M.C.W. and Nutt, D.J. (1983), “Regional GABA concentration and [3H]-diazepam binding in rat brain following repeated electroconvulsive shock”, *Journal of Neural Transmission*, Vol. 56 No. 1, pp. 3–12.

Bulumulla, C., Krasley, A.T., Walpita, D. and Beyene, A.G. (2022), “Visualizing Synaptic Dopamine Efflux with a 2D Nanofilm”, *BioRxiv*, p. 2022.01.19.476937.

Chen, J.Y., Wang, E.A., Cepeda, C. and Levine, M.S. (2013), “Dopamine imbalance in Huntington’s disease: A mechanism for the lack of behavioral flexibility”, *Frontiers in Neuroscience*, Vol. 7, pp. 1–14.

Chen, T.K., Luo, G. and Ewing, A.G. (1994), “Amperometric Monitoring of Stimulated Catecholamine Release from Rat Pheochromocytoma (PC12) Cells at the Zeptomole Level”, *Analytical Chemistry*, Vol. 66 No. 19, pp. 3031–3035.

Chen, X., Kudo, T., Lapa, C., Buck, A. and Higuchi, T. (2020), “Recent advances in radiotracers targeting norepinephrine transporter: structural development and radiolabeling improvements”, *Journal of Neural Transmission*, Springer Vienna, Vol. 127 No. 6, pp. 851–873.

Diaz, R.J., McVeigh, P.Z., O’Reilly, M.A., Burrell, K., Bebenek, M., Smith, C., Etame, A.B., et al. (2014), “Focused ultrasound delivery of Raman nanoparticles across the blood-brain barrier: Potential for targeting experimental brain tumors”, *Nanomedicine: Nanotechnology, Biology, and Medicine*, Elsevier Inc., Vol. 10 No. 5, pp. e1075–e1087.

Fernandes, A.M.A.P., Vendramini, P.H., Galaverna, R., Schwab, N. V., Alberici, L.C., Augusti, R., Castilho, R.F., et al. (2016), “Direct Visualization of Neurotransmitters in Rat Brain Slices by Desorption Electrospray Ionization Mass Spectrometry Imaging (DESI - MS)”, *Journal of the American Society for Mass Spectrometry*, Journal of The American Society for Mass Spectrometry, Vol. 27 No. 12, pp. 1944–1951.



Freudiger, C.W., Min, W., Saar, B.G., Lu, S., Holtom, G.R., He, C., Tsai, J.C., et al. (2008), “Label-Free Biomedical Imaging with High Sensitivity by Stimulated Raman Scattering Microscopy”, *Science*, Vol. 322, pp. 1857–1861.

Fu, D., Yang, W. and Xie, X.S. (2017), “Label-free imaging of neurotransmitter acetylcholine at neuromuscular junctions with stimulated Raman scattering”, *Journal of the American Chemical Society*, Vol. 139 No. 2, pp. 583–586.

Grienberger, C. and Konnerth, A. (2012), “Imaging Calcium in Neurons”, *Neuron*, Elsevier Inc., Vol. 73 No. 5, pp. 862–885.

Henke, A., Kovalyova, Y., Dunn, M., Dreier, D., Gubernator, N.G., Dincheva, I., Hwu, C., et al. (2018), “Toward Serotonin Fluorescent False Neurotransmitters: Development of Fluorescent Dual Serotonin and Vesicular Monoamine Transporter Substrates for Visualizing Serotonin Neurons”, *ACS Chemical Neuroscience*, Vol. 9 No. 5, pp. 925–934.

Hill, A.H., Manifold, B. and Fu, D. (2020), “Tissue imaging depth limit of stimulated Raman scattering microscopy”, *Biomedical Optics Express*, Vol. 11 No. 2, p. 762.

Hu, F., Zeng, C., Long, R., Miao, Y., Wei, L., Xu, Q. and Min, W. (2018), “Supermultiplexed optical imaging and barcoding with engineered polyynes”, *Nature Publishing Group*, Nature Publishing Group, No. August 2017, available at:<https://doi.org/10.1038/nmeth.4578>.

Ji, M., Orringer, D.A., Freudiger, C.W., Ramkissoon, S., Liu, X., Lau, D., Golby, A.J., et al. (2013), “Rapid, Label-Free Detection of Brain Tumors with Stimulated Raman Scattering Microscopy”, *Science Translational Medicine*, Vol. 5 No. 201.

Jiang, C., Wang, Y., Song, W. and Lu, L. (2019), “Delineating the tumor margin with intraoperative surface-enhanced Raman spectroscopy”, *Analytical and Bioanalytical Chemistry*, *Analytical and Bioanalytical Chemistry*, Vol. 411 No. 18, pp. 3993–4006.

Jones, S.R., Joseph, J.D., Barak, L.S., Caron, M.G. and Wightman, R.M. (1999), “Dopamine neuronal transport kinetics and effects of amphetamine”, *Journal of Neurochemistry*, Vol. 73 No. 6, pp. 2406–2414.

Karabeber, H., Huang, R., Iacono, P., Samii, J.M., Pitter, K., Holland, E.C. and Kircher, M.F. (2014), “Guiding brain tumor resection using surface-enhanced Raman scattering nanoparticles and a hand-held Raman scanner”, *ACS Nano*, Vol. 8 No. 10, pp. 9755–9766.

Kesby, J.P., Eyles, D.W., McGrath, J.J. and Scott, J.G. (2018), “Dopamine, psychosis and schizophrenia: The widening gap between basic and clinical neuroscience”, *Translational Psychiatry*, *Springer US*, Vol. 8 No. 1, available at:<https://doi.org/10.1038/s41398-017-0071-9>.

Kim, S., Dorlhiac, G., Cotrim Chaves, R., Zalavadia, M. and Streets, A. (2021), “Paper-thin mul-

tilayer microfluidic devices with integrated valves”, *Lab on a Chip*, Royal Society of Chemistry, Vol. 21 No. 7, pp. 1287–1298.

Koo, T.-W., Chan, S. and Berlin, A.A. (2005), “Single-molecule detection of biomolecules by surface-enhanced coherent anti-Stokes Raman scattering”, *Optics Letters*, Vol. 30 No. 9, p. 1024.

Kruss, S., Salem, D.P., Vukovi, L., Lima, B., Ende, E. Vander, Boyden, E.S. and Strano, M.S. (2017), “High-resolution imaging of cellular dopamine efflux using a fluorescent nanosensor array”, *PNAS*, Vol. 3, available at:<https://doi.org/10.1073/pnas.1613541114>.

Lacin, E., Muller, A., Fernando, M., Kleinfeld, D. and Slesinger, P.A. (2016), “Construction of Cell-based Neurotransmitter Fluorescent Engineered Reporters (CNiFERs) for Optical Detection of Neurotransmitters In Vivo”, *Journal of Visualized Experiments*, No. 111, pp. 1–13.

Lecat-Guillet, N., Monnier, C., Rovira, X., Kniazeff, J., Lamarque, L., Zwier, J.M., Trinquet, E., et al. (2017), “FRET-Based Sensors Unravel Activation and Allosteric Modulation of the GABA B Receptor”, *Cell Chemical Biology*, Elsevier Ltd., Vol. 24 No. 3, pp. 360–370.

Lee, H.J., Zhang, D., Jiang, Y., Wu, X., Shih, P.Y., Liao, C.S., Bungart, B., et al. (2017), “Label-Free Vibrational Spectroscopic Imaging of Neuronal Membrane Potential”, *Journal of Physical Chemistry Letters*, Vol. 8 No. 9, pp. 1932–1936.

Lieberman, L.M., Beierwaltes, W.H., Varma, V.M., Weinhold, P. and Ling, R. (1969), “Labeled dopamine concentration in human adrenal medulla and in neuroblastoma.”, *Journal of Nuclear Medicine*, Vol. 10 No. 2, pp. 93–97.

Liu, B., Lee, H.J., Zhang, D., Liao, C.S., Ji, N., Xia, Y. and Cheng, J.X. (2015), “Label-free spectroscopic detection of membrane potential using stimulated Raman scattering”, *Applied Physics Letters*, Vol. 106 No. 17, available at:<https://doi.org/10.1063/1.4919104>.

Maity, B.K. and Maiti, S. (2018), “Label-free imaging of neurotransmitters in live brain tissue by multi-photon ultraviolet microscopy”, *Neuronal Signaling*, Vol. 2, pp. 1–14.

Marvin, J.S., Shimoda, Y., Magloire, V., Leite, M., Kawashima, T., Jensen, T.P., Kolb, I., et al. (2019), “A genetically encoded fluorescent sensor for in vivo imaging of GABA”, *Nature Methods*, Springer US, Vol. 16 No. 8, pp. 763–770.

Masharina, A., Reymond, L., Maurel, D., Umezawa, K. and Johnsson, K. (2012), “A fluorescent sensor for GABA and synthetic GABAB receptor ligands”, *Journal of the American Chemical Society*, Vol. 134 No. 46, pp. 19026–19034.

McCamant, D.W., Kukura, P. and Mathies, R.A. (2003), “Femtosecond Broadband Stimulated Raman: A New Approach for High-Performance Vibrational Spectroscopy”, *Applied Spectroscopy*, Vol. 57 No. 11, pp. 1317–1323.

Nicolson, F., Andreiuk, B., Andreou, C., Hsu, H.T., Rudder, S. and Kircher, M.F. (2019), “Non-invasive in vivo imaging of cancer using Surface-Enhanced spatially offset raman spectroscopy (SESORS)”, *Theranostics*, Vol. 9 No. 20, pp. 5899–5913.

Nuriya, M., Ashikari, Y., Iino, T., Asai, T., Shou, J., Karasawa, K., Nakamura, K., et al. (2021), “Alkyne-Tagged Dopamines as Versatile Analogue Probes for Dopaminergic System Analysis”, *Analytical Chemistry*, Vol. 93 No. 27, pp. 9345–9355.

Nutt, D.J., Lingford-Hughes, A., Erritzoe, D. and Stokes, P.R.A. (2015), “The dopamine theory of addiction: 40 years of highs and lows”, *Nature Reviews Neuroscience*, Nature Publishing Group, Vol. 16 No. 5, pp. 305–312.

Olefirowicz, T.M. and Ewing, A.G. (1990), “Dopamine concentration in the cytoplasmic compartment of single neurons determined by capillary electrophoresis”, *Journal of Neuroscience Methods*, Vol. 34 No. 1–3, pp. 11–15.

Ortiz, G., Liu, P., Deal, P.E., Nensel, A.K., Martinez, K.N., Shamardani, K., Adesnik, H., et al. (2021), “A silicon-rhodamine chemical-genetic hybrid for far red voltage imaging from defined neurons in brain slice”, *RSC Chemical Biology*, Royal Society of Chemistry, Vol. 2 No. 6, pp. 1594–1599.

Patriarchi, T., Cho, J.R., Merten, K., Howe, M.W., Marley, A., Xiong, W., Folk, R.W., et al. (2018), “Ultrafast neuronal imaging of dopamine dynamics with designed genetically encoded sensors”, *Science*, Vol. 360 No. 6396, available at:<https://doi.org/10.1126/science.aat4422>.

Pereira, D.B., Shmitz, Y., Mészáros, J., Merchant, P., Hu, G., Li, S., Henke, A., et al. (2016), “Fluorescent false neurotransmitter reveals functionally silent dopamine vesicle clusters in the striatum”, *Nature Neuroscience*, Vol. 19 No. 4, pp. 578–586.

Rodriguez, P.C., Pereira, D.B., Borgkvist, A., Wong, M.Y., Barnard, C., Sonders, M.S., Zhang, H., et al. (2013), “Fluorescent dopamine tracer resolves individual dopaminergic synapses and their activity in the brain”, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 110 No. 3, pp. 870–875.

Rothman, D.L., Petroff, O.A.C., Behar, K.L. and Mattson, R.H. (1993), “Localized <sup>1</sup>H NMR measurements of -aminobutyric acid in human brain in vivo”, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 90 No. 12, pp. 5662–5666.

Sankaranarayanan, S., De Angelis, D., Rothman, J.E. and Ryan, T.A. (2000), “The use of pHluorins for optical measurements of presynaptic activity”, *Biophysical Journal*, Elsevier, Vol. 79 No. 4, pp. 2199–2208.

Sarkar, B., Banerjee, A., Das, A.K., Nag, S., Kaushalya, S.K., Tripathy, U., Shameem, M., et al. (2014), “Label-Free Dopamine Imaging in Live Rat Brain Slices”, *ACS Chemical Neuroscience*, Vol. 5 No. 5, pp. 329–334.

Shabel, S.J., Proulx, C.D., Piriz, J. and Malinow, R. (2014), “GABA/glutamate co-release controls habenula output and is modified by antidepressant treatment”, *Science*, Vol. 345 No. 6203, pp. 1494–1498.

Shariatgorji, M., Nilsson, A., Goodwin, R.J.A., Källback, P., Schintu, N., Zhang, X., Crossman, A.R., et al. (2014), “Direct targeted quantitative molecular imaging of neurotransmitters in brain tissue sections”, *Neuron*, Vol. 84 No. 4, pp. 697–707.

Shen, Y., Xu, F., Wei, L., Hu, F. and Min, W. (2014), “Live-cell quantitative imaging of proteome degradation by stimulated raman scattering”, *Angewandte Chemie - International Edition*, Vol. 53 No. 22, pp. 5596–5599.

Shi, L., Zheng, C., Shen, Y., Chen, Z., Silveira, E.S., Zhang, L., Wei, M., et al. (2018), “Optical imaging of metabolic dynamics in animals”, *Nature Communications*, Springer US, Vol. 9 No. 2995, available at:<https://doi.org/10.1038/s41467-018-05401-3>.

Spurny, B., Seiger, R., Moser, P., Vanicek, T., Reed, M.B., Heckova, E., Michenthaler, P., et al. (2020), “Hippocampal GABA levels correlate with retrieval performance in an associative learning paradigm”, *NeuroImage*, Vol. 204 No. August 2019.

Starr, M.S. (1996), “The Role of Dopamine in Epilepsy”, *Synapse*, Vol. 22, pp. 159–194.

Stehouwer, J.S. and Goodman, M.M. (2009), “Fluorine-18 Radiolabeled PET Tracers for Imaging Monoamine Transporters: Dopamine, Serotonin, and Norepinephrine”, *PET Clinics*, Vol. 4 No. 1, pp. 101–128.

Steuwe, C., Kaminski, C.F., Baumberg, J.J. and Mahajan, S. (2011), “Surface enhanced coherent anti-stokes raman scattering on nanostructured gold surfaces”, *Nano Letters*, Vol. 11 No. 12, pp. 5339–5343.

Südhof, T.C. (2004), “The Synaptic Vesicle Cycle”, *Annu. Rev. Neurosci*, Vol. 27, pp. 509–547.  
Sun, F., Zeng, J., Jing, M., Zhou, J., Feng, J., Owen, F., Luo, Y., et al. (2018), “A genetically-encoded fluorescent sensor enables rapid and specific detection of dopamine in flies, fish, and mice”, *Cell*, Vol. 174 No. 2, pp. 481–496.

Warren, N., O’Gorman, C., Lehn, A. and Siskind, D. (2017), “Dopamine dysregulation syndrome in Parkinson’s disease: A systematic review of published cases”, *Journal of Neurology, Neurosurgery and Psychiatry*, Vol. 88 No. 12, pp. 1060–1064.

Watson, C.J., Venton, B.J. and Kennedy, R.T. (2006), “In vivo measurements of neurotransmitters by microdialysis sampling”, *Analytical Chemistry*, Vol. 78 No. 5, pp. 1391–1399.

Wei, L., Hu, F., Shen, Y., Chen, Z., Yu, Y., Lin, C.C., Wang, M.C., et al. (2014), “Live-cell imaging of alkyne-tagged small biomolecules by stimulated Raman scattering”, *Nature Methods*,

Vol. 11 No. 4, pp. 410–412.

Westerink, R.H.S. and Ewing, A.G. (2008), “The PC12 cell as model for neurosecretion”, *Acta Physiologica*, Vol. 192 No. 2, pp. 273–285.

Westerink, R.H.S., De Groot, A. and Vijverberg, H.P.M. (2000), “Heterogeneity of catecholamine-containing vesicles in PC12 cells”, *Biochemical and Biophysical Research Communications*, Vol. 270 No. 2, pp. 625–630.

Wightman, R.M., Jankowski, J.A., Kennedy, R.T., Kawagoe, K.T., Schroeder, T.J., Leszczyszyn, D.J., Near, J.A., et al. (1991), “Temporally resolved catecholamine spikes correspond to single vesicle release from individual chromaffin cells”, *Proceedings of the National Academy of Sciences of the United States of America*.

Wu, Y., Wang, W., Díez-Sampedro, A. and Richerson, G.B. (2007), “Nonvesicular Inhibitory Neurotransmission via Reversal of the GABA Transporter GAT-1”, *Neuron*, Vol. 56 No. 5, pp. 851–865.

Yamakoshi, H., Dodo, K., Okada, M., Ando, J., Palonpon, A., Fujita, K., Kawata, S., et al. (2011), “Imaging of EdU, an alkyne-tagged cell proliferation probe, by Raman microscopy”, *Journal of the American Chemical Society*, Vol. 133 No. 16, pp. 6102–6105.

Yamakoshi, H., Palonpon, A., Dodo, K., Ando, J., Kawata, S., Fujita, K. and Sodeoka, M. (2015), “A sensitive and specific Raman probe based on bisarylbutadiyne for live cell imaging of mitochondria”, *Bioorganic and Medicinal Chemistry Letters*, Elsevier Ltd, Vol. 25 No. 3, pp. 664–667.

Zhang, Y., Zhen, Y.R., Neumann, O., Day, J.K., Nordlander, P. and Halas, N.J. (2014), “Coherent anti-Stokes Raman scattering with single-molecule sensitivity using a plasmonic Fano resonance”, *Nature Communications*, Nature Publishing Group, Vol. 5, pp. 1–7.

Zhao, Z., Shen, Y., Hu, F. and Min, W. (2017), “Applications of vibrational tags in biological imaging by Raman microscopy”, *The Analyst*, Royal Society of Chemistry, pp. 4018–4029.

Zhu, H., Zou, G., Wang, N., Zhuang, M., Xiong, W. and Huang, G. (2017), “Single-neuron identification of chemical constituents, physiological changes, and metabolism using mass spectrometry”, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 114 No. 10, pp. 2586–2591.

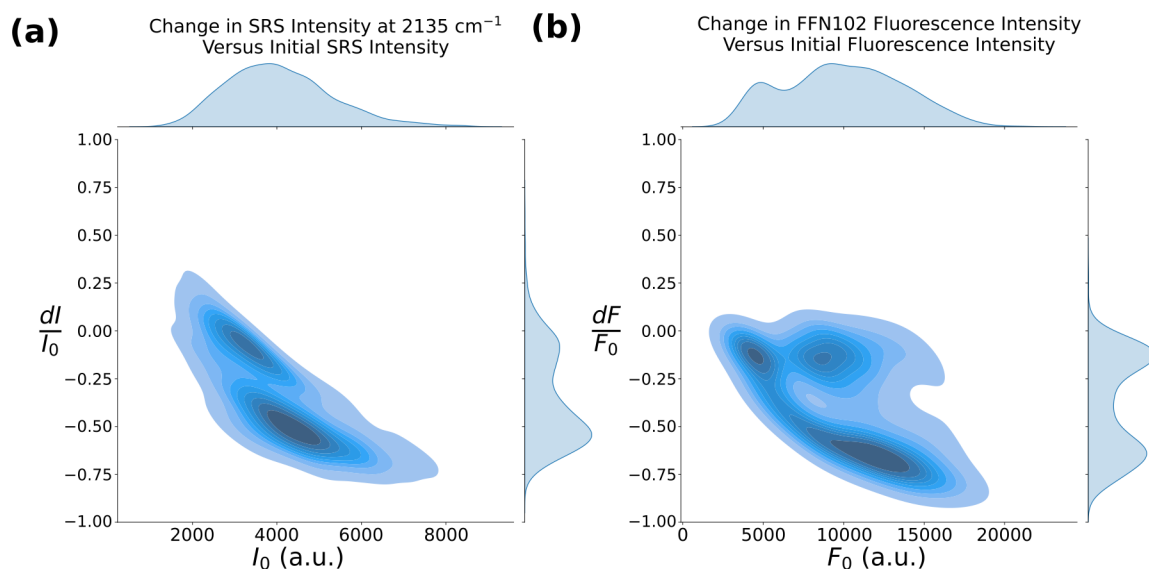
Zhu, K.Y., Fu, Q., Leung, K.W., Wong, Z.C.F., Choi, R.C.Y. and Tsim, K.W.K. (2011), “The establishment of a sensitive method in determining different neurotransmitters simultaneously in rat brains by using liquid chromatography-electrospray tandem mass spectrometry”, *Journal of Chromatography B: Analytical Technologies in the Biomedical and Life Sciences*, Elsevier B.V., Vol. 879 No. 11–12, pp. 737–742.

Zong, C., Cheng, R., Chen, F., Lin, P., Zhang, M., Chen, Z., Li, C., et al. (2021), “Wide-Field Surface-Enhanced Coherent Anti-Stokes Raman Scattering Microscopy”, *ACS Photonics*, avail-

able at:<https://doi.org/10.1021/acsp Photonics.1c02015>.

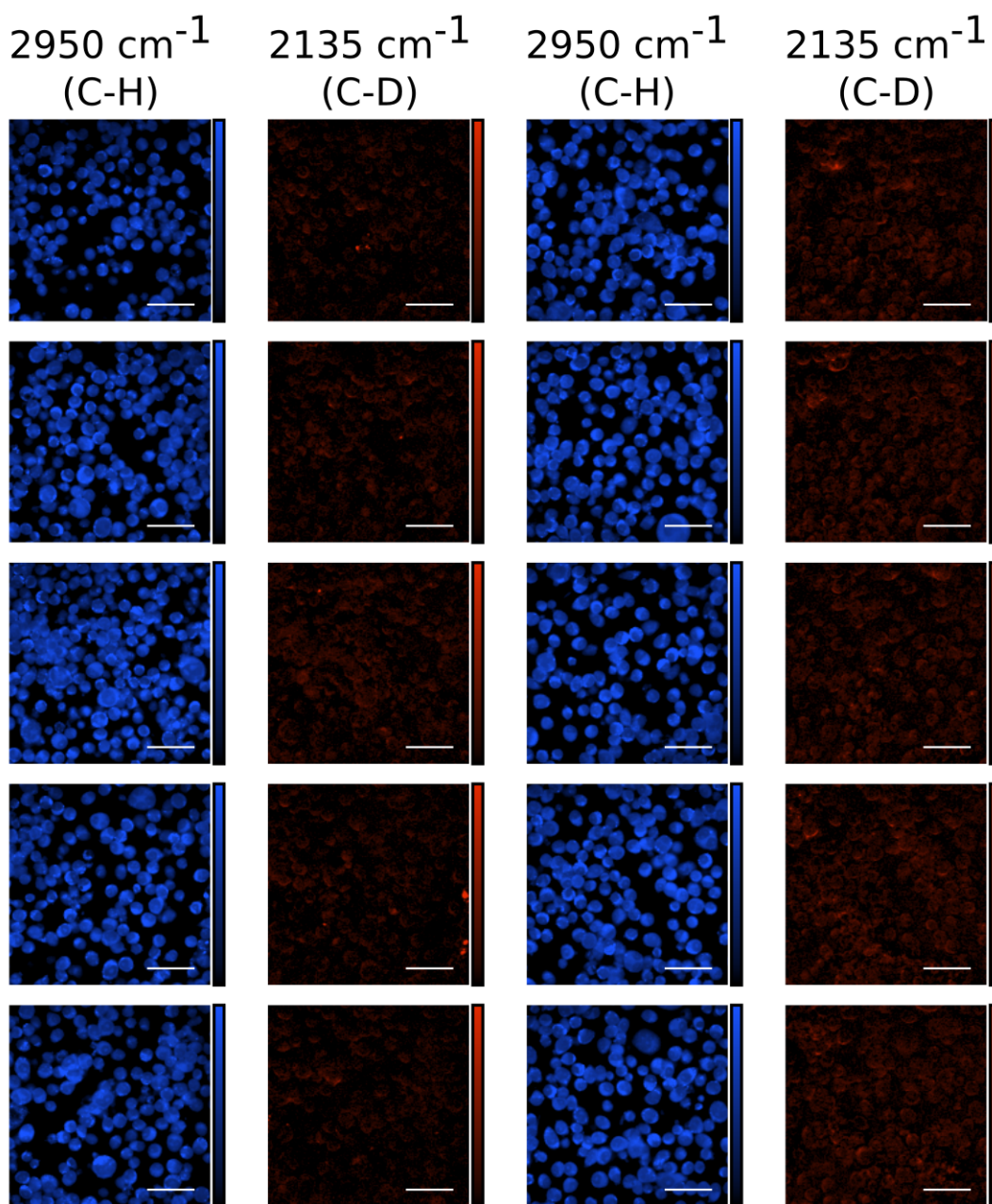
Zong, C., Premasiri, R., Lin, H., Huang, Y., Zhang, C., Yang, C., Ren, B., et al. (2019), “Plasmon-enhanced stimulated Raman scattering microscopy with single-molecule detection sensitivity”, *Nature Communications*, Springer US, Vol. 10 No. 1, pp. 1–11.

## 4.9 Supplementary Material



**Figure 4.8:** Relative intensity change of d<sub>4</sub>-DA and FFN102 due to potassium stimulation versus initial intensity.

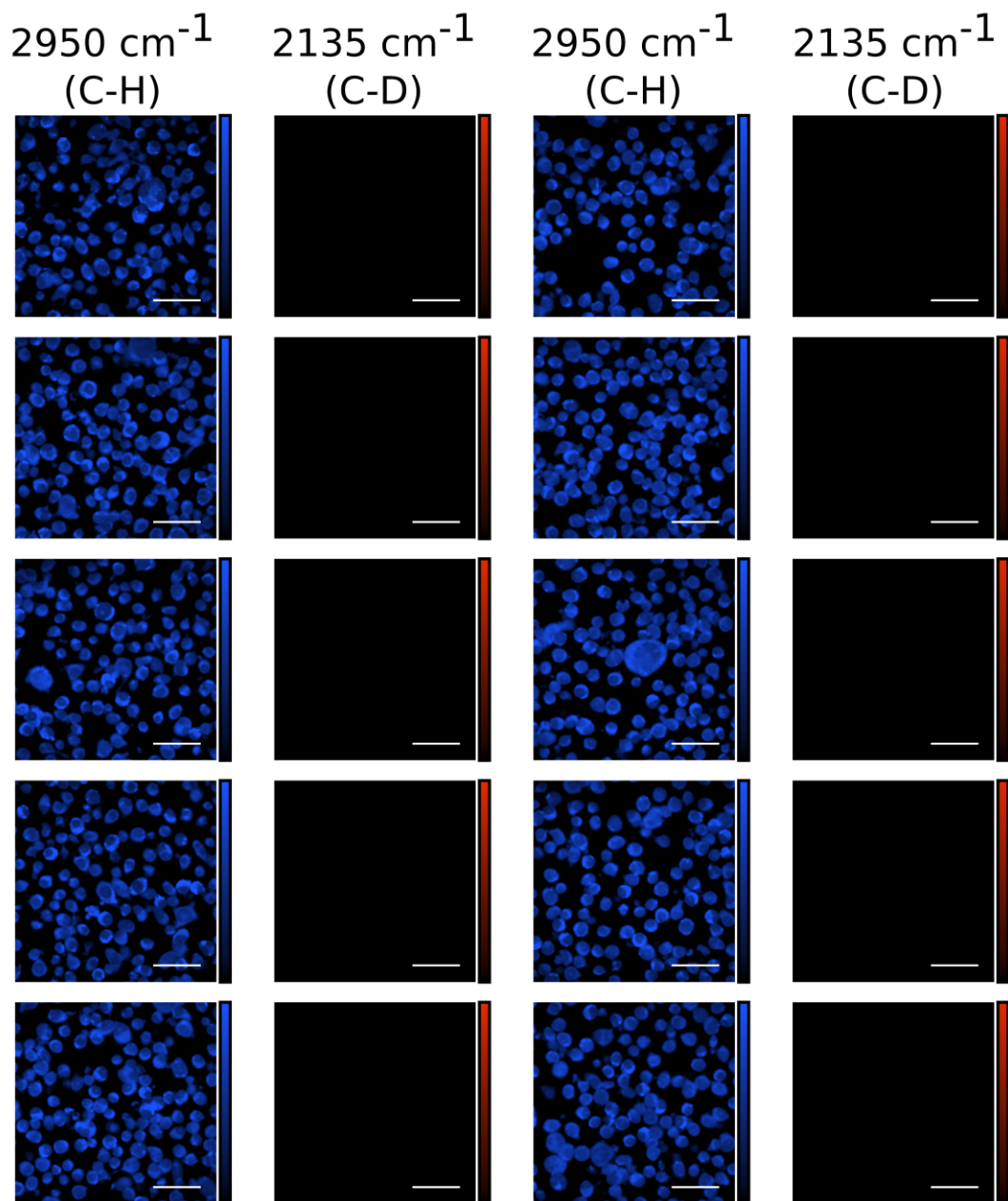
(a) The two-dimensional histogram displaying the relative change in SRS intensity of d<sub>4</sub>-DA at 2135 cm<sup>-1</sup> due to potassium stimulation, versus the initial SRS intensity at 2135 cm<sup>-1</sup> per cell. The one-dimensional histograms for the relative change distribution, and initial intensity distribution, are displayed on the right and top axes, respectively. The distribution on the right axis corresponds to the violin plot depicted in figure 4.4b. (b) The corresponding two-dimensional histogram for the relative change in FFN102 fluorescence intensity due to potassium stimulation versus the initial fluorescence intensity per cell. The one-dimensional histogram on the right axis corresponds to the violin plot in figure 4.4c. There is an additional third population of cells among those incubated with FFN102, not present in the PC12 cells incubated with d<sub>4</sub>-DA, which display an high initial fluorescence intensity but little to no change in intensity in response to potassium stimulation.



**Figure 4.9:** Representative SRS images of PC12 cells incubated with  $d_4$ -DA.

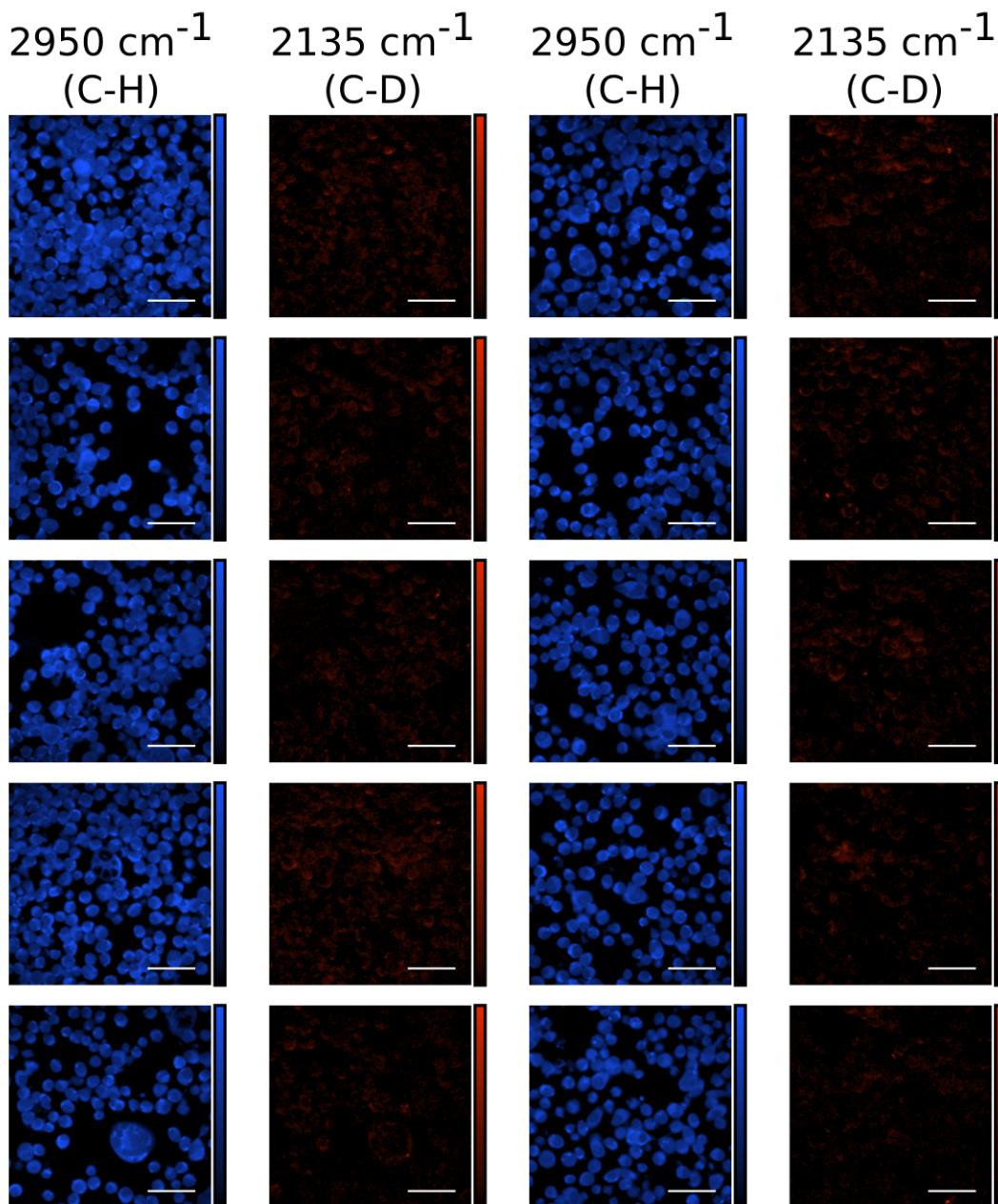
The first and second columns, and third and fourth columns are paired images acquired of the same field of view in different SRS channels. In each pair, the left column displays images at  $2950\text{ cm}^{-1}$ , the C-H stretching frequency range, broadly corresponding to protein signal and allowing cell localization. The right column displays images at  $2135\text{ cm}^{-1}$ , the C-D stretching frequency range, corresponding to the signal originating from  $d_4$ -DA. Scale bar =  $50\text{ }\mu\text{M}$ .





**Figure 4.10:** Representative SRS images of PC12 cells not incubated with d<sub>4</sub>-DA.

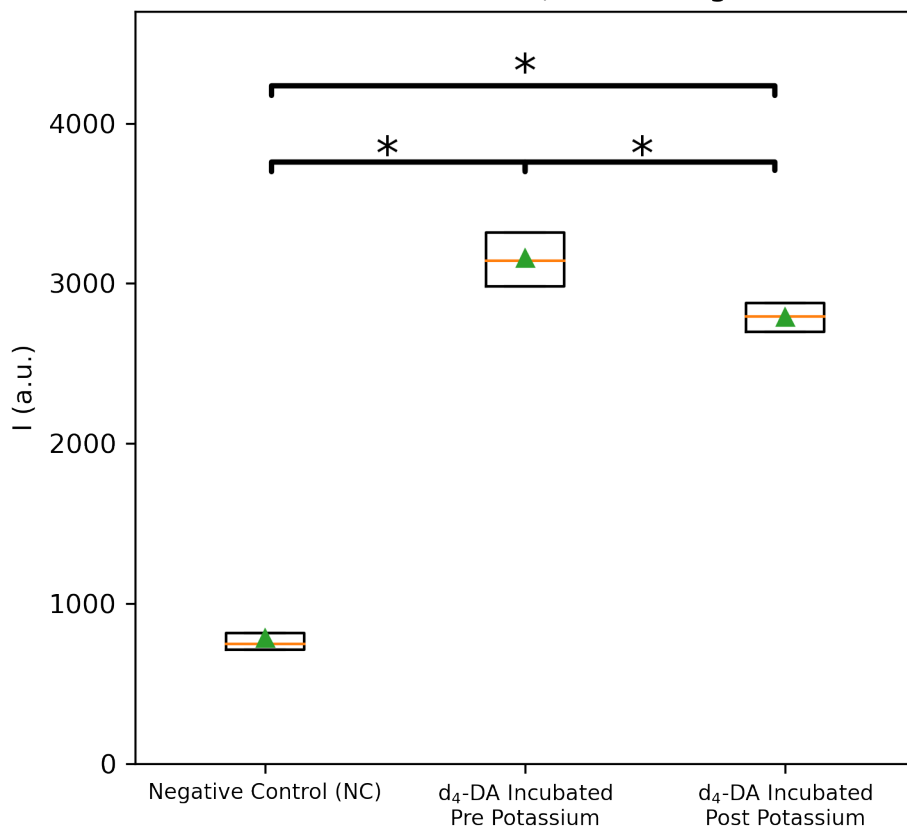
The first and second columns, and third and fourth columns are paired images acquired of the same field of view in different SRS channels. In each pair, the left column displays images at 2950  $\text{cm}^{-1}$ , the C-H stretching frequency range, broadly corresponding to protein signal and allowing cell localization. The right column displays images at 2135  $\text{cm}^{-1}$ , the C-D stretching frequency range, corresponding to the signal originating from d<sub>4</sub>-DA. Scale bar = 50  $\mu\text{M}$ .



**Figure 4.11:** Representative SRS images of PC12 cells incubated with  $d_4$ -DA two minutes after stimulation with 50 mM potassium.

The first and second columns, and third and fourth columns are paired images acquired of the same field of view in different SRS channels. In each pair, the left column displays images at  $2950\text{ cm}^{-1}$ , the C-H stretching frequency range, broadly corresponding to protein signal and allowing cell localization. The right column displays images at  $2135\text{ cm}^{-1}$ , the C-D stretching frequency range, corresponding to the signal originating from  $d_4$ -DA. Scale bar =  $50\text{ }\mu\text{M}$ .

SRS Intensity at  $2135\text{ cm}^{-1}$  Pre and Post Stimulation  
For Cells Incubated with  $d_4$ -DA vs Negative Control



**Figure 4.12:** Cell-averaged SRS intensity distributions for cells in the fields of view included in figures 4.9-4.11.

Cell-averaged SRS intensity for cells that were not incubated with  $d_4$ -DA (left column), cells incubated with  $d_4$ -DA (middle column), and cells incubated with  $d_4$ -DA and stimulated to release  $d_4$ -DA with 50 mM potassium (right column). \*  $p < 0.001$ ,  $\Delta > 23$  (incubated pre stimulation vs negative control),  $\Delta > 0.24$  (incubated pre stimulation vs incubated post stimulation),  $\Delta > 19$  (incubated post stimulation vs negative control).  $N(\text{negative control}) = 400$ ,  $N(\text{Incubated, pre-stimulation}) = 364$ ,  $N(\text{Incubated, post-stimulation}) = 323$ .

## 4.10 Appendix

### 4.10.1 A note on cell culture

SRS microscopy, and imaging techniques generally, rely on samples remaining stationary on the timescale of data acquisition. Sample motion can lead to imaging artifacts that are improperly interpreted as features of a static object. Given that image acquisition in SRS microscopy takes on the order of a few seconds, adherent cells are usually sufficiently stationary to avoid motion artifacts. On the other hand, non-adherent cells can exhibit detectable motion from the slightest mechanical perturbations during imaging, affecting the measured signals. The PC12 cell line used in this study is **not** strongly adherent (Wiatrak et al., 2020). The cells also tend to stick to themselves, forming 3D spherical structures instead of monolayers (figure 4.13a). During differentiation the cells more readily adhere to their growth substrate, either glass or plastic; however, they remain easily dissociable. As such it is important to further reflect on the protocols used to gather the data presented above.

The PC12 cell line used was repeatedly passaged under different culture conditions, resulting in a more adherent cell culture that grew efficiently as a monolayer (figure 4.13b). The entire process took approximately one month; however, each step and culture condition was not independently validated. As such fewer conditions may be sufficient to obtain the same end result. The growth medium conditions, in order, are as follows:

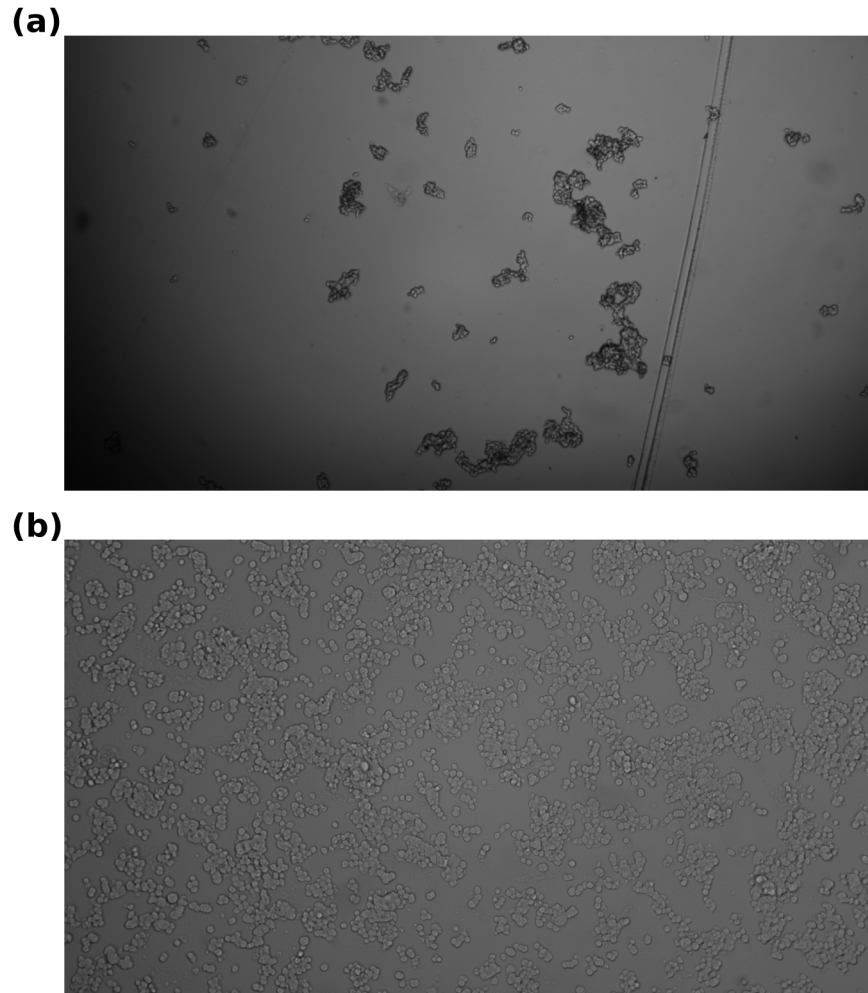
1. RPMI-1640 medium with L-glutamine, containing 10% goat serum, 10% FBS, and 1% penicillin/streptomycin
2. RPMI-1640 medium with L-glutamine, containing 10% goat serum, 5% FBS, and 1% penicillin/streptomycin
3. RPMI-1640 medium with L-glutamine, containing 10% goat serum, and 1% penicillin/streptomycin
4. DMEM with L-glutamine and without pyruvate, containing 15% FBS and 1% penicillin/streptomycin
5. DMEM with L-glutamine and without pyruvate, containing 10% FBS and 1% penicillin/streptomycin, grown on collagen treated substrate
6. DMEM with L-glutamine and without pyruvate, containing 5% FBS and 1% penicillin/streptomycin
7. DMEM with L-glutamine and without pyruvate, containing 1% FBS and 1% penicillin/streptomycin
8. RPMI-1640 medium with L-glutamine, containing 10% goat serum, 10% FBS, and 1% penicillin/streptomycin

Notably, one of the culture conditions (DMEM + 10% FBS) was used while growing the cells on a collagen coated substrate; however, this was the only condition in which collagen was used. Collagen is a well-known additive for promoting cell adhesion. However, while it is sometimes

used in SRS microscopy experiments, we found that it contributed a strong background signal at  $2950\text{ cm}^{-1}$  which made distinguishing cells challenging. This is hypothesized to be due to some out of plane SRS signal generation. The reason this was a factor in our experiment but not others is likely due to the use of unchirped femtosecond pulses which can more readily generate out of plane signal. The femtosecond pulse duration, however, was necessary to better detect the weaker signals from deuterated compounds. Substrate treatment with lysine, another promoter of cell adhesion, also produced the background signals.

Additionally, serum deprivation, used in some of the conditions, is a part of the differentiation protocol. In fact, except for the absence of NGF- $\beta$ , the second to last culture condition listed is identical to the final differentiation medium used. As a result, it is possible that the increase in adhesion is due to differentiation/partial differentiation. The cells remained proliferative, however, and, for the most part, did not display the characteristic neurite growth. As cells exposed to the differentiation medium were ultimately used for imaging, using "partially" differentiated cells would also not necessarily be detrimental.

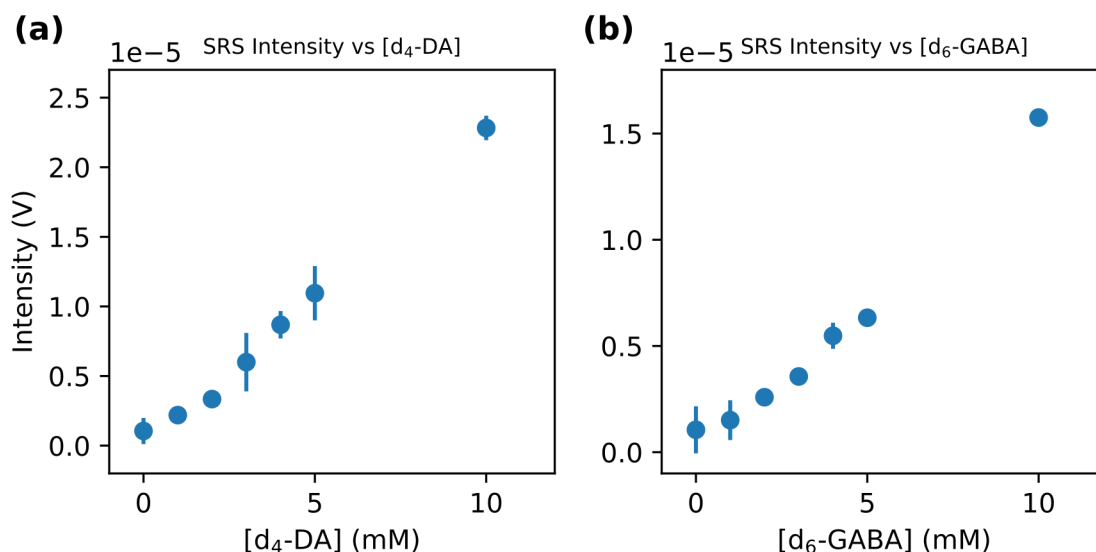
Finally, it should be noted that an adherent version of the PC12 cell line has previously been developed. For example, it is available from ATCC. However, this particular line has been found to be inappropriate for neuroscience-based studies (Wiatrak et al., 2020). This was the original motivation behind using the standard, non-adherent, variant. This study, however, did not set out to validate the PC12 cell line produced for use as a neuroscience model. The cell line was able to demonstrate the capability to detect a relevant neurotransmitter isotopologue, and responded in an appropriate and anticipated way to potassium stimulation. Additional validation may be warranted to use this technique with these particular cells; however, as a general tool, the methodology can be applied to other model systems.



**Figure 4.13:** Non-adherent and adherent PC12 cells

**(a)** PC12 as they were originally received from UC Berkeley’s cell culture facility. The cells are very weakly adherent/non-adherent. They cluster into 3D quasi-spherical structures with few attachments to the substrate. If that attachment breaks, many cells are lost simultaneously. **(b)** PC12 cells after repeated passaging under different culture conditions and serum deprivation. The cells form more attachments to the substrate and grow more readily as a monolayer. The attachments are still weak enough to be dissociated with pipetting, but are sufficient for imaging experiments. Fewer 3D clusters of cells exist.

### 4.10.2 A note on sensitivity



**Figure 4.14:** SRS signal intensity vs deuterated neurotransmitter concentration

The SRS signal intensity was recorded as a function of the concentration of d<sub>4</sub>-DA and d<sub>6</sub>-GABA. The signal was measured using the experimental parameters for the study presented above. I.e., pump and Stokes powers were set to 20 mW and a 3  $\mu$ s time constant was used on the lock-in amplifier. (a) The resultant signal vs concentration curve for d<sub>4</sub>-DA. (b) The resultant signal vs concentration curve for d<sub>6</sub>-GABA.

The methodology as presented in this study lacks the sensitivity required to investigate extracellular populations of neurotransmitter. Under the excitation conditions used, limited by phototoxicity, the detection limits of the deuterated analogues are approximately in the range of 1-2 mM (figure 4.14). As a result, study of extracellular populations requiring orders of magnitude increases in sensitivity remain out of reach. Nonetheless the methodology fills an important role in its ability to detect intracellular populations. Coupled with additional tools, such as extracellular nanoparticle-based neurotransmitter sensors, this technique can contribute to a complete understanding of neurotransmitter concentrations and dynamics.

Additionally, Raman can potentially be used to detect extracellular populations as well, given appropriate signal enhancement techniques. In particular, surface enhancement has the possibility to achieve single molecule sensitivity. As mentioned previously, surface enhancement has recently been shown to be possible when used with stimulated Raman scattering (Zong et al., 2019). The use of surface-enhanced spontaneous Raman has furthermore been shown to be a valuable tool in certain neuroscience, and even clinical, contexts, such as when guiding tumor resection (Karabeber et al. 2014; Yamazoe et al., 2014; Harmen et al., 2016; Huang et al., 2016). The success of other nanoparticle-based sensors in neuroscience (Wang et al., 2010) further supports the possibility of using plasmonic nanoparticles for surface-enhanced SRS (SESRS). Taken together, this recent advancement, the successful historical application of surface-enhancement in neuroscience, and the proliferation of nanoparticle-based neurotransmitter sensors suggest that SESRS may be a viable strategy for the detection of neurotransmitter isotopologues both intra- and extracellularly. We

tested the viability of this strategy by attempting to extend the detectable concentration range of our dopamine isotopologue.

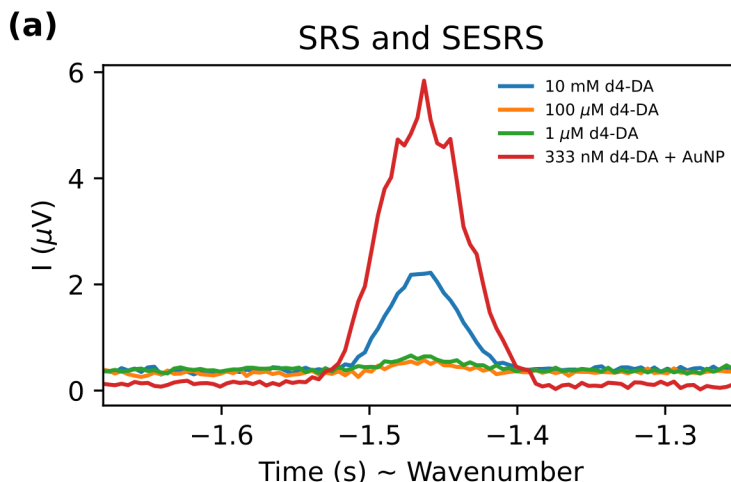
Citrate-coated gold nanospheres, with a diameter of 70-80 nm, provided by OndaVia Inc., were used to attempt surface-enhanced detection of  $d_4$ -DA. Generally, surface enhancement is most efficient when the excitation wavelengths fall within, or near, the plasmonic resonance of the nanoparticle used. Both the pump and Stokes wavelengths used for  $d_4$ -DA detection are well within the near infrared wavelength range (851 nm and 1040 nm, respectively). Unfortunately, the plasmonic resonance of gold nanospheres of this diameter is in the visible range; however, nanoparticle aggregation broadens and shifts the resonance to longer wavelengths closer to the excitation wavelengths used. As such, aggregation was promoted by exposure to concentrated NaCl (1 M) (Christau et al., 2017). The nanoparticle aggregates were then dried on a coverslip for imaging.

The SRS (or SESRS) signal of solutions of various concentrations of  $d_4$ -DA was then measured using pulse powers of 10 mW each (pump and Stokes) for SRS measurements, and a total combined power of <2 mW for SESRS measurements. The exact powers of the pump and Stokes pulses could not be determined for the SESRS measurements due to the resolution of the available power meter. As a background signal of nanoparticle absorption needs to be subtracted from SESRS measurements (refer to Zong et al., 2019, for information), the signals were acquired by scanning the interpulse delay between the pump and Stokes pulses. The absorption due to nanoparticles leads to a temporally long-lived signal compared to the sharp SRS signal that occurs when the two pulses are perfectly overlapped in time.

For SRS measurements, while 10 mM concentrations of  $d_4$ -DA could readily be detected, signals from concentrations in the  $\mu$ M range were not measurable. On the other hand, after background subtraction, signals in the 100s of nM range were apparently detectable at gold nanosphere aggregates (figure 4.10.2a). One should note, however, that the blind aggregation strategy used to shift the nanoparticle plasmon resonance comes with a number of caveats and disadvantages. First, there is a loss of homogenous signal coverage. The aggregate size distribution is non-uniform leading to varying degrees of signal enhancement and uneven substrate coverage. This leads to an inability to accurately quantify analyte concentration, if it is not known *a priori*. The lack of concentration quantification may potentially be of less importance, if the kinetics and dynamics are the key parameters of interest; however, heterogenous substrate coverage means that potentially important signals can be missed. As such, to generate a more reliable signal it will be worth studying the aggregation process and focusing on reproducible generation.

Nonetheless, these preliminary observations suggest that SESRS could potentially be an additional tool for studying neurotransmitter populations and kinetics.





### 4.10.3 References (Appendix)

Christau, S., Moeller, T., Genzer, J., Koehler, R., and von Klitzing, R. (2017) "Salt-induced aggregation of negatively charged gold nanoparticles confined in a polymer brush matrix", *Macromolecules*, Vol. 50 No. 18.

Harmsen, S., Huang, R., Wall, M. A., Karabeber, H., Samii, J. M., Spaliviero, M., White, J. R., Monette, S., Sastra, S. A., Saborowski, M., Holland, E. C., Singer, S., Olive, K. P., Lowe, S. W., Blasberg, R. G., and Kircher, M. F. (2016), "Surface-Enhanced Resonance Raman Scattering Nanostars for High Precision Cancer Imaging", *Sci Transl Med*, Vol. 7 No. 271.

Huang, R., Harmsen, S., Samii, J. M., Karabeber, H., Pitter, K. L., Eric, C., and Kircher, M. F. (2016), "High Precision Imaging of Microscopic Spread of Glioblastoma with a Targeted Ultra-sensitive SERRS Molecular Imaging Probe", *Theranostics*, Vol. 6 No. 8, pp. 1075–1084.

Karabeber, H., Huang, R., Iacono, P., Samii, J. M., Pitter, K., Holland, E. C., and Kircher, M. F. (2014), "Guiding brain tumor resection using surface-enhanced Raman scattering nanoparticles and a hand-held Raman scanner", *ACS Nano*, Vol. 8 No. 10, pp. 9755–9766.

Yamazoe, S., Naya, M., Shiota, M., Morikawa, T., Kubo, A., Tani, T., Hishiki, T., Horiuchi, T., Suematsu, M., and Kajimura, M. (2014), "Large-area surface-enhanced raman spectroscopy imaging of brain ischemia by gold nanoparticles grown on random nanoarrays of transparent boehmite", *ACS Nano*, Vol. 8 No. 6, pp. 5622–5632.

Wang, Y., Seebald, J. L., Szeto, D. P., and Irudayaraj, J. (2010), "Biocompatibility and Biodistribution of Surface-Enhanced Raman Scattering Nanoprobes in Zebrafish Embryos: In vivo and Multiplex Imaging", *ACS Nano*, Vol. 4 No. 7, pp. 4039–4053.

Wiatrak, B., Kubis-Kubiak, A., Piwowar, A., and Barg, E. (2020), "PC12 cell line: cell types, coating of culture vessels, differentiation and other culture conditions", *Cells*, Vol. 9 No. 4).

# Chapter 5

## Compressed sensing SRS (CSSRS) microscopy

\*This chapter contains results which have not been previously published.

### 5.1 Introduction

Compared to the introduction of fluorescence microscopy, the uptake of SRS microscopy as a routine analysis tool in the life sciences has been relatively slow. On the one hand, there is the issue of accessibility as there are high costs, both monetary, and in expertise, associated with maintaining a SRS microscope and making measurements. A slowly, but steadily, increasing community of experts in the field will help to alleviate this issue. Furthermore, a number of companies are addressing the need for turn-key instrumentation to help flatten the learning curve. There are now commercially available SRS microscopes for basic research which come fully assembled. Additionally, at least one company is bringing a SRS-based product to market for clinical applications such as histology (Orringer et al. 2017; Einstein et al., 2022; Neidert et al. 2022).

Beyond these barriers, however, two other issues faced by SRS microscopy are those of sensitivity and acquisition speed. Fluorescence is highly sensitive, down even to the single-molecule level. Despite the appealing characteristics of SRS detection, such as minimal sample perturbation and linear intensity and concentration signal dependence, in its currently adopted format it is not capable of such sensitivity. As a result, the majority of applications of SRS microscopy in the life sciences have focused on high concentration biomolecules, measuring bulk protein, or bulk lipid, for example (Hill and Fu, 2019; Hu, Shi, and Min 2019). In chapter 4, we demonstrated that under certain conditions it is possible to use SRS microscopy to study specific small molecules, i.e., neurotransmitters; however, we were again relying on the biology to help us by maintaining comparatively high concentrations of the molecule of interest. In the appendix to chapter 4, we showed how it may be possible to extend the detectable range of neurotransmitter through the use of surface-enhancement. Other strategies to increase sensitivity include the use of specially designed Raman tags which make use of electronic resonant enhancement to increase signal (Wei and Min, 2018; Shi et al., 2022).

Another possible solution to sensitivity issues would seem to be to increase excitation power, however, issues of phototoxicity quickly arise. Our recently published work, among other find-

ings, suggests that biological samples are less sensitive to the effects of increased exposure time compared to increased excitation power (Zhang et al., 2022). As such, one possibility would be to increase the pixel dwell-time to more reliably observe small changes in signal intensity. Note that while signal does not increase with pixel dwell time, as discussed in chapter 2, the noise can be reduced. This strategy, while useful for certain applications, such as when working with fixed samples, runs into the second limiting factor of SRS microscopy: slow acquisition. When imaging dynamic processes, such as neurotransmitter release, for example, acquisition times in SRS microscopy can become too long to capture the relevant information, or, in the worst case, lead to motion artifacts in the detected image. Furthermore, long acquisition times lead to lower experimental throughput, making application to problems such as cytometry unappealing. A number of examples of high-speed SRS imaging systems have been reported in the literature (Saar et al., 2010; He et al., 2017; Nao et al., 2020); however, they rely on custom electronics further raising the barrier to adoption.

It would seem then, that in the absence of surface-enhancement or the ability to use certain dyes, high sensitivity and rapid SRS microscopy is infeasible, limiting the classes of life science problems to which it can be applied. This is the common tradeoff between speed and SNR. However, by asking a slightly different question, new strategies appear. Namely, how much, or what portions of an image need to be measured to get the information needed? It is well-known that images are sparse (in some basis). This forms the basis of compression. Spatial correlations within an image mean that the number of pixels we use to represent the image are redundant. Using common compression algorithms, images can be stored losslessly with 10% (or less) of memory requirement to store the raw data. This suggests that it may be possible to sparsely sample an image and reconstruct it afterwards.

Indeed, recent theoretical (Lin et al., 2017) and proof-of-principle experimental work (Lin et al., 2018; shows that it is possible to reconstruct SRS images with high degrees of compression/undersampling. Even at 10%-20% sampling, reliable reconstructions can be obtained. The modifications to a traditional SRS microscope required to sparsely sample a field of view (FOV) in the above implementations are non-trivial, however. An alternative approach, that achieves a similar result, is to use compressed sensing (CS), the theory of which was first outlined in the mid-2000s (Candes, Romberg, and Tao, 2006; Donoho, 2006; Tsaig and Donoho, 2006). In recent years, the field of spontaneous Raman microscopy has benefited greatly from the introduction of CS techniques to achieve the desired undersampling, and concomitant increase in acquisition speed (Davis et al., 2010; Thompson et al., 2017; Zhang et al., 2020). Using CS techniques, spontaneous Raman microspectroscopy has been able to reach pixel dwell times on the order of 10s-100s of  $\mu\text{s}$  - orders of magnitude faster than traditional techniques (Soldevila et al., 2019; Scotté et al., 2020).

Compressed sensing relies on a number of key factors. First, it requires that the data we wish to reconstruct has a sparse representation, at least in some basis. As mentioned, this is known to be true for SRS images. Second, it relies on the ability to acquire information from multiple spectral or spatial components simultaneously, i.e. to make multiplexed measurements. This is similar to techniques such as Hadamard or Fourier transform spectroscopy which measure information from multiple places in spectrum simultaneously. Unlike in these spectroscopies, however, the total number of measurements made in CS is fewer than the number of data points ultimately reconstructed. This can be accomplished because each measurement contains information about multiple places in the sample (or multiple spectral components) and we are assuming that the data has a sparse representation.

In Hadamard spectroscopy, multiplexed measurements are accomplished through a variety of means, originally, though, through the use of multi-slit apertures (Nelson and Fredman, 1970; Sloane and Harwit, 1976). However, in the 1980's Texas Instruments introduced the first digital micromirror devices (DMD). These optoelectronics consist of an array of individually addressable micrometer scale mirrors. Each mirror is on the size order of a pixel in a digital camera, allowing them to be manufactured on a compact scale. When placed at the Fourier plane of an imaging system they can be used as a coded aperture, determining which components of either spectral, or spatial, information are directed through the rest of the imaging path, and ultimately to the detector.

In the following chapter we explore the use of a DMD to make CSSRS (compressed sensing SRS) measurements. By using a DMD, and a line focus at the sample, multiplexed spatial measurements can be acquired, allowing reconstruction of an image with fewer measurements than the total number of pixels in the image. Likewise, the DMD can be used in conjunction with a polychromatic pump excitation beam and grating to acquire hyperspectral data using CS reconstruction techniques. Additionally, as a DMD is two dimensional, both space and spectrum can be multiplexed and reconstructed using CS simultaneously (Scotté et al., 2020).

The use of a DMD offers a number of advantages over alternative solutions. To begin with, DMDs are commercially available. The technology is robust and has found application in a variety of fields from projector systems to basic science research (Yoon et al., 2018). This level of adoption ensures reliability, and comparative turn-key operation compared to bespoke electronics. Additionally, it allows for the fast transfer and uptake of the technology for any laboratories that are potentially interested in applying it. Furthermore, DMDs can be operated at rapid pattern rates (10s of kHz) allowing for increases in acquisition speed. The results presented in this chapter suggest that DMD-based CSSRS is an avenue to more rapid data acquisition without loss of SNR, and potentially a viable option for measurements made in a low-power regime, such as when phototoxicity becomes limiting.

## 5.2 Results

### 5.2.1 DMD-based detection supports multiple experimental modes

Each micromirror of a DMD is individually addressable, and can generally be in one of two states, on or off. We denote the on state as the state in which light incident on the micromirror is directed to a detector, and the off state as the state in which incident light is directed towards a beam dump. Combinations of on or off mirrors along the rows and/or columns of the DMD constitute binary filters which direct light from different spatial or spectral components to the detector. Through the introduction or removal of a small number of optical elements, the use of DMD-based binary filters allows for CSSRS measurements to be made spatially, spectrally, or in both domains (figure 5.1). Furthermore, switching between imaging modes is straightforward.

By placing a grating after the condenser, as in figure 5.1a, a polychromatic pump beam can be dispersed and then focused onto the DMD using a cylindrical lens, allowing for multiplexing of the spectral components sent towards the detector. To maintain high spectral resolution in this configuration, the instantaneous bandwidth of the Stokes beam must be small while that of the pump beam is large. This can be achieved by stretching the Stokes beam, as described in chapter 2, while the pump remains unstretched. An alternative strategy is to use spectral filtering of the Stokes

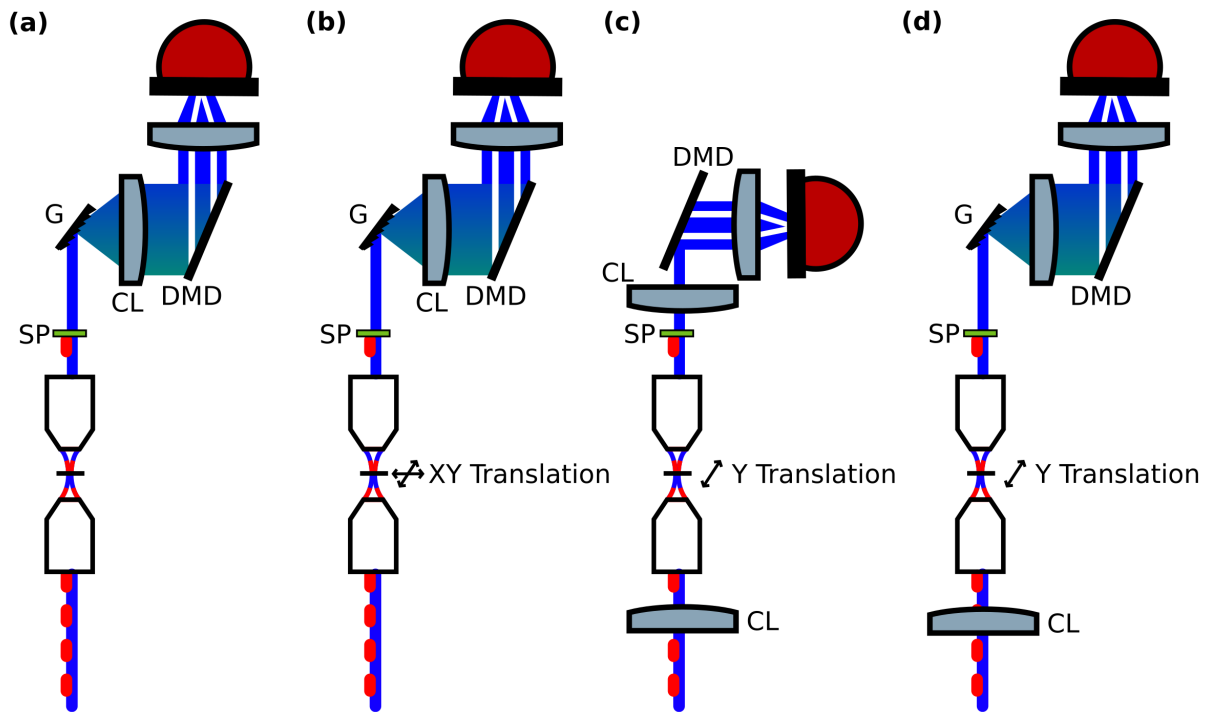
beam, by, e.g., placing a slit in the grating systems previously covered, while again leaving the pump beam unmodified. This has the effect of producing a narrowband picosecond pulse instead of a chirped picosecond pulse. In the configuration of figure 5.1a, the microscope functions as a spectrometer. Multiple DMD patterns are used, resulting in a set of multiplexed measurements, which are used to retrieve the spectrum. If the total number of patterns used is equivalent to the number of data points in the retrieved spectrum, this is equivalent to performing traditional Hadamard spectroscopy. If instead the number of patterns used is fewer than the number of data points, it is compressed sensing. This imaging mode is explored in more depth below.

With a slight change, as shown in figure 5.1b, microspectroscopy can be performed in a multiplexed/CS manner. The only difference compared to figure 5.1a is that the sample is scanned in the x and y dimensions. Note that, traditionally, pump-probe microscopes make use of scanning galvo or resonant mirrors to raster scan the beam across the sample. While fast, and accurate, the use of these mirrors leads to movement of the beams through downstream optics on the detection path. Given the alignment sensitivity of the grating and DMD components, it is simpler to maintain a system with a stationary beam, and instead scan the sample through it. It should be noted that this is no longer the drawback it once was. Top-of-the-line sample stages for microscopy can now achieve speeds approaching those of mirror-based systems. The stage documented in the additional parts list in table 5.2, for instance, has a nominal top scanning speed equivalent to  $4 \mu\text{s}$  per pixel. This is well within the range of scanning velocities generally used for these types of microscopies. Furthermore, it should be noted that it is possible to implement a DMD-based system using scanning mirrors. The scanned beams all overlap at the back aperture of the objective. It is possible to image this aperture multiple times along the detection path so that although the beam is moving, it arrives at the same location on the grating and DMD for all scanned angles. This requires the introduction of additional lenses, and is necessarily more alignment sensitive, but it is available as an option for laboratories which have scanning-mirror systems but may not have a scanning sample stage.

Alternatively, the multiplexed, and therefore CS, measurements can be made in space instead of spectrum. This is accomplished by introducing a cylindrical lens before the excitation objective as in figure 5.1c. This results in a line focus at the sample. One possible way to introduce the requisite cylindrical lens is through a lens tube adapter for the objective, depicted in figure 5.7). In this imaging mode, the grating could be removed, and the cylindrical lens will allow the DMD to select which spatial components from the line that imaged the sample are directed to the detector. A two dimensional image is formed by scanning the sample in the direction perpendicular to the long axis of the line focus. By employing as many patterns as there are pixels in the image, the image can be exactly reconstructed. However, again, as discussed below, it is possible to undersample and still obtain high-fidelity reconstructions of the spatial distribution of objects in the FOV, even with as low as 9% sampling.

In figure 5.1d, the modes from figure 5.1a and 5.1c are combined. A line-focus is used to sample information from an entire row of the FOV simultaneously. In addition, a grating is placed after the condenser, dispersing the spectral components of the light in the direction perpendicular to the long axis of line-focus. A spherical lens then focuses the light onto the DMD, allowing one dimension of the DMD to select spatial components for multiplexing, and the other dimension to select spectral components. This type of setup has previously been shown to work for spontaneous Raman imaging (Scotté et al., 2020). Note that the microscope can be constructed in this final configuration and still perform all the other imaging modes depending on the filters, or lack thereof,

displayed on the DMD. The most sensitive component of the system is the grating; however, in principle, even this optic can remain in place for "traditional" operation, so long as the DMD is set to direct all spectral and spatial components to the detector.



**Figure 5.1:** Multiplexed and compressed sensing imaging modes using a digital micromirror device and a point detector

(a) A DMD-based multiplexed SRS spectrometer. The sample remains stationary with a grating (G) placed after the condenser, and a short pass filter (SP) to filter the Stokes beam. The grating disperses the spectral components of the polychromatic pump beam. A cylindrical lens (CL) focuses the beam along the axis of spectral dispersion onto a DMD which is used to select various spectral components to send to the detector. After the DMD a lens is used to relay the selected components to the detector. (b) Similar optical setup to (a); however, the sample is scanned using a scanning stage. This allows for multiplexed microspectroscopy to be performed, retrieving a spectrum for each spatial pixel. (c) Line-scan compressed sensing. A cylindrical lens is placed before the excitation objective, producing a line-focus at the sample. After the condenser, a cylindrical lens refocuses the line on to the DMD which selects different spatial components to send to the detector. A relay lens is again used to direct the selected spatial components on to the detector. As the line-focus covers one axis of the field of view, the sample is only scanned in the perpendicular direction, in this case  $y$ . (d) A combination of (a) and (c). A line-focus is again used for multiplexed/compressed spatial sampling. A grating is placed after the condenser, however, spectrally dispersing the pump beam along the axis perpendicular to the line-focus. The dispersed beam is refocused using a spherical lens which allows the DMD to select both spatial and spectral components to send to the detector. One axis of the DMD selects spatial components, while the other axis selects spectral components for each spatial position in the sample.

## 5.2.2 Mathematical description of DMD-based CSSRS

We begin by considering the optical systems proposed in figure 5.1a and c. The system described by 5.1b requires the application of the equations for 5.1a to each spatial pixel in turn. Some slight modifications are required to describe the system in figure 5.1d, which are introduced later on. For each spectral point, if using the system in figure 5.1a, or each row of the FOV we scan through the line-focus, if using system 5.1c, a binary pattern is displayed on the DMD. As is typically done in spontaneous CS Raman, we choose the patterns by beginning with a Hadamard matrix. Hadamard matrices are orthogonal binary matrices with 1 and -1 as the entries. Hadamard matrices with dimensions equal to a power of 2 can easily be obtained using Sylvester's construction (figure 5.2a) (Steepleton 2019).

The binary nature of the Hadamard matrix simplifies its translation to an experimental setup involving a DMD, and its orthogonality is appealing mathematically as the inverse is equal to the transpose. However, since the entries of the matrix are 1 and -1, an experimental realization of selection based upon this matrix would require two detectors. One detector would acquire all the light corresponding to matrix entries of 1, while the other would acquire the light corresponding to matrix entries of -1. Subsequently, the total intensity detected at the second detector would be subtracted from the intensity detected at the first. Such a system is feasible to construct, and there is some evidence that it may be preferable to one that emulates a binary system of 0 and 1 (Sloane and Harwit, 1976), however, we will propose a modification of the Hadamard matrix which simplifies the experimental setup further. The first row and column of the Hadamard matrix produced using Sylvester's construction are removed, and entries with a value of 1 are converted to 0, while the entries with a value of -1 are converted to 1. The result is often termed the S-matrix, or  $S$  (figure 5.2b). Each row of  $S$  then becomes a DMD pattern for a different row of the FOV. For implementation of an SRS spectrometer, the governing equation then simply becomes:

$$\vec{y} = S \vec{x} \quad (5.1)$$

where  $\vec{y}$  are the measured multiplexed data points,  $S$  is the constructed S-matrix, and  $\vec{x}$  is the true SRS spectrum. Retrieval of the spectrum is performed by inversion of the S-matrix, for full sampling, or through solution of a minimization problem outlined below. Microspectroscopy is performed by solving the retrieval problem for each spatial pixel in turn.

Spatial multiplexing requires a slight modification to the equation above. Mathematically, the operation performed by a spatially multiplexed imaging system - applying each row of the matrix  $S$  to a corresponding row the FOV - is equivalent to a Hadamard product, denoted by  $\circ$ , between the matrix representing the sample intensity data,  $X$ , and the measurement matrix,  $S$ . This is followed by a sum along the rows, which can also be written as a matrix product from the right with a column vector of ones,  $\vec{\mathbb{1}}$ , as shown below.

$$\sum_j (S \circ X)_{.,j} = (S \circ X) \vec{\mathbb{1}} \quad (5.2)$$

Algorithms for image reconstruction in general do not employ Hadamard products, but rather traditional matrix products. This allows for the use of matrix decompositions, for which there are numerically stable algorithms, to find exact solutions. Furthermore, the  $n \times n$  2D data matrix can be more efficiently operated upon in vectorized form. I.e. as a 1D column vector,  $vec(X) = \vec{x}$ . This



operation is defined in the "C-style" format to stack the matrix in a row-centric way, i.e. row 1, above row 2, above row 3, and so on, as illustrated below. The alternative would be to stack the columns, which is equivalent to the former operation applied to the data matrix transpose.

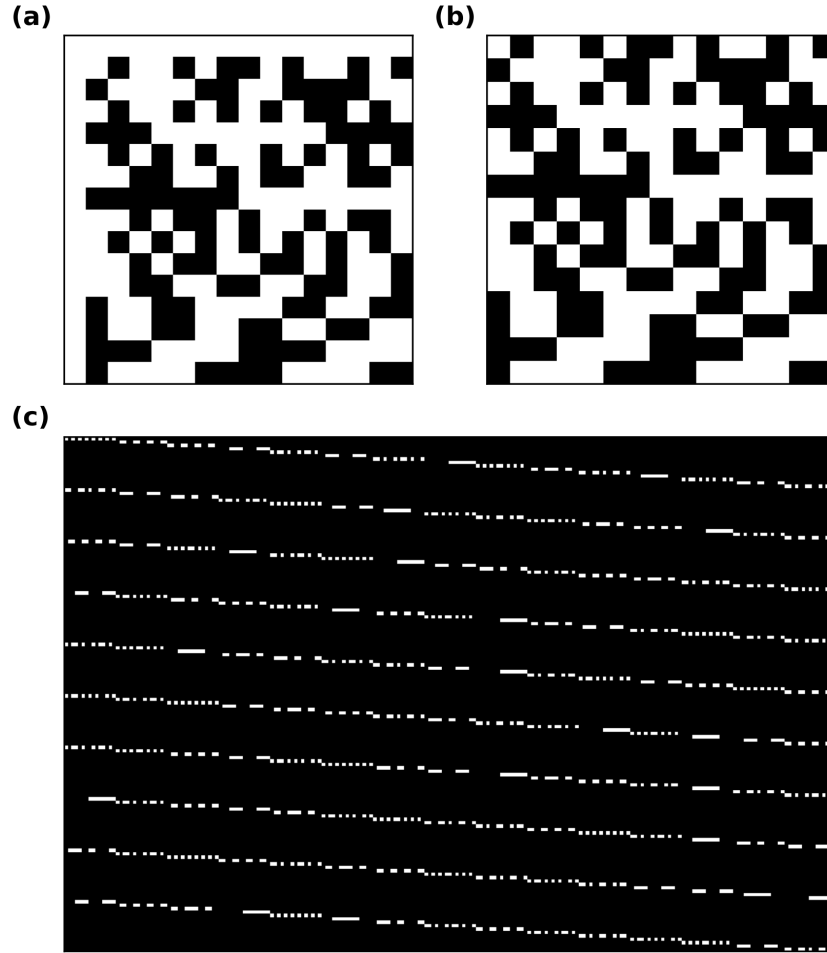
$$\text{vec}(X) = \begin{bmatrix} X_{1,\cdot} \\ X_{2,\cdot} \\ \vdots \\ X_{n,\cdot} \end{bmatrix} = \vec{x} \quad (5.3)$$

In order to work with vectorized data, and avoid the use of Hadamard products, we make use of the following transformation,  $\text{mat}(\cdot)$ , of our measurement matrix  $S$ . This operation produces a block matrix where the rows of the original matrix  $S$  are arrayed along the diagonal with zeros elsewhere.

$$\text{mat}(S) = \begin{bmatrix} S_{1,\cdot} & \vec{0}^{(n)} & \dots & \vec{0}^{(n)} \\ \vec{0}^{(n)} & S_{2,\cdot} & \dots & \vec{0}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{0}^{(n)} & \vec{0}^{(n)} & \dots & S_{n,\cdot} \end{bmatrix} \quad (5.4)$$

Left multiplication of this matrix with our data vector,  $\vec{x}$ , produces a single column vector consisting of the multiplexed row sums of our original data matrix. For the algorithms below, our full measurement matrix,  $M$ , consists of stacking the matrices of row permuted variants of  $S$  (created by the permutation matrix  $P$ ) produced by this transformation, as shown below. The final result is a,  $(k * n) \times n^2$  matrix, where  $k$  is the number of S-matrix permutations to use, and  $n$  is again the length of the rows and columns of the data matrix. Note that this introduces considerable computational overhead, as the scaling of operations increases from  $O(n^2)$  to  $O(n^4)$ . This ultimately leads to memory limitations with increasing image size at which point exact solutions of the matrix equations are no longer feasible and approximate solutions must be found instead, using algorithms such as conjugate gradients.

$$M = \begin{bmatrix} \text{mat}(S) \\ \text{mat}(P_1 S) \\ \vdots \\ \text{mat}(P_k S) \end{bmatrix} = \begin{bmatrix} S_{1,\cdot} & \vec{0}^{(n)} & \dots & \vec{0}^{(n)} \\ \vec{0}^{(n)} & S_{2,\cdot} & \dots & \vec{0}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{0}^{(n)} & \vec{0}^{(n)} & \dots & S_{n,\cdot} \\ (P_1 S)_{1,\cdot} & \vec{0}^{(n)} & \dots & \vec{0}^{(n)} \\ \vec{0}^{(n)} & (P_1 S)_{2,\cdot} & \dots & \vec{0}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{0}^{(n)} & \vec{0}^{(n)} & \dots & (P_1 S)_{n,\cdot} \\ \vdots & \vdots & \ddots & \vdots \\ (P_k S)_{1,\cdot} & \vec{0}^{(n)} & \dots & \vec{0}^{(n)} \\ \vec{0}^{(n)} & (P_k S)_{2,\cdot} & \dots & \vec{0}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{0}^{(n)} & \vec{0}^{(n)} & \dots & (P_k S)_{n,\cdot} \end{bmatrix} \quad (5.5)$$



**Figure 5.2:** Construction of a compressed sensing measurement matrix

(a) A 16x16 Hadamard matrix produced using Sylvester’s construction. (b) A 15x15 *S*-matrix created by removing the first row and first column of the Hadamard matrix in (a). As Hadamard matrices are binary matrices with the values of -1 and 1, the *S*-matrix converts all entries with a value of 1 to 0, and all entries with a value of -1 to 1. The purpose of this conversion is to simplify experimental realization of the corresponding measurement. A binary matrix of 0 and 1 detects entries where the value of the matrix equals 1 and ignores entries where the value equals 0. This requires the use of a single detector. Using a Hadamard matrix instead of an *S*-matrix would require an additional detector to detect entries corresponding to -1. Subsequently, the value recorded at this additional detector would need to be subtracted from the value recorded at the detector corresponding to matrix entries with a value of 1. (c) The final measurement matrix, *M*, produced using the procedure outlined in the methods. For computational ease the *S*-matrix is converted to a measurement matrix which operates on a vectorized data matrix. The measurement matrix in this panel shows the conversion of the *S*-matrix in (b) along with nine additional row permutations of that matrix. Experimentally, this corresponds to measuring 10 DMD patterns.

The measurements obtained, in vectorized form, can then be written by the simple matrix equation.

$$\vec{y} = M\vec{x} \quad (5.6)$$

The general problem of compressed sensing then seeks to minimize the number of non-zero coefficients of  $\vec{x}$ , i.e., find the sparsest possible solution, given the data,  $\vec{y}$ .

$$\min \|\vec{x}\|_0 \quad (5.7)$$

The minimization of the  $\ell_0$  pseudonorm, however, is a non-convex optimization problem, and NP-hard (Hyder and Mahata, 2009). It has been found that relaxing the problem to an  $\ell_1$  minimization problem generally approximates the desired sparse solutions and is far more computationally tractable using convex programming techniques (Hastie, Tibshirani, and Wainwright, 2015). The general problem for our system is then to find the sparsest possible solution representing the data,  $\vec{x}$ , through minimization of its  $\ell_1$  norm subject to a variety of possible constraints imposed by the observed data. There are a number of possible choices for constraints:

1. Equality constraints:

$$M\vec{x} = \vec{y}$$

2. Squared residuals:

$$\|M\vec{x} - \vec{y}\|_2^2 < \epsilon$$

3. The (modified) Dantzig selector:

$$\|M^{-1}(M\vec{x} - \vec{y})\|_\infty$$

Two things are important to note. First, that the data need only be sparse in some basis. I.e., the minimization can occur over  $\|\phi\vec{x}\|_1$  for an appropriate choice of basis transformation  $\phi$ . Second, the choice of constraint is important and depends on the structure of the data. The solution to the minimization problem under equality constraints can **exactly** recover the data; however, it requires that the data is piecewise constant and highly sparse in the chosen basis (Candes and Tao, 2006; Candes, Romberg, and Tao, 2006). On the other hand, squared residual minimization is more appropriate for recovery of data based on actual noisy observations (Candes, Romberg, and Tao, 2006 (2)). The Dantzig selector, and its modified version presented here, has been shown to be highly successful in a number of cases (Candes and Tao, 2007); however, most of the work involving it has been purely theoretical and involves the use of an orthogonal measurement matrix (Candes and Tao, 2007; Efron, Hastie, and Tibshirani, 2007; James, Radchenko, and Lv, 2009). In our proposed system, the measurement matrix is not orthogonal and the utility of this constraint remains to be seen.

### 5.2.3 Compressed spectrum reconstruction

We begin by first simulating a compressed reconstruction that might be obtained from the optical setup described by figure 5.1a. The use of a DMD in an SRS spectrometer has been previously demonstrated (Berto et al., 2017). The main purpose of this demonstration was to show the utility of performing multiplexed detection of a broadband SRS signal. As elaborated in chapter 2, if the measurements performed by the microscope are truly shot-noise limited then there should be no benefit in multiplexing vs acquiring a spectrum one spectral channel at a time. Berto et al. found

that there was indeed a SNR advantage in using multiplexing for their optical system. They also noted that the DMD spectrometer could be used to perform compressed sensing to increase acquisition speed. It should be noted that although there was an advantage in multiplexing compared to raster detection using the DMD, no comparison was made to the SNR achievable through spectral focusing. This multiplexed result is important for spectroscopy applications which frequently use single or multichannel detection which is analogous to the raster scanning demonstrated in the paper. However, for microscopy purposes, a comparison to spectral focusing is needed, as this is the predominant mechanism currently employed for spectral acquisition in this field. Such a comparison is dependent on many parameters, such as stretching factors, linearity of chirp, pulse power distributions and more. We reserve a full discussion for another publication.

To test the performance of a DMD-based spectrometer we used two test spectra: a sparse spectrum (figure 5.3a) and a dense spectrum (figure 5.3c). The former case is analogous to the type of spectrum which may be observed from a solvent, such as chloroform or benzene, in the C-H stretching region. These types of solvents typically contain one or two narrow linewidth modes over the measured frequency window. They are often used for calibration purposes, particularly for calibrating wavenumber vs interpulse delay when using a spectral focusing system. On the other hand, the dense spectrum is analogous to the type of spectrum which may be observed from a biomolecule in the C-H stretching region. The presence of multiple close-lying modes from larger proteins or lipid molecules results in a broad spectrum, with few gaps, covering 100-200  $\text{cm}^{-1}$ . See for example the reference spectra for bovine serum albumin (BSA) and oleic acid (OA) in figure 5.8.

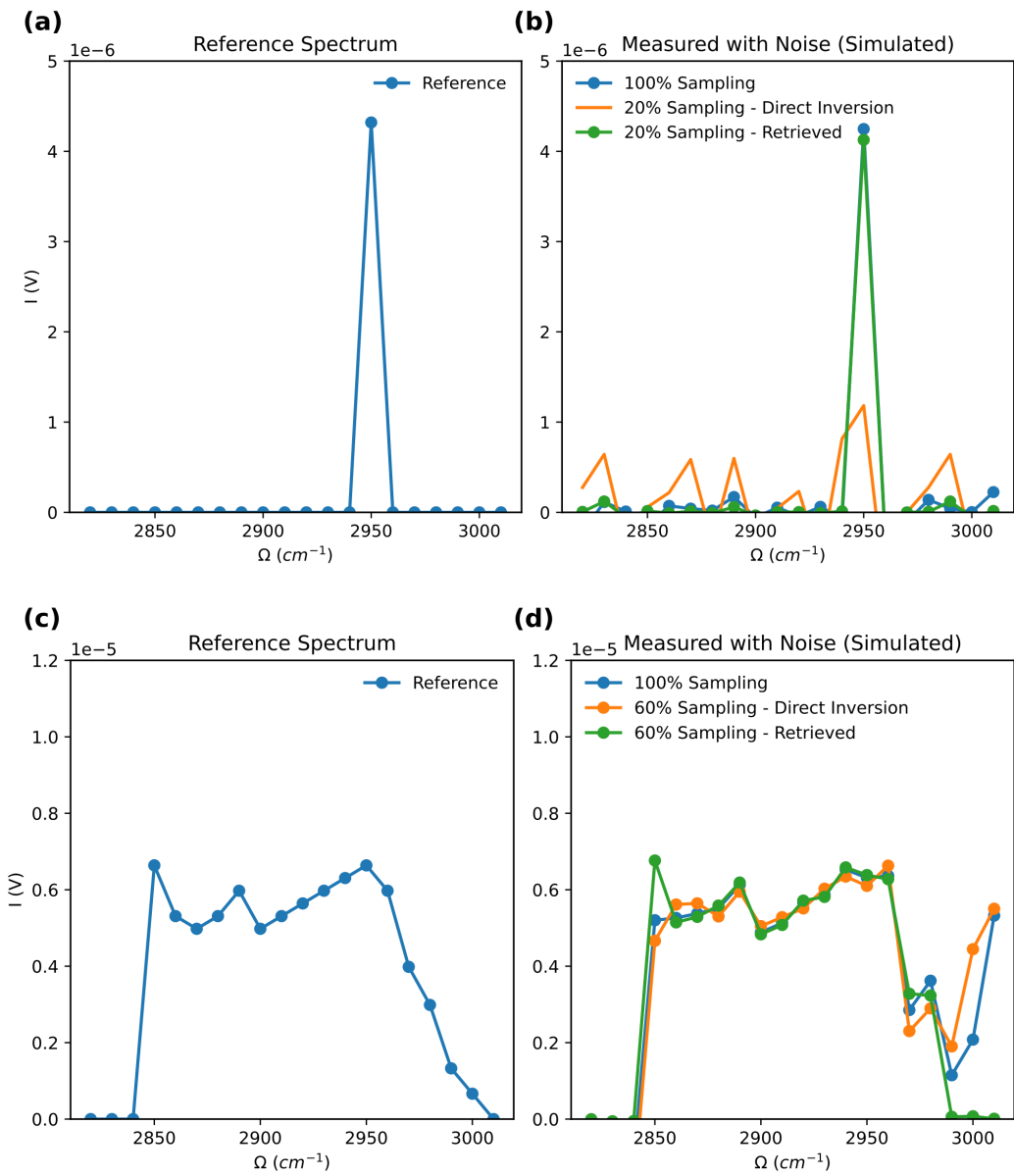
From these test spectra, consisting of 20 wavenumber channels, we sampled a variable number of DMD pattern equivalents under the noise model introduced in chapter 2, modelling a lock-in time constant of 3  $\mu\text{s}$  and a fourth order filter. 20 patterns is equivalent to full sampling, while fewer leads to compressed measurement. When fully sampled, it is possible to simply invert the measurement matrix to retrieve the spectrum; however, when under sampling, this procedure is not guaranteed to give accurate results (orange curves in figure 5.3b and d). Instead, we impose sparsity on the retrieved spectrum under quadratic constraints of the measured data, leading to the following optimization problem:

$$\begin{aligned} \min_x \quad & \|\vec{x}\|_1 \\ \text{s.t.} \quad & \|M\vec{x} - \vec{y}\|_2^2 \leq \epsilon \end{aligned} \tag{5.8}$$

We solve this minimization problem using a log-barrier algorithm after formulation as a second order conic problem (Chapter 11, Boyd and Vandenberghe, 2004). As expected, for the sparse spectrum, this procedure performs very well (figure 5.3b green curve). At 20% sampling (a five-fold increase in theoretical acquisition speed), direct inversion of the matrix (via pseudo-inverse as  $M$  is no longer square) is unable to retrieve the spectrum; however, under sparsity constraints the retrieved spectrum accurately represents the ground truth.

Interestingly, while the spectrum in figure 5.3c is not sparse, imposing sparsity constraints on the reconstruction of undersampled data accurately reproduced it (figure 5.3d green curve). In fact, the compressed reconstruction under sparsity actually more closely resembled the ground truth than the result of direct inversion with the fully sampled data (figure 5.3d blue curve). This is not anticipated to hold for every case, and in particular not for images. However, as discussed, the sparsity constraint does not need to be imposed on the data in the basis in which it was sampled,

but rather in any basis in which it is anticipated to be sparse. I.e., the data as you present it need not be sparse so long as there exists some representation of it which is.



**Figure 5.3:** Simulated DMD-based multiplexed and compressed sensing retrieval for two test spectra

**(a-b)** A sparse reference spectrum (a) and the retrieved result using simulated DMD-based multiplexed or compressed sensing detection (b). The measurement was simulated under the noise model introduced in the text. The sparse reference spectrum corresponds to the case of a single sharp Raman peak over the measured frequency window. For the C-H window this scenario is encountered for solvents which are frequently used as analytical standards in SRS microscopy. 100% sampling (blue curve in (b)) corresponds to a multiplexed measurement of 20 DMD patterns, one for each of the 20 spectral channels. 20% sampling corresponds to a compressed sensing measurement where only 4 DMD patterns are used. Using direct inversion of the corresponding measurement matrix it is impossible to reconstruct the spectrum (orange curve in (b)); however, imposing a sparsity on the retrieved spectrum under equality constraints of the data accurately reproduces the reference (green curve in (b)). **(c-d)** Analogously to (a-b), a reference spectrum (c) and the retrieved result using simulated DMD-based multiplexed or compressed sensing detection (d). In this case, however, the reference spectrum is dense, i.e., relatively few of the measured spectral channels have zero intensity. This scenario is frequently encountered with biomolecule spectra in the C-H window. See, for instance, figure 5.8 for measured SRS spectra of proteins and lipids. Surprisingly, both direct inversion (orange curve in (d)) and inversion under the condition of a sparse spectrum (green curve in (d)) produce fairly accurate results. Note that greater sampling is required (12 DMD patterns vs 4 for the sparse case) to accurately retrieve the spectrum.

#### 5.2.4 *in silico* image reconstruction from spectral focusing data

We next moved towards simulation of the setup described by figure 5.1d. In this imaging system, one axis of the DMD is used to sample the spatial components of the sample. This is achieved by producing a line-focus at the sample plane through the use of cylindrical optics. The other axis of the DMD encodes the spectral information. This is achieved by using a polychromatic (e.g. femtosecond) pump beam at the sample, which is then dispersed by a grating after the SRS interaction. The DMD then selects which spectral components are to be sampled, in addition to which spatial components. We furthermore adopt the strategy implemented for spontaneous Raman scattering in (Scotté et al., 2020). Here, instead of attempting to reconstruct an unknown spectrum, we simplify our problem into the decomposition of the image into concentrations of a number of species using known reference spectra. This is a problem which is frequently encountered in the field of SRS microscopy. For instance, for creation of false-color histology images (Sarri et al., 2019; Hollon et al., 2020), reference spectra of protein and lipid (figure 5.8) are frequently used to estimate their respective concentrations.

The use of the DMD to encode both spatial and spectral information requires the following modification to our governing equation.

$$Y = MC\lambda F \tag{5.9}$$

$M$  is the same spatial/spectral measurement matrix previously introduced above; however,  $\vec{x}$ , a column vector of the image data, is replaced by  $C$ . This is a matrix of  $n^2$  rows and  $m$  columns, where  $m$  is the number of species being measured. I.e., each column of  $C$  represents the vectorized spatial distribution of concentrations of a single species we wish to retrieve.  $\lambda$  is a matrix where

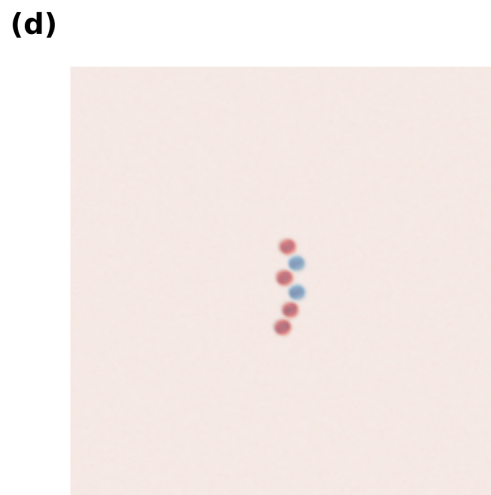
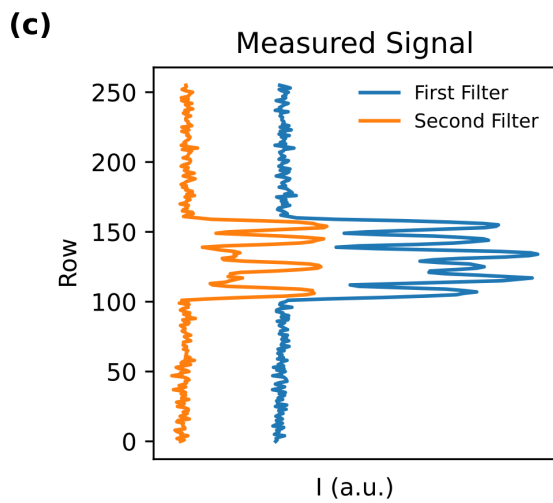
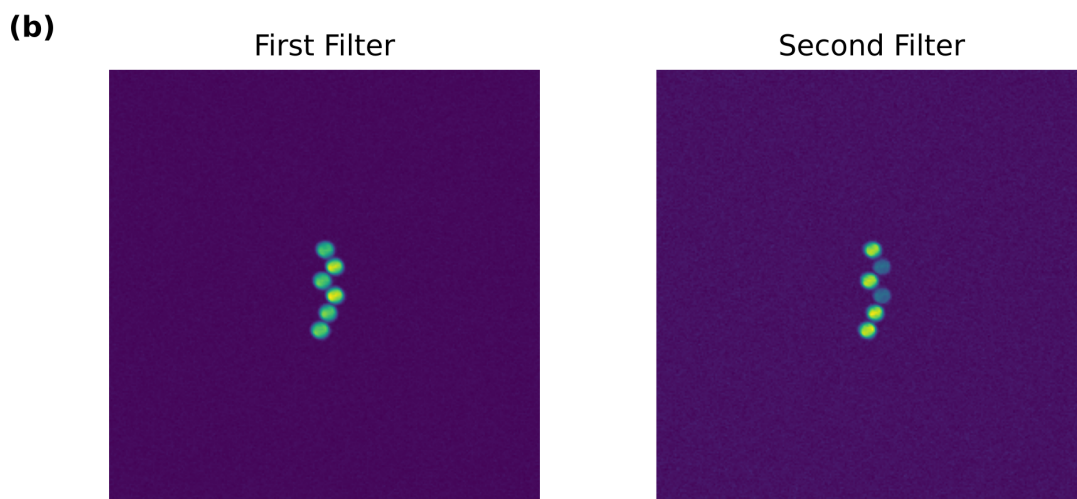
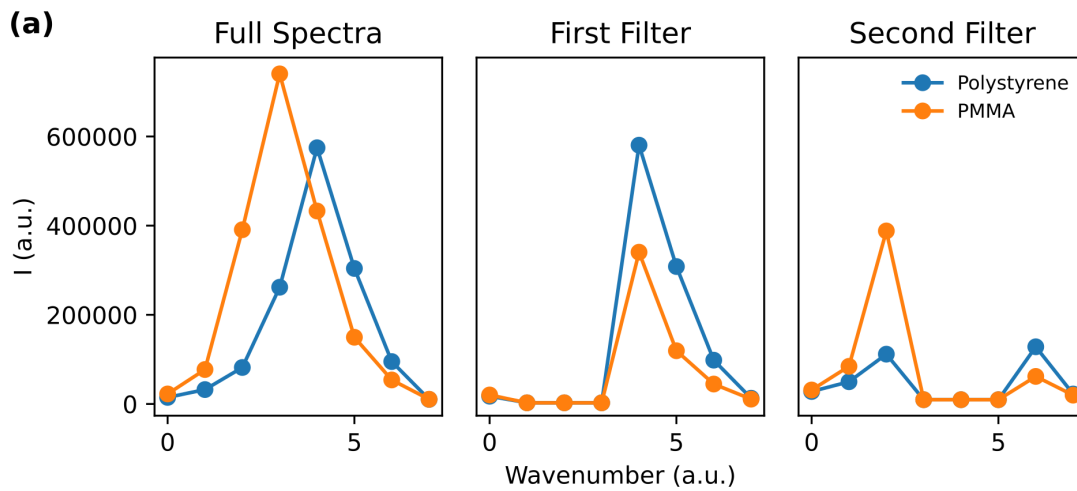
each row is one known reference spectrum for the corresponding column of  $C$ , and  $F$  is a matrix of spectral filters. Experimentally, the DMD patterns would be determined by the tensor product of  $M$  and  $F$ . For each spectral filter, a given number of spatial filters (the  $S$  matrix and its row permutations which determine  $M$ ) would be acquired.

As proposed in Scotté et al. the image is then reconstructed using a two-step procedure. First, an expectation maximization (EM) algorithm (Moon, 1996) is used to estimate  $C\lambda F$ , whose elements represent the estimated pixel intensities under application of each spectral filter. The species concentrations are then estimated through application of the pseudoinverse of  $(\lambda F)$  to the result of the first step. The procedure is iteratively performed in a line by line fashion to reconstruct the full 2D image.

For a test image, we began from hyperspectral data acquired using the SRS microscope and spectral focusing. A 256x256 image was acquired of a mixture of 10  $\mu\text{m}$  diameter poly(methyl methacrylate) (PMMA) and polystyrene beads for seven spectral channels in the C-H stretching window. The FOV contained four PMMA and two polystyrene beads. Over these seven channels, the two types of beads are distinguishable via their spectra (figure 5.4a left). Using these spectra, two binary filters were created to select different spectral components. Their application to the reference spectra is shown in the middle and rightmost panel of figure 5.4a. Note that while there are procedures for generating filters which maximally distinguish between two species, these particular filters were chosen ad hoc based on the ability to visually distinguish the plastics.

To simulate the DMD-based measurements, each spectral filter was applied along the spectral axis of the 256x256x7 hyperspectral cube. The cube was then summed along this axis, producing the two reference images in figure 5.4b. The first row and column of the resultant images were removed to align with the dimensions of the measurement matrix  $M$ . The final measured signals were created by applying the  $M$  matrix to each of these two spectrally filtered images in turn, producing traces as shown in figure 5.4c. Note that while this test data is constructed from actual SRS measurements, the SNR profile of the spectral focusing measurement does not reflect the actual SNR of a DMD-based measurement. As mentioned above, the relationship between the SNR in these two imaging modality is dependent on multiple factors and is reserved for future work.

Finally, using a total of 51 spatial masks, in combination with the two spectral filters (for a total of 102 equivalent measurements), the data was reconstructed into a two channel image where red was used for PMMA and blue was used for polystyrene. At 10% sampling (or equivalently, a 10-fold increase in anticipated acquisition speed), the image reliably represented the ground truth. Upon close examination it becomes clear that there was some misidentification of plastics, i.e., there is some blue visible within the PMMA beads; however, the overall interpretation of the result is not affected. The overall performance could likely be improved through optimization of the spectral filters.





**Figure 5.4:** *in silico* compressed reconstruction of plastic beads from real SRS data

(a) The measured spectra (left) of polystyrene and PMMA plastic beads over 7 evenly spaced wavenumbers in the C-H stretching window. Two binary spectral filters were applied for compressed sample reconstruction and are shown as the middle and rightmost panels. A value of 1 meant that spectral component was detected, and a value of zero meant it was ignored. (b) The fully spatially sampled data under application of the two spectral filters in (a). A seven channel hyperspectral stack was acquired of a single field of view (FOV) containing four PMMA and two polystyrene beads. The two images were constructed by multiplication along the spectral dimension by the same binary filters which produced the filtered spectra in (a). (c) An example of sparsely sampled row sums for each of the two spectral filters. (d) Retrieved two-component image from a 10% sampling of pixels. A total of 51 patterns, producing traces as in (c), were used for the reconstruction.

### 5.2.5 Extending the model to general images

While the EM algorithm performed well for the above sample, showing its utility even in the presence of actual observed experimental noise, we found that for certain test cases it was unable to produce a reasonable solution at similar compression rates (figure 5.6c-d). The particular structure of the measurement is a likely contributor to its capability, and the reconstruction fidelity as a function of sampling and measurement distributions will be the subject of future investigation.

Nonetheless, it is worth considering whether a different approach, or algorithm, may perform better for the types of samples that are expected to be encountered in an actual biological experiment. In particular, two types of test samples, representing simplifications of particularly common observations (figure 5.6a-b), that were met with varying degrees of success were the presence of multiple objects along the axis of summation, and the presence of nested objects. Both of these test cases are common in biological microscopy. The former occurs frequently as the placement of multiple objects (e.g. cells) can not generally be finely controlled, and the latter occurs because biological structures contain internal information. To be a useful technique, information such as the subcellular structure and organization (e.g. nucleus/organelle distribution) must be accurately reconstructed.

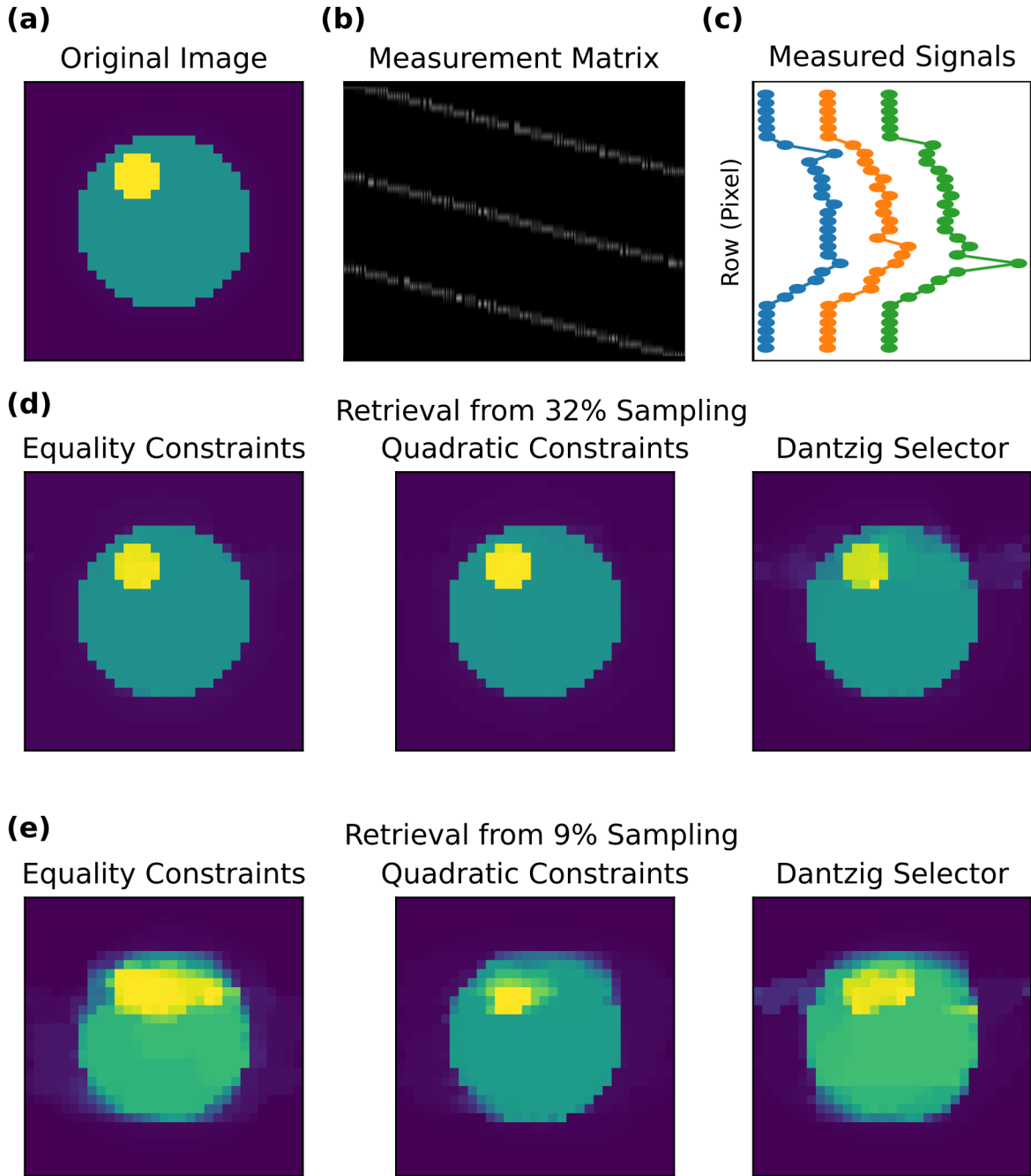
As mentioned above, one of the most surprising, and useful discoveries regarding compressed sensing and sparse retrieval is that the data need only be sparse in some basis. I.e., the minimization problem in equation 5.8, can be modified as follows:

$$\begin{aligned} \min_x \quad & \|\phi\vec{x}\|_1 \\ \text{s.t.} \quad & \|M\vec{x} - \vec{y}\|_2^2 \leq \epsilon \end{aligned} \tag{5.10}$$

for some change of basis operator  $\phi$ . Note that the additional data constraint need not be the square residuals, but can be one of the other options discussed above, for example. One common choice for  $\phi$  is the gradient operator  $\nabla$ . Employing the gradient operator, this problem is frequently referred to as total variation minimization. For fully sampled data, the total variation minimization problem is often used for denoising (Rudin, Osher, and Fatemi, 1992; Esedoglu and Osher, 2004). This procedure is also known as the Rudin-Osher-Fatemi model, after its inventors, and is generally very successful. This is because noise is typically high frequency compared to the information of interest in the image. Rapid fluctuations in intensity, and thus in the image gradient, can be

removed by imposing sparsity on the gradient.

The use of total variation minimization has also been applied widely in compressed sensing (Krahmer, Kruschel, and Sandbichler, 2017). We tested the applicability of the procedure under the three constraint models discussed above for a variety of test images. The results for one such test image are shown in figure 5.5. Using modified versions of the interior point methods presented in (Candes and Romberg, 2005), derived from (Chapter 11, Boyd and Vandenberghe, 2004), reliable reconstruction of a dense object containing internal structure could be achieved. With under 10% sampling the reconstructed image accurately represented the ground truth (figure 5.5e), and near-exact reconstructions were obtained for 32% sampling under certain constraints (figure 5.5d). Together these observations suggest that these algorithms may be appropriate for CSSRS. An additional reconstruction using 9% sampling under quadratic constraints, for a different test image, is presented in figure 5.6f.



**Figure 5.5:** Sparsity imposed total variation retrieval of a test image under different constraints of the data

(a) The reference 31x31 test image. The color scale goes from purple to yellow, with yellow representing the highest pixel intensity. (b) The measurement matrix constructed as described in the text. The matrix is displayed for 3 DMD sampling patterns. (c) Example traces of the measured data as would be detected for the measurement matrix in (b). The measurement vector is displayed as three offset traces as opposed to the single vectorized result described by the equations in the text to highlight the row sums with respect to the reference image in (a). Note that depending on the measurement matrix, the measured row sums with the largest values may not correspond to the rows in the original image with the greatest total intensity sum. (d) Sparse retrieval under equality (left), quadratic (middle), and the Dantzig selector (right) constraints, given 32% sampling. This level of sampling corresponds to 10 DMD patterns for a 31x31 image, or, equivalently, 310 out of 961 pixels. (e) The sparse retrieval as in (d) except for 9% sampling of the data. This level of sampling corresponds to 3 DMD patterns for a 31x31 image, or, equivalently, 93 out of 961 pixels.

## 5.3 Discussion

SRS microscopy has many attractive qualities. Spontaneous Raman, and therefore SRS, spectra are chemically specific. The SRS signal is linear in concentration, making accurate concentration quantification feasible (Oh et al., 2022). Signals are (bi-)linear in excitation power. Furthermore, the orders of magnitude increase in signal of the SRS process compared to spontaneous Raman scattering means it has allowed for its application in microscopy to reach acquisition speeds in the range of standard confocal fluorescence scanning microscopes (McCamant, Kukura, and Mathies, 2003; Freudiger et al., 2008).

Nonetheless, in its current form, SRS microscopy for unsuitable to many problems due to a need for still greater speed and higher sensitivity. Many low-concentration biomolecules cannot be reliably detected with SRS, and dynamic processes are still challenging to capture as video-rate SRS hardware is not commonplace. Many strategies exist to improve sensitivity, although they tend to sacrifice experimental simplicity in the pursuit. The introduction of additional dyes or nanoparticles allows for the detection of molecules in the  $\mu\text{M}$  or even nM ranges, but it sacrifices the label-free aspect of SRS microscopy often cited as one of its main advantages (Wei and Min, 2018; Shi et al., 2022). On the other hand, methods for increasing acquisition speed, in the absence of these sensitivity boosts, are scarce.

The techniques of compressed sensing have exploded in popularity over the last ten years and have been highly successful in the fields to which they have been applied (Yoon et al., 2018). Applications include the aforementioned work in the field of spontaneous Raman microscopy, as well as the development of single pixel camera techniques (Chan et al., 2008; Duarte et al., 2008; Edgar, Gibson, and Padgett, 2019). Another body of work has even been aimed at the extension of compressed sensing techniques to quadratic problems (Ohlsson et al., 2012; Bahmani, and Romberg, 2015). While much of the literature has focused on the application to linear problems, such as the one we have just described, many important problems include quadratic terms. Examples include phase retrieval problems, such as those encountered in X-ray crystallography or optical pulse characterization, such as by using the FROG technique covered in chapter 2.

The work presented in this chapter suggests that CSSRS may be a viable method for increasing

acquisition speed in SRS microscopy. Use of these techniques lowers the barrier to entry compared to custom solutions for rapid image acquisition that have been previously proposed, such as resonant circuit arrays (Liao et al., 2015), or high frequency wavelength tuning laser systems (He et al., 2017). This makes it possible to increase experimental throughput, and applications such as cytometry start to become more widely realizable. Additionally, the multiplexing process inherent to CSSRS has been demonstrated to improve the SNR for SRS measurements made with certain optical systems (Berto et al., 2017). Even for microscopes operating in a shot-noise limited regime regularly, the simple, and highly optimistic, noise model introduced in chapter 2, suggests that low-power measurements are likely to be affected by thermal noise, and can thus also receive a SNR benefit from multiplexing.

Furthermore, with CSSRS it may be possible to reduce phototoxicity without sacrificing acquisition speed. If it remains true across cell lines that phototoxic effects are more apparent from greater excitation power than acquisition time, as suggested in (Zhang et al., 2022), the optical powers employed in CSSRS can be lowered in favor of longer per-pixel exposure times. The total acquisition time need not be affected in CSSRS due to the undersampling allowing for fewer pixels to be acquired.

Finally, this strategy can benefit from the continued development of high-frequency amplified lasers. Fiber-optic systems operating at MHz frequencies with mJ pulse energies are now available (Zhao and Kobayashi, 2016; Seidel et al., 2022). Systems such as these provide enough energy to disperse the pulse over a large area, as required by this compressed sensing strategy, while still allowing for modulation frequencies far from  $1/f$  laser noise. In fact, such a system could be used to sample the full field of view, removing the need for scanning electronics entirely. The traditional difficulty in producing a widefield SRS microscope is the reliance on high-frequency modulation and lock-in detection. With CSSRS, provided there is ample excitation power, a single point detector can be used in combination with widefield sample illumination and a DMD to reconstruct the image.

There are of course still drawbacks to using CSSRS. CSSRS as implemented in this chapter makes use of the grating and DMD on the detection side; however, this could be viewed as a rather wasteful design. In the current setup, given the nature of the Hadamard matrix, and its derived S-matrix, half the power used for sample illumination never sees the detector. This "wasted" power, however, is still deposited on the sample. If instead, the multiplexing were to occur on the excitation side, all of the power used to illuminate the sample would also be detected, which could have further benefits in limiting phototoxicity. The same SNR could in principle be achieved with half the excitation power seen by the sample. Furthermore, as mentioned previously, the matrices in the governing equations of CSSRS start to become too large to work with for larger images, requiring approximate methods to find solutions to the optimization problems.

Nonetheless, CSSRS provides an attractive alternative to traditional SRS microscopy, especially in situations where rapid acquisition speed or low excitation powers are necessary. The continued theory and algorithm development in the field of compressed sensing promises to lead to further improvements in the applicability of this technique to problems in the life sciences.

## 5.4 Additional methods

### 5.4.1 Spectrum and image retrieval

A total of seven problems, and corresponding algorithms, were considered for this work. All implementations were written in Python 3.10, and make use of the NumPy and SciPy packages. The first algorithm addressed was the expectation-maximization algorithm used for the bead reconstruction presented in the results. This was implemented in the fashion of reports (Moon, 1992). Note that this algorithm was implemented in a line-by-line fashion for image reconstruction. There are two important consequences of this implementation choice. First, it can be applied without modification to images of a large size. This is because by addressing each row independently, the measurement matrix (and/or change of basis matrices) can be explicitly constructed by considering only the portion of the total matrix that addresses each row. Given their size for larger images (e.g. 255x255 or 511x511 pixels), the entire matrix is too big to be explicitly constructed and fit in memory. E.g. for a 511x511 image, the corresponding measurement matrix under full sampling ( $511^2 \times 511^2$ ) requires over 500 GB of memory in Python (64 bit integers). Even with improved memory management in C/C++ over 65 GB would be required. The second implication of implementing the algorithm in this fashion is that it treats each row of the image as independent. This means that it cannot benefit from structure between rows, which can help constrain the solution. For this reason, the additional six algorithms were considered.

These other algorithms can be categorized into two general categories: two algorithms which are recast as linear programming problems and are solved by a primal-dual algorithm, and four algorithms which are recast as second order cone problems and recovered using a log-barrier method. All implementations follow the general methods introduced by (Boyd, and Vandenberghe, 2004; Candes, and Romberg, 2005). The optimization procedure employs Newton’s method, where the difference between the six algorithms resides in the specific form of the Newton step. While included to in the text, for completeness, the six minimization problems are explicitly presented here.

#### Linear problems:

---

$$\begin{aligned} \min_x \quad & \|\vec{x}\|_1 \\ \text{s.t.} \quad & M\vec{x} = \vec{y} \end{aligned}$$

---

$$\begin{aligned} \min_x \quad & \|\vec{x}\|_1 \\ \text{s.t.} \quad & \|M^{-1}(M\vec{x} - \vec{y})\|_\infty \leq \epsilon \end{aligned}$$

---

#### Second-order conics:

---

$$\begin{aligned} \min_x \quad & \|\vec{x}\|_1 \\ \text{s.t.} \quad & \|M\vec{x} - \vec{y}\|_2^2 \leq \epsilon \end{aligned}$$

---

$$\begin{aligned} \min_x \quad & \|\nabla \vec{x}\|_1 \\ \text{s.t.} \quad & M\vec{x} = \vec{y} \end{aligned}$$

---

$$\begin{aligned} \min_x \quad & \|\nabla \vec{x}\|_1 \\ \text{s.t.} \quad & \|M\vec{x} - \vec{y}\|_2^2 \leq \epsilon \end{aligned}$$

---


$$\begin{aligned} \min_x \quad & \|\nabla \vec{x}\|_1 \\ \text{s.t.} \quad & \|M^{-1}(M\vec{x} - \vec{y})\|_\infty \leq \epsilon \end{aligned}$$


---

For large images, the work in (Boyd, and Vandenberghe, 2004) and (Candes, and Romberg, 2005) suggests the use of the conjugate gradients algorithm after the work presented in (Shechuk, 1994). However, while this algorithm works well, and can avoid an explicit construction of the measurement matrix (or change of basis matrix), it still requires definition of a function which performs the equivalent operation the linear operator does. More importantly, it also requires an accurate function for the inverse operation. While it is easy to define a function that performs the forward operation (sparse sampling and row summation), the reverse operation is hard to formulate. I.e., there is no easily definable function that takes data as measured by the imaging system and returns the data in its original basis. This procedure works well for examples such as sparse-sampling in the Fourier domain, as the reverse operation can easily be defined as the inverse Fourier transform; however, for the specific type of operation performed by this DMD-based imaging system we were unable to effectively implement the algorithm. Investigation into alternatives remains on-going.

As a final note, the implementations of the Dantzig selector were modified where appropriate for explicit inclusion of the inverse. For non-square measurement matrices (i.e., when performing compressed sensing) the pseudo-inverse was substituted. As mentioned in the main text, the mathematical properties of this algorithm have not been rigorously investigated for non-orthogonal design (measurement) matrices. The results of the numerical simulations presented above suggest that it can be used. At low compression rates, the retrieval is worse than the retrieval under equality or quadratic constraints; however, at higher rates, it performs remarkably well.

## 5.4.2 SRS imaging of plastic beads

SRS images were acquired at seven wavenumbers in the C-H stretching window using the microscope described in chapter 2. A 60x 1.2 NA water immersion objective (Olympus) was used for excitation, and a 1.4 NA oil immersion condenser (Thorlabs) was used to collect the transmitted light. The Stokes beam was intensity modulated at 10.28 MHz using a function generator, power amplifier, and quarter waveplate with a resonant EOM (Thorlabs), followed by a polarizer. The SRL signal on the pump was detected using a photodiode (Hamamatsu) with a home-built detection circuit, and a lock-in amplifier (Zurich Instruments). The pump and Stokes pulses were stretched to 1.9 ps and 1.7 ps, respectively, using the grating stretchers depicted in figure 2.1. The average power of both the pump and Stokes beam was 10 mW. The Stokes beam has a fixed wavelength of 1040 nm, while the pump beam was tuned to a center wavelength of 799 nm. Under these conditions, images were acquired with an approximate resolution of 10-15  $\text{cm}^{-1}$  over a frequency window spanning 2820-2980  $\text{cm}^{-1}$ . A pixel dwell time of 10  $\mu\text{s}$  was used in conjunction with a lock-in time constant of 3  $\mu\text{s}$  with a fourth-order low-pass filter.

Polystyrene (PS) and poly(methyl methacrylate) (PMMA) beads with a 10  $\mu\text{m}$  diameter (both from Sigma Aldrich) were mixed in a 50/50 ratio in water. Approximately 100  $\mu\text{L}$  of the solution was mounted onto a coverslip surrounded by Scotch magic tape. A second coverslip was placed

above it to allow for the immersion fluid of the oil condenser.

## 5.5 References

Bahmani, S., and Romberg, J. (2015) “Efficient compressive phase retrieval with constrained sensing vectors”, *Advances in Neural Information Processing Systems*, Vol. 28.

Boyd, S., and Vandenberghe, L. (2004), “*Convex Optimization*”. Cambridge University Press, 2004

Berto, P., Scottee, C., Galland, G., Rigneault, H., and de Aguiar, H. B. (2017), “Programmable single-pixel-based broadband stimulated Raman scattering”, *Optics Letters*, Vol. 42 No. 9, pp. 1696-1699.

Candes, E., and Romberg, J., (2005), “l1-magic: Recovery of sparse signals via convex programming”, <https://statweb.stanford.edu/candes/l1magic/>

Candes, E. J., Romberg, J., and Tao, T. (2006), “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information”, *IEEE Transactions on Information Theory*, Vol. 52 No. 2.

Candes, E., and Tao, T. (2006), “Near-optimal signal recovery from random projections and universal encoding strategies”, *IEEE Transactions on Information Theory*, Vol. 52 No. 12.

Candes, E., Romberg, J. K., and Tao, T. (2006), “Stable signal recovery from incomplete and inaccurate measurements”, *Communications on Pure and Applied Mathematics*, Vol. 59 No. 8, pp. 1207-1223.

Candes, E., and Tao, T. (2007), “The Dantzig selector: Statistical estimation when  $p$  is much larger than  $n$ ”, *Ann. Statist.*, Vol 35 No. 6, 2313-2351.

Chan, W. L., Charan, K., Takhar, D., Kelly, K. F., Baraniuk, R. G., and Mittleman, D. M. (2008) “A single-pixel terahertz imaging system based on compressed sensing”, *Applied Physics Letters*, Vol. 93 No. 12.

Davis, B. M., Hemphill, A. J., Maltas, D. C., Zipper, M. A., Wang, P., and Ben-Amotz, D. (2011), “Multivariate hyperspectral Raman imaging using compressive detection”, *Anal. Chem.*, Vol. 83 No. 13, pp. 5086-5092.

Donoho, D. L. (2006), “Compressed sensing”, *IEEE Transactions on Information Theory*, Vol. 52 No. 4.

Duarte, M. F., Davenport, M. A., Takhar, D., Laska, J. N., Sun, T., Kelly, K. F. and Baraniuk, R. G. (2008) “Single-pixel imaging via compressive sampling”, *IEEE Signal Processing Maga-*



zine, Vol. 25 No. 2, pp. 83-91.

Edgar, M. P., Gibson, G. M., Padgett, M. J. (2019) “Principles and prospects for single-pixel imaging”, *Nature Photonics*, Vol. 13, pp. 13-20.

Efron, B., Hastie, T., Tibshirani, R. (2007), “Discussion: The Dantzig selector: statistical estimation when  $p$  is much larger than  $n$ ”, *The Annals of Statistics*, Vol. 35 No. 6, 2358-2364.

Einstein, E. H., Ablyazova, F., Rosenberg, A., Harshan, M., Wahl, S., Har-El, G., Constantino, P. D., Ellis, J. A., Boockvar, J. A., Langer, D. J., and D’Amico, R. S. (2022), “Stimulated Raman histology facilitates accurate diagnosis of neurosurgical patients: a one-to-one noninferiority study”, *Journal of Neuro-Oncology*, Vol. 159, pp. 369-375.

Esedoglu, S., and Osher, S. J. (2004), “Decomposition of images by the anisotropic Rudin-Osher-Fatemi model”, *Communications on Pure and Applied Mathematics*, Vol. 57 No. 12, 1609-1626.

Freudiger, C. W., Min, W., Saar, B. G., Lu, S., Holtom, G. R., Chengwei, H., Tsai, J. C., Kang, J. X., and Xie, X. S. (2008), “Label-free biomedical imaging with high sensitivity by stimulated Raman scattering microscopy”, Vol. 322 No. 5909, pp. 1857-1861.

Hastie, T., Tibshirani, R., Wainwright, M. (2015), “Statistical learning with sparsity”, *Monographs on statistics and applied probability*, Vol. 143.

He, R., Liu, Z., Xu, Y., Huang, W., Ma, H., Ji, M. (2017), “Stimulated Raman scattering microscopy and spectroscopy with a rapid scanning optical delay line”, *Optics Letters*, Vol. 42 No. 4, pp. 659-662.

Hill, A. H., and Fu, D. (2019), “Cellular imaging using stimulated Raman scattering microscopy”, *Anal. Chem.*, Vol. 15, pp. 9333-9342.

Hollon, T. C., Pandian, B., Adapa, A. R., Urias, E., Save, A. V., Khalsa, S. S. S., Eichberg, D. G., D’Amico, R. S., Farooq, Z. U., Lewis, S., Petridis, P. D., Marie, T., Shah, A. H., Garton, H. J. L., Maher, C. O., Heth, J. A., McKean, E. L., Sullivan, S. E., Hervey-Jumper, S. L., Patil, P. G., Thompson, B. G., Sagher, O., McKhann II, G. M., Komotar, R. J., Ivan, M. E., Snuderl, M., Otten, M. L., Johnson, T. D., Sisti, M. B., Bruce, J. N., Muraszko, K. M., Trautman, J., Freudiger, C. W., Canoll, P., Lee, H., Camelo-Piragua, S., and Orringer, D. A. (2020), “Near real-time intraoperative brain tumor diagnosis using stimulated Raman histology and deep neural networks”, *Nature Medicine*, Vol. 26, pp. 52-58.

Hu, F., Shi, L., and Min, W. (2019), “Biological imaging of chemical bonds by stimulated Raman scattering microscopy”, *Nature Methods*, Vol. 16, pp. 830-842.

Hyder, M., and Mahata, K. (2009), “An approximate  $L_0$  norm minimization algorithm for compressed sensing”, *IEEE International Conference on Acoustics, Speech and Signal Processing*.

- James, G. M., Radchenko, P., and Lv, J. (2009), “DASSO: connections between the Dantzig selector and lasso”, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 71 No. 1, pp. 127-142.
- Krahmer, F., Kruschel, C., and Sandbichler, M. (2017), “Total variation minimization in compressed sensing”, In: Boche, H., Caire, G., Calderbank, R., Marz, M., Kutyniok, G., and Mathar, R. (eds) “Compressed sensing and its Applications”, *Applied and Numerical Harmonic Analysis*. Birkhauser, Cham.
- Liao, C-S., Slipchenko, M. N., Wang, P., Li, J., Lee, S-Y., Oglesbee, R. A., and Cheng, J-X. (2015) “Microsecond scale vibrational spectroscopic imaging by multiplex stimulated Raman scattering microscopy”, *Light: Sciences & Applications*, Vol. 4 No. e265.
- Lin, H., Liao, C., Wang, P., Huang, K., Bouman, C. A., Kong, N., and Cheng, J.-X. (2017), “Sparsely-sampled hyperspectral stimulated Raman scattering microscopy: A theoretical investigation”, *Proc. of SPIE*, 10069.
- Lin, H., Liao, C-S., Wang, P., Kong, N., and Cheng, J-X. (2018), “Spectroscopic stimulated Raman scattering imaging of highly dynamic specimens through matrix completion”, *Light: Science & Applications*, Vol. 7 No. 17179.
- McCamant, D. W., Kukura, P., and Mathies R. A. (2003) “Femtosecond broadband stimulated Raman: A new approach for high-performance vibrational spectroscopy” *Appl. Spectrosc.*, Vol. 57, pp. 1317-1323.
- Moon, T. K. (1996), “The expectation-maximization algorithm”, *IEEE Signal Processing Magazine*, Vol. 13 No. 6, pp. 47-60.
- Nao, N., Iino, T., Isozaki, A., Yamagishi, M., Kitahama, Y., Sakuma, S., Suzuki, Y., Tezuka, H., Oikawa, M., Arai, F., Asai, T., Deng, D., Fukuzawa, H., Hase, M., Hasunuma, T., Hayakawa, T., Hiraki, K., Kotaro, H., Hoshino, Y., Inaba, M., Inoue, Y., Ito, T., Kajikawa, M., Karakawa, H., Kasai, Y., Kato, Y., Kobayashi, H., Lei, C., Matsusaka, S., Mikami, H., Nakagawa, A., Numata, K., Ota, T., Sekiya, T., Shiba, K., Shirasaki, Y., Suzuki, N., Tanaka, S., Ueno, S., Watarai, H., Yamano, T., Yazawa, M., Yonamine, Y., Di Carlo, D., Hosokawa, Y., Uemura S., Sugimura, T., Ozeki, Y., and Goda, K. (2020), “Raman image-activated cell sorting”, *Nature Communications*, Vol. 11.
- Neidert, N., Straehle, J., Erny, D., Sacalean, V., El Rahal, A., Steybe, D., Schmelzeisen, R., Vlachos, A., Reinacher, P. C., Coenen, V. A., Mizaikoff, B., Heiland, D. H., Prinz, M., Beck, J., and Schnell, O. (2022), “Stimulated Raman histology in the neurosurgical workflow of a major European neurosurgical center - part A”, *Neurosurg Rev*, Vol. 45, pp. 1731–1739.
- Nelson, E. D., and Fredman, M. L. (1970), “Hadamard Spectroscopy”, *Journal of the Optical Society of America*, Vol. 60 No. 12, pp. 1664-1669.

Oh, S., Lee, C.H, Yang, W., Li, A., Mukherjee, A., Basan, M., Ran, C., Yin, W., Tabin, C. J., Fu, D., Xie, X. S., and Kirschner, M. W. (2022), “Protein and lipid mass concentration measurement in tissues by stimulated Raman scattering microscopy”, *Proceedings of the National Academy of Sciences*, Vol. 119 No. 17.

Ohlsson, H., Yang, A., Dong, R., Sastry, S. (2012) “CPRL – An extension of compressive sensing to the phase retrieval problem”, *Advances in Neural Information Processing Systems*, Vol. 25.

Orringer, D. A., Pandian, B., Niknafs, Y. S., Hollon, T. C., Boyle, J., Lewis S., Garrard, M., Hervey-Jumper, S. L., Garton, H. J. L., Maher, C. O., Heth, J. A., Sagher, O., Wilkinson, D. A., Snuderl, M., Venneti, S., Ramkissoon, S. H., McFadden, K. A., Fisher-Hubbard, A., Lieberman, A. P., Johnson, T. D., Xie, X. S., Trautman, J. K., Freudiger, C. W., and Camelo-Piragua, S. (2017), “Rapid intraoperative histology of unprocessed surgical specimens via fibre-laser-based stimulated Raman scattering microscopy”, *Nat Biomed Eng*, Vol. 1.

Rudin, L. I., Osher, S., and Fatemi, E. (1992), “Nonlinear total variation based noise removal algorithms”, *Physica D: Nonlinear Phenomena*, Vol. 60 No. 1-4, pp. 259-268.

Saar, B., G., Freudiger, C. W., Reichman, J., Stanley, C. M., Holtom, G. R., and Xie, X. S. (2010), “Video-rate molecular imaging in vivo with stimulated Raman scattering”, *Science*, Vol. 330 No. 6009, pp. 1368-1370.

Sarri, B., Poizat, F., Heuke, S., Wojak, J., Franchi, F., Caillol, F., Giovannini, M., Rigneault, H. (2019), “Stimulated Raman histology: one to one comparison with standard hematoxylin and eosin staining”, *Biomedical Optics Express*, Vol. 10 No. 1, pp. 5378-5384.

Scotte, C., Sivankutty, S., Bartels, R. A., and Rigneault, H. (2020), “Line-scan compressive Raman imaging with spatiospectral encoding”, *Optics Letters*, Vol. 45 No. 19, pp. 5567-5570.

Seidel, M., Pressacco, F., Akcaalan, O., Binhammer, T., Darvill, J., Ekanayake, N., Frede, M., Grosse-Wortmann, U., Heber, M., Heyl, C. M., Kutnyakhov, D., Li, C., Mohr, C., Muller, J., Puncken, O., Redlin, H., Schirmel, N., Schulz, S., Swiderski, A., Tavakol, H., Tunnermann, H., Vidolim C., Wenthaus, L., Wind, N., Winkelmann, L., Manschwetus, B., and Hartl, I. (2022) “Ultrafast MHz-rate burst-mode pump-probe laser for the FLASH FEL facility based on nonlinear compression of ps-level pulses from an Yb-amplifier chain” *Laser & Photonics Reviews*, Vol. 16 No. 3.

Shewchuk, J. R. (1994) “An introduction of the conjugate gradient method without the pain” Manuscript. <https://www.cs.cmu.edu/quake-papers/painless-conjugate-gradient.pdf>

Shi, L., Wei, M., Miao, Y., Qian, N., Shi, L., Singer, R. A., Benninger, R. K. P., Min, W. (2022), “Highly-multiplexed volumetric mapping with Raman dye imaging and tissue clearing”, *Nature Biotechnology*, Vol. 40, pp. 364-373.

Sloane, N. J. A., and Harwit, M. (1976) “Masks for Hadamard transform optics, and weighing

designs”, *Applied Optics*, Vol. 15 No. 10, pp. 107-114.

Soldevila, F., Dong, J., Tajahuerce, E., Gigan, S., and de Aguiar, H. B. (2019), “Fast compressive Raman bio-imaging via matrix completion”, *Optica*, Vol. 6 No. 3, pp. 341-346.

Steepleton, J. (2019), “Constructions of Hadamard Matrices”, Chancellor’s Honors Program Projects, [https://trace.tennessee.edu/utk\\_cihanhonoproj/2266](https://trace.tennessee.edu/utk_cihanhonoproj/2266)

Thompson, J. V., Bixler, J. N., Hokr, B. H., Noojin, G. D., Scully, M. O., and Yakovlev, V. V. (2017), “Single-shot chemical detection and identification with compressed hyperspectral Raman imaging”, *Optics Letters*, Vol. 42, No. 11, pp. 2169-2172.

Tsaig, Y. and Donoho, D. L. (2006), “Extensions of compressed sensing”, *Signal Processing*, Vol. 86 No. 3, pp. 549-571.

Wei, L., and Min, W. (2018), “Electronic preresonance stimulated Raman scattering microscopy”, *J. Phys. Chem. Lett.*, Vol. 9 No. 15, pp. 4294-4301.

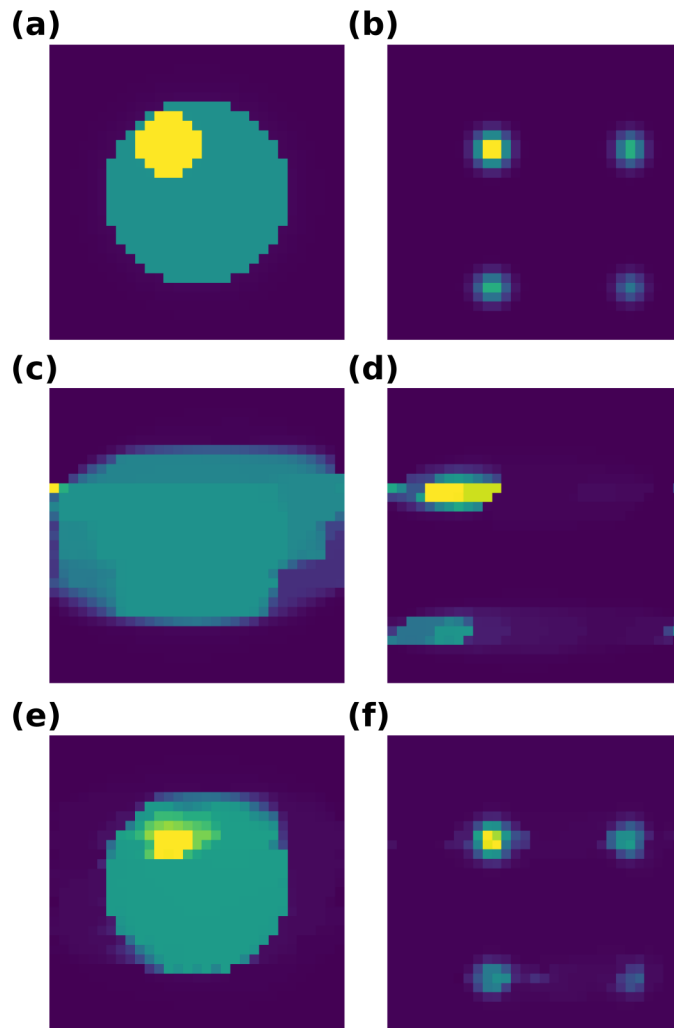
Yoon, T., Kim, C-S., Kim, K., and Choi, J-R. (2018), “Emerging applications of digital micromirror devices in biophotonic fields”, *Optics & Laser Technology*, Vol. 104, pp. 17-25.

Zhang, P., Wang, G., Zhang, X., and Li, Y-Q. (2020), “Single-acquisition 2-D multifocal Raman spectroscopy using compressive sensing”, *Anal. Chem.*, Vol. 92 No. 1, pp. 1326-1332.

Zhang, X., Dorlhiac, G., Landry, M. P., and Streets, A. (2022), “Phototoxic effects of nonlinear optical microscopy on cell cycle, oxidative states, and gene expression”, *Scientific Reports*, Vol. 12.

Zhao, Z., and Kobayashi, Y. (2016) “Ytterbium fiber-based, 270 fs, 100 W chirped pulse amplification laser system with 1 MHz repetition rate”, *Applied Physics Express*, Vol. 9 No. 1.

## 5.6 Supplementary Material



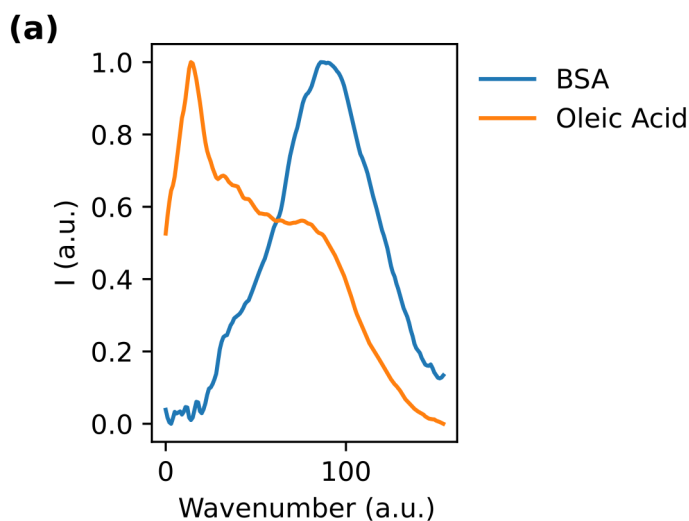
**Figure 5.6:** EM retrieval of a number of test images

(a-b) Two 31x31 pixel test images. The image in (b) is identical to the image used in figure 5.5a. (c-d) Retrieved images using the EM algorithm used for the bead reconstruction in figure 5.4 for the two images in (a) and (b) under 9% sampling. (e-f) Retrieved images using the total variation under quadratic constraints algorithm for the two test images under 9% sampling. The image in (e) is the same as the middle image in figure 5.5e.



**Figure 5.7:** Line-focus objective adapter

An adapter is used to attach a lens tube containing a cylindrical lens to the back of the objective. The lens tube has an adjustable length to allow for the cylindrical lens to correctly focus the beam to the back focal plane of the objective. With the current design, a separate adapter needs to be configured for each objective which may be used in the line-scan compressive modality. While objective lenses from the same manufacturer are typically designed with the same parfocal distance, the back focal distance of the lens system usually varies. This is particularly true for lenses of differing optical power. Additionally, it is important to note that the back focal plane typically lies somewhere within the objective housing. To maintain the highest possible beam quality, this needs to be carefully considered in conjunction with the size of the back aperture and any intermediate apertures that may exist within the housing. The cylindrical lens also needs to be carefully oriented to focus along the proper axis. With the current setup this is a time-intensive process of manually adjusting the lens orientation with respect to the threads of the lens tube and the thread adapters.



**Figure 5.8:** SRS spectra of bovine serum albumin and oleic acid

(a) Representative spectra of BSA and oleic acid in the C-H stretching window acquired on the SRS microscope using spectral focusing. The x-axis is left as wavenumbers as a precise calibration was not performed during collection of the spectra. The window roughly covers  $2830\text{-}3000\text{ cm}^{-1}$ .

### 5.6.1 Additional parts list

To modify the standard SRS microscope described in chapter 2 to perform multiplexed/compressed sensing measurements, a small number of additional components are required and are tabulated below. These include the DMD and its control board, some cylindrical lenses, and lens tubes and adapters for modifying the objective to produce a line-focus as shown in figure 5.7. The cylindrical lens can alternatively be placed elsewhere in the optical path which would require different optomechanics than those listed below. The grating(s) used for pulse stretching in the optical setup in figure 2.1 can be repurposed to the detection side to reproduce the imaging modes described in figure 5.1.

| <b>Description</b>   | <b>Quantity</b> | <b>Part Number/Supplier</b> |
|--|-----------------|-----------------------------|
| Near-infrared DMD  | 1               | DLPLCR65NEVM<br>Digi-Key    |
| Control board for DMD                                      | 1               | DLPLCRC410EVM<br>Digi-Key   |
| 1", 1" diameter, lens tube                                 | 1               | SM1L10<br>Thorlabs          |
| 0.31" adjustable, 1" diameter, lens tube                   | 1               | SM1V05<br>Thorlabs          |
| SM1 to RMS thread adapter                                  | 1               | SM1A3<br>Thorlabs           |
| Retaining ring for 1" diameter lens tube                   | 1               | SM1RR<br>Thorlabs           |
| Mounted, 1" diameter, 50 mm focal length cylindrical lens  | 1               | LJ1695RM-B<br>Thorlabs      |
| Mounted, 1" diameter, 200 mm focal length cylindrical lens | 1               | LJ1653RM-B<br>Thorlabs      |
| Scanning sample stage                                      | 1               | MLS203-1<br>Thorlabs        |

**Table 5.2:** Additional optics and optomechanics for DMD-based SRS microscopy

# Chapter 6

## Conclusion

Due to the chemically specific nature of the SRS spectrum, microscopy employing the SRS interaction can be used for label-free imaging. For imaging of high-concentration biomolecules, in aggregate, this has proven to be an excellent alternative to other microscopy techniques, such as those based upon stains or fluorescent dyes. In turn, this has led to its adoption for a number of specific purposes, such as label-free histology. Yet, despite the potential benefit of label-free imaging, SRS microscopy has not enjoyed the widespread adoption and application that fluorescence microscopy has in the life sciences. To some degree, the technology remains confined to the research of a number of specialists in the field. This can partly be explained by the technological difficulty of SRS imaging compared to older, more established, microscopy methods. However, this does not account for the lack of uptake in its entirety. A major challenge faced by SRS microscopy continues to be an at least perceived lack of specificity and imaging speed to tackle many classes of biomedical imaging problems. This dissertation has provided evidence that this need not be the case. Methods have been presented to extend both the range of applicability and the accessibility of the SRS microscopy technique in biological sciences.

We began with an overview of the construction of a "traditional" SRS microscope. On its own this provides a useful guide for any researchers wishing to tackle the fabrication of a similar system in their own laboratories. Observations made about the implementation help to define the limits within which the methodology operates - i.e., what can and cannot be expected to be achieved by employing it - and furthermore provide a reference for troubleshooting technical issues. Beyond this, a new measurement procedure was proposed for optical pulse characterization in nonlinear microscopy which can serve as a starting point for making more accurate, high-resolution, measurements.

We next considered the analysis of SRS imaging results. Certain best practices were reiterated, some of which, while once common knowledge, have been lost over years amidst the increasing technical complexity of making optical measurements. Easy-to-implement algorithms were presented for quantitative data extraction. Finally, an objective metric for the evaluation and comparison of images in general was introduced. This provides a mechanism to ascertain the quality of any given measurement, and inform decisions about the interpretation of a result.

After optimization of the SRS microscope and refinement of our analysis techniques we focused on the application of SRS microscopy to the imaging of a new class of metabolites, neurotransmitters. This extends the use of the technology to the study of small molecules in living systems, and, most importantly, in a specific manner. Microscopy-driven investigations of small



molecules are challenging due to their non-fluorescent nature, and how perturbative work-arounds to this can be. The capability to image this class of molecules using SRS microscopy, in a non-perturbative manner, opens the door to new fundamental investigations.

Finally, extensions to the standard SRS microscope are proposed. The study of dynamic processes in living systems is ultimately limited by the speed at which data can be acquired. The nature of the detection process in SRS microscopy makes it challenging to reach the fast acquisition speeds required to study some of the most interesting processes in biology, such as the rapid communication between neurons in the form of neurotransmission. Through careful consideration of the structure of the data ultimately acquired, SRS imaging can be sped up through new sampling methods, extending the possible applications of the technique in the life sciences to everything from cytometry to the observation of rapid signalling processes.

In conclusion, the results presented in this dissertation demonstrate new avenues for the application of SRS microscopy in biology. SRS microscopy can be used to image with both specificity and high-speed. This opens the door to new applications of the technology, including problems that may be intractable with other methods.

# Appendix A

## *srsmicro* API

The following provides a reference documentation for the software package *srsmicro* for controlling the SRS microscope. This documentation provides a reference at the time of dissemination publication. For the most up-to-date documentation, visit the associated github repository. This documentation was adapted from that generated with Doxygen 1.9.5.

### A.1 Namespace Documentation

### A.2 optosky Namespace Reference

#### Classes

- class CppSpec

### A.3 srsmicro Namespace Reference

#### Namespaces

- namespace `__main__`  
*The main GUI application entry point managing the top level GUI, controller, and data object instances and their Signal/Slot connections.*
- namespace `_version`
- namespace `control`
- namespace `gui`
- namespace `setup`
- namespace `utilities`

### A.4 srsmicro.\_\_main\_\_ Namespace Reference

The main GUI application entry point managing the top level GUI, controller, and data object instances and their Signal/Slot connections.

## Variables

- `app = QtWidgets.QApplication(sys.argv)`
- `mw = MainWindow()`
- `controller = MainController()`

### A.4.1 Detailed Description

The main GUI application entry point managing the top level GUI, controller, and data object instances and their Signal/Slot connections.

### A.4.2 Variable Documentation

#### A.4.2.1 `app`

```
srsmicro.__main__.app = QtWidgets.QApplication(sys.argv)
```

#### A.4.2.2 `controller`

```
srsmicro.__main__.controller = MainController()
```

#### A.4.2.3 `mw`

```
srsmicro.__main__.mw = MainWindow()
```

## A.5 `srsmicro._version` Namespace Reference

### Variables

- `string __version__ = '0.1'`

### A.5.1 Variable Documentation

#### A.5.1.1 `__version__`

```
string srsmicro._version.__version__ = '0.1' [private]
```

## A.6 srsmicro.control Namespace Reference

### Namespaces

- namespace devices
- namespace maincontroller
- namespace olympusappcontroller

*Definition of the OlympusAppController class which has hard-coded routines for managing the proprietary FV1200 Olympus software, without user input.*

- namespace statusreporter

*Definition of the StatusReporter class which is intended as a worker on a separate thread for reporting device status updates.*

## A.7 srsmicro.control.devices Namespace Reference

### Namespaces

- namespace delaystage

*Definition of the DelayStage class for interaction with the Newport FCL200 delay stage as well as custom exception classes PositionerError and CommandError for parsing the corresponding error codes.*

- namespace device

*Definition of base class for communicating with physical devices.*

- namespace insight

*Definition of the Insight class for interacting with the Insight DS+ femtosecond laser.*

- namespace samplestage

*Definition of the SampleStage class for interaction with the Prior Scientific sample XY stage.*

- namespace serialdevice

*Definition of base class for devices using serial communication.*

- namespace zurichdaq

*Definition of the ZurichDaq class for managing the dataAcquisitionModule for the ZI API asynchronously.*

- namespace zurichlockin

*Definition of the ZurichLockin class for interaction with the Zurich Instruments HF2LI lock-in amplifier as well as custom exception classes APIError and DeviceNotFoundError for zhinst API errors.*

## A.8 srsmicro.control.devices.delaystage Namespace Reference

Definition of the DelayStage class for interaction with the Newport FCL200 delay stage as well as custom exception classes PositionerError and CommandError for parsing the corresponding error codes.

## Classes

- class `CommandError`

*Exception class for command errors for the Newport FCL200 delay stage.*

- class `DelayStage`

*The `DelayStage` class for controlling the Newport FCL200 delay stage.*

- class `PositionerError`

*Exception class for positioner errors for the Newport FCL200 delay stage.*

### A.8.1 Detailed Description

Definition of the `DelayStage` class for interaction with the Newport FCL200 delay stage as well as custom exception classes `PositionerError` and `CommandError` for parsing the corresponding error codes.

## A.9 `srsmicro.control.devices.device` Namespace Reference

Definition of base class for communicating with physical devices.

### Classes

- class `Device`

*The base device class.*

### A.9.1 Detailed Description

Definition of base class for communicating with physical devices.

## A.10 `srsmicro.control.devices.insight` Namespace Reference

Definition of the `Insight` class for interacting with the Insight DS+ femtosecond laser.

### Classes

- class `Insight`

*The `Insight` class for controlling the SpectraPhysics Insight DS+ femtosecond laser/OPO.*

### A.10.1 Detailed Description

Definition of the `Insight` class for interacting with the Insight DS+ femtosecond laser.

## **A.11 srsmicro.control.devices.samplestage Namespace Reference**

Definition of the SampleStage class for interaction with the Prior Scientific sample XY stage.

### **Classes**

- class SampleStage

*Facilitates serial communication with Prior Scientific sample stage.*

### **A.11.1 Detailed Description**

Definition of the SampleStage class for interaction with the Prior Scientific sample XY stage.

## **A.12 srsmicro.control.devices.serialdevice Namespace Reference**

Definition of base class for devices using serial communication.

### **Classes**

- class SerialDevice

*The base class for serial devices.*

### **A.12.1 Detailed Description**

Definition of base class for devices using serial communication.

## **A.13 srsmicro.control.devices.zurichdaq Namespace Reference**

Definition of the ZurichDaq class for managing the dataAcquisitionModule for the ZI API asynchronously.

### **Classes**

- class ZurichDaq

*The class for managing the Lock-in amplifier's data acquisition module.*

### **A.13.1 Detailed Description**

Definition of the ZurichDaq class for managing the dataAcquisitionModule for the ZI API asynchronously.

## **A.14 srsmicro.control.devices.zurichlockin Namespace Reference**

Definition of the ZurichLockin class for interaction with the Zurich Instruments HF2LI lock-in amplifier as well as custom exception classes APIError and DeviceNotFoundError for zhinst API errors.

### **Classes**

- class APIError  
*Exception class for errors due to ZI API.*
- class DeviceNotFoundError  
*Exception class when unable to find Lockin devices connected.*
- class ZurichLockin  
*The base class for the ZI HF2LI lock-in amplifier.*

### **A.14.1 Detailed Description**

Definition of the ZurichLockin class for interaction with the Zurich Instruments HF2LI lock-in amplifier as well as custom exception classes APIError and DeviceNotFoundError for zhinst API errors.

## **A.15 srsmicro.control.maincontroller Namespace Reference**

### **Classes**

- class MainController  
*The MainController class for interfacing the GUI elements with device objects.*

## **A.16 srsmicro.control.olympusappcontroller Namespace Reference**

Definition of the OlympusAppController class which has hard-coded routines for managing the proprietary FV1200 Olympus software, without user input.

## Classes

- class OlympusAppController

### A.16.1 Detailed Description

Definition of the OlympusAppController class which has hard-coded routines for managing the proprietary FV1200 Olympus software, without user input.

## A.17 srsmicro.control.statusreporter Namespace Reference

Definition of the StatusReporter class which is intended as a worker on a separate thread for reporting device status updates.

## Classes

- class StatusReporter

*Worker class for device status querying.*

### A.17.1 Detailed Description

Definition of the StatusReporter class which is intended as a worker on a separate thread for reporting device status updates.

## A.18 srsmicro.gui Namespace Reference

## Namespaces

- namespace basicpanel

*Definition of the BasicPanel class for providing the parsing functions to populate device specific GUI panels.*

- namespace delaystagepanel

*Definition of the DelayStagePanel class for providing the necessary GUI elements for controlling the Newport FCL200 delay stage's parameters.*

- namespace femtoramanpanel

*Definition of the FemtoRamanPanel class for providing the necessary GUI elements for controlling a fsSRS microscopy experiment.*

- namespace insightpanel

*Definition of the InsightPanel class for providing the necessary GUI elements for controlling the Insight DS+ parameters and device state.*

- namespace mainwindow

*Definition of the MainWindow class for organization and arrangement of all GUI elements.*



- namespace zipanel

*Definition of the ZiPanel class for providing the necessary GUI elements for interacting with the Zurich Instruments HF2LI lock-in amplifier.*

## **A.19 srsmicro.gui.basicpanel Namespace Reference**

Definition of the BasicPanel class for providing the parsing functions to populate device specific GUI panels.

### **Classes**

- class BasicPanel

*The BasicPanel serves as a template class for device specific GUI elements.*

### **A.19.1 Detailed Description**

Definition of the BasicPanel class for providing the parsing functions to populate device specific GUI panels.

## **A.20 srsmicro.gui.delaystagepanel Namespace Reference**

Definition of the DelayStagePanel class for providing the necessary GUI elements for controlling the Newport FCL200 delay stage's parameters.

### **Classes**

- class DelayStagePanel

*The DelayStagePanel provides the GUI elements for interacting with the Newport FCL200 optical delay stage.*

### **A.20.1 Detailed Description**

Definition of the DelayStagePanel class for providing the necessary GUI elements for controlling the Newport FCL200 delay stage's parameters.

## **A.21 srsmicro.gui.femtoramanpanel Namespace Reference**

Definition of the FemtoRamanPanel class for providing the necessary GUI elements for controlling a fsSRS microscopy experiment.

## Classes

- class FemtoRamanPanel

*The FemtoRamanPanel provides the GUI elements needed for performing femtosecond SRS microscopy measurements and displaying the results.*

### A.21.1 Detailed Description

Definition of the FemtoRamanPanel class for providing the necessary GUI elements for controlling a fsSRS microscopy experiment.

## A.22 srsmicro.gui.insightpanel Namespace Reference

Definition of the InsightPanel class for providing the necessary GUI elements for controlling the Insight DS+ parameters and device state.

## Classes

- class InsightPanel

*The InsightPanel provides the GUI elements for interacting with the Insight DS+.*

### A.22.1 Detailed Description

Definition of the InsightPanel class for providing the necessary GUI elements for controlling the Insight DS+ parameters and device state.

## A.23 srsmicro.gui.mainwindow Namespace Reference

Definition of the MainWindow class for organization and arrangement of all GUI elements.

## Classes

- class MainWindow

*The main SRS microscope window populated with the necessary elements for various experiment types and loading of data/projects.*

### A.23.1 Detailed Description

Definition of the MainWindow class for organization and arrangement of all GUI elements. Serves as the contact point for communication with the controller/devices.

## A.24 srsmicro.gui.zipanel Namespace Reference

Definition of the ZiPanel class for providing the necessary GUI elements for interacting with the Zurich Instruments HF2LI lock-in amplifier.

### Classes

- class ZiPanel

### A.24.1 Detailed Description

Definition of the ZiPanel class for providing the necessary GUI elements for interacting with the Zurich Instruments HF2LI lock-in amplifier.

## A.25 srsmicro.setup Namespace Reference

### Variables

- name
- version
- packages
- entry\_points

### A.25.1 Variable Documentation

#### A.25.1.1 entry\_points

`srsmicro.setup.entry_points`

#### A.25.1.2 name

`srsmicro.setup.name`

#### A.25.1.3 packages

`srsmicro.setup.packages`

#### A.25.1.4 version

`srsmicro.setup.version`

## A.26 srsmicro.utilities Namespace Reference

### Namespaces

- namespace conversions
- namespace experimentresult

*Definition of ExperimentResult class which manages all data generated, useful metadata such as experimental parameters, and any log files generated.*

## A.27 srsmicro.utilities.conversions Namespace Reference

### Functions

- def calc\_omega (w\_p, w\_s=1040)  
*Calculate the wavenumber given a pump and Stokes wavelength in nm.*
- def logstotext (h5, output)  
*Produces a readable text file of recorded experimental logs.*
- def load\_zi\_yaml (self, path)  
*Load in the parameter hierarchy from the included configuration file.*

### A.27.1 Detailed Description

@brief Useful functions for unit and type conversions, e.g. OPO wavelength to SRS wavenumber, switching between file formats, and loading configuration files.

### A.27.2 Function Documentation

#### A.27.2.1 calc\_omega()

```
def srsmicro.utilities.conversions.calc_omega (
    w_p,
    w_s = 1040 )
```

Calculate the wavenumber given a pump and Stokes wavelength in nm.

Parameters

|       |   |
|-------|---|
| $w_p$ | (float) Pump wavelength in nm.                    |
| $w_s$ | (float) Stokes wavelength in nm. Default is 1040. |

### A.27.2.2 `load_zi_yaml()`

```
def srsmicro.utilities.conversions.load_zi_yaml (
    self,
    path )
```

Load in the parameter hierarchy from the included configuration file.

One copy is loaded into a list-of-lists format which can be passed directly to ZI API objects for setting parameters. A separate copy is converted to a dictionary which matches the format used by other devices for accessing parameters. The config file is used by the actual device control object as well as GUI elements.

Returns

`cond_vars_list` (list[list]) All possible ZI parameters in list of list format. Each list is of the form [str, int/float/long] where the first entry is a parameter path, and the second its setting.

`cond_vars` (dict[str]) The above list converted to a dictionary for easier parameter accession and to match the format used by other devices.

### A.27.2.3 `logstotext()`

```
def srsmicro.utilities.conversions.logstotext (
    h5,
    output )
```

Produces a readable text file of recorded experimental logs.

Parameters

|               |  |
|---------------|--|
| <i>h5</i>     | (str) Path to the experimental hdf5 file.        |
| <i>output</i> | Desired path for the output plain text log file. |

## A.28 `srsmicro.utilities.experimentresult` Namespace Reference

Definition of ExperimentResult class which manages all data generated, useful metadata such as experimental parameters, and any log files generated.

### Classes

- class ExperimentResult

#### A.28.1 Detailed Description

Definition of ExperimentResult class which manages all data generated, useful metadata such as experimental parameters, and any log files generated.

## A.29 thorlabs Namespace Reference

### Classes

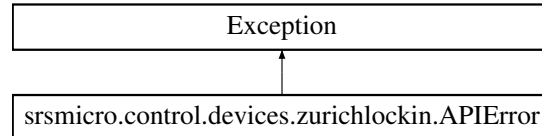
- class CppKcube

## A.30 Class Documentation

### A.31 srsmicro.control.devices.zurichlockin.APIError Class Reference

Exception class for errors due to ZI API.

Inheritance diagram for srsmicro.control.devices.zurichlockin.APIError:



### Public Member Functions

- def `__init__` (self, msg)  
*APIError class initializer.*
- def `__str__` (self)  
*String representation of the PositionError on print.*

### Public Attributes

- msg

#### A.31.1 Detailed Description

Exception class for errors due to ZI API.

#### A.31.2 Constructor & Destructor Documentation

### A.31.2.1 `__init__()`

```
def srsmicro.control.devices.zurichlockin.APIError.__init__(
    self,
    msg )
```

APIError class initializer.

Parameters

|            |                                      |
|------------|--------------------------------------|
| <i>msg</i> | (str) Additional message to include. |
|------------|--------------------------------------|

## A.31.3 Member Function Documentation

### A.31.3.1 `__str__()`

```
def srsmicro.control.devices.zurichlockin.APIError.__str__(
    self )
```

String representation of the PositionError on print.

Returns

(str) self.msg

## A.31.4 Member Data Documentation

### A.31.4.1 `msg`

srsmicro.control.devices.zurichlockin.APIError.msg

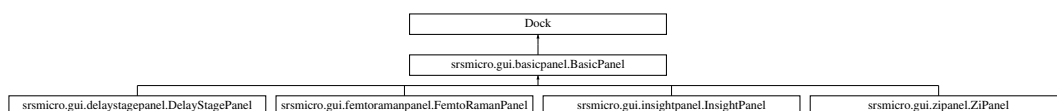
The documentation for this class was generated from the following file:

- control/devices/zurichlockin.py

## A.32 srsmicro.gui.basicpanel.BasicPanel Class Reference

The BasicPanel serves as a template class for device specific GUI elements.

Inheritance diagram for srsmicro.gui.basicpanel.BasicPanel:



## Public Member Functions

- `def __init__ (self, *args, **kwargs)`  
*The BasicPanel constructor.*
- `def update_state (self, params)`  
*Slot to update status widgets.*
- `def update_data (self, data)`  
*Slot to update display upon receipt of new data.*
- `def update_log (self, msg)`  
*Slot to update the device log.*

## Public Attributes

- `h1font`  
*(QtGui.QFont) The font used for section headers on the panel.*
- `h2font`  
*(QtGui.QFont) The font used for subsection headers on the panel.*
- `h1_patn`  
*(str) Regex pattern for matching header 1 type text (?<^# ) Looks behind for the line starting with '#' pattern (.*
- `h2_patn`  
*(str) Regex pattern for matching header 1 type text (?<^## ) Looks behind for the line starting with '##' pattern (.*
- `status_vars`  
*(dict) Contains GUI elements that report on device parameters.*
- `control_vars`  
*(dict) Contains GUI elements that modify device parameters.*
- `cmd_patn`  
*(str) Regex pattern for matching a command.*
  
- `param_patn`  
*) which matches a terminating '}' character*

## Static Public Attributes

- `expmt_msg = Signal(str)`  
*(Signal) Emit a log message/statusbar update.*
- `cmd = Signal(object, object, object)`



## Private Member Functions

- `def _status_widgets (self, path)`  
*Parses a provided text file (.fmt extension) to populate the status GUI elements.*
- `def _control_widgets (self, path, funcs)`  
*Parses a provided text file (.ctrl extension) to populate the control GUI elements.*
- `def _update_controls (self, param, val)`  
*Private, device specific, function called by the update\_state slot to perform updating of control GUI elements.*
- `def _log_widget (self)`  
*A widget to maintain device specific logging information.*

### A.32.1 Detailed Description

The BasicPanel serves as a template class for device specific GUI elements. Basic organizational units are included, as are global parameters like font sizes/choices.

### A.32.2 Constructor & Destructor Documentation

#### A.32.2.1 `__init__()`

```
def srsmicro.gui.basicpanel.BasicPanel.__init__ (
    self,
    * args,
    ** kwargs )
```

The BasicPanel constructor.

Parameters

|                 |   |
|-----------------|---|
| <i>statpath</i> | (str) The path to the text file containing the GUI's status elements. These elements only display information.                      |
| <i>ctrlpath</i> | (str) The path to the text file containing the GUI's control elements. These elements allow interaction with the associated device. |

Reimplemented in `srsmicro.gui.delaystagepanel.DelayStagePanel`, `srsmicro.gui.femtoramanpanel.FemtoRamanPanel`, `srsmicro.gui.insightpanel.InsightPanel`, and `srsmicro.gui.zipanel.ZiPanel`.

### A.32.3 Member Function Documentation

### A.32.3.1 `_control_widgets()`

```
def srsmicro.gui.basicpanel.BasicPanel._control_widgets (
    self,
    path,
    funcs ) [private]
```

Parses a provided text file (.ctrl extension) to populate the control GUI elements. Saves the elements in the object's `control_vars` variable allowing them to be updated automatically based upon the associated device's status. Associates functions to the GUI elements.

Parameters

|              |   |
|--------------|---|
| <i>path</i>  | (str) Path to the format file.  |
| <i>funcs</i> | (dict) Dictionary of str:method pairs associating function names written in the text file to the actual python methods. |

### A.32.3.2 `_log_widget()`

```
def srsmicro.gui.basicpanel.BasicPanel._log_widget (
    self ) [private]
```

A widget to maintain device specific logging information. This can include more detailed information (such as specific hardware failures) than the more general experiment logs.

### A.32.3.3 `_status_widgets()`

```
def srsmicro.gui.basicpanel.BasicPanel._status_widgets (
    self,
    path ) [private]
```

Parses a provided text file (.fmt extension) to populate the status GUI elements. Saves the elements in the object's `status_vars` variable allowing them to be updated automatically based upon the associated device's status.

Parameters

|             |                                |
|-------------|--------------------------------|
| <i>path</i> | (str) Path to the format file. |
|-------------|--------------------------------|

### A.32.3.4 `_update_controls()`

```
def srsmicro.gui.basicpanel.BasicPanel._update_controls (
    self,
    param,
    val ) [private]
```

Private, device specific, function called by the `update_state` slot to perform updating of control GUI elements.

This allows changing the display of those elements to provide additional information to that provided by the status elements. E.g button colors or toggling for indicating activation.

Parameters

|              |   |
|--------------|---|
| <i>param</i> | The parameter from the associated device. @val The value of that parameter. |
|--------------|---|

Reimplemented in `srsmicro.gui.insightpanel.InsightPanel`.

### A.32.3.5 `update_data()`

```
def srsmicro.gui.basicpanel.BasicPanel.update_data (
    self,
    data )
```

Slot to update display upon receipt of new data.

Parameters

|             |                               |
|-------------|-------------------------------|
| <i>data</i> | The new data to be displayed. |
|-------------|-------------------------------|

### A.32.3.6 `update_log()`

```
def srsmicro.gui.basicpanel.BasicPanel.update_log (
    self,
    msg )
```

Slot to update the device log.

Parameters

|            |                              |
|------------|------------------------------|
| <i>msg</i> | Text to be added to the log. |
|------------|------------------------------|

### A.32.3.7 `update_state()`

```
def srsmicro.gui.basicpanel.BasicPanel.update_state (
    self,
    params )
```

Slot to update status widgets.

Parameters

|               |   |
|---------------|---|
| <i>params</i> | Device parameters and the values to be displayed. |
|---------------|---|

## A.32.4 Member Data Documentation

### A.32.4.1 cmd

`srsmicro.gui.basicpanel.BasicPanel.cmd = Signal(object, object, object) [static]`

### A.32.4.2 cmd\_patn

`srsmicro.gui.basicpanel.BasicPanel.cmd_patn`

(str) Regex pattern for matching a command.

(?<=) Looks behind for a '\$' character (.\* ) Then matches any number of characters before (?=) which matches a terminating '\$' character

### A.32.4.3 control\_vars

`srsmicro.gui.basicpanel.BasicPanel.control_vars`

(dict) Contains GUI elements that modify device parameters.

These elements, like buttons, take user input.

### A.32.4.4 expmt\_msg

`srsmicro.gui.basicpanel.BasicPanel.expmt_msg = Signal(str) [static]`

(Signal) Emit a log message/statusbar update.

### A.32.4.5 h1\_patn

`srsmicro.gui.basicpanel.BasicPanel.h1_patn`

(str) Regex pattern for matching header 1 type text (?<^# ) Looks behind for the line starting with '#' pattern (.

\*) Matches any type and any number of characters following it

### A.32.4.6 h1font

`srsmicro.gui.basicpanel.BasicPanel.h1font`

(QtGui.QFont) The font used for section headers on the panel.

### A.32.4.7 h2\_patn

`srsmicro.gui.basicpanel.BasicPanel.h2_patn`

(str) Regex pattern for matching header 1 type text (?<^## ) Looks behind for the line starting with '##' pattern (.

\*) Matches any type and any number of characters following it

#### A.32.4.8 h2font

`srsmicro.gui.basicpanel.BasicPanel.h2font`  
(QtGui.QFont) The font used for subsection headers on the panel.

#### A.32.4.9 param\_patn

`srsmicro.gui.basicpanel.BasicPanel.param_patn`  
) which matches a terminating '}' character  
(str) Regex pattern for matching command parameters.  
Must be used with `re.findall` (not `search`) to capture all parameters. (?<=) Looks behind for a '{'  
character (.\*) Then matches any number of characters before (?=

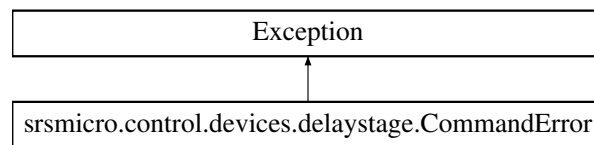
#### A.32.4.10 status\_vars

`srsmicro.gui.basicpanel.BasicPanel.status_vars`  
(dict) Contains GUI elements that report on device parameters.  
These elements do not take user input.  
The documentation for this class was generated from the following file:

- `gui/basicpanel.py`

### A.33 srsmicro.control.devices.delaystage.CommandError Class Reference

Exception class for command errors for the Newport FCL200 delay stage.  
Inheritance diagram for `srsmicro.control.devices.delaystage.CommandError`:



#### Public Member Functions

- `def __init__(self, error_code)`  
*CommandError class initializer.*
- `def __str__(self)`  
*String representation of the CommandError on print.*

#### Public Attributes

- `msg`

## Static Private Attributes

- dictionary `_cmd_errors`

*(dict[str]:str) String definitions for given command error codes.*

### A.33.1 Detailed Description

Exception class for command errors for the Newport FCL200 delay stage.

### A.33.2 Constructor & Destructor Documentation

#### A.33.2.1 `__init__()`

```
def srsmicro.control.devices.delaystage.CommandError.__init__(  
    self,  
    error_code )
```

CommandError class initializer.

Parameters

|                         |  |
|-------------------------|--|
| <code>error_code</code> | (str) Character for the received command error. Definitions are accessed through class dict <code>_cmd_errors</code> . |
|-------------------------|--|

### A.33.3 Member Function Documentation

#### A.33.3.1 `__str__()`

```
def srsmicro.control.devices.delaystage.CommandError.__str__(  
    self )
```

String representation of the CommandError on print.

Returns

(str) `self.msg`

### A.33.4 Member Data Documentation

#### A.33.4.1 `_cmd_errors`

`srsmicro.control.devices.delaystage.CommandError._cmd_errors` [static], [private]

### Initial value:

```
= {
    '@' : 'No error',
    'A' : 'Unknown message code or floating point controller address.',
    'B' : 'Controller address not correct.',
    'C' : 'Parameter missing or out of range.',
    'D' : 'Command not allowed.',
    'E' : 'Home sequence already started.',
    'G' : 'Displacement out of limits.',
    'H' : 'Command not allowed in NOT REFERENCED state.',
    'I' : 'Command not allowed in CONFIGURATION state.',
    'J' : 'Command not allowed in DISABLE state.',
    'K' : 'Command not allowed in READY state.',
    'L' : 'Command not allowed in HOMING state.',
    'M' : 'Command not allowed in MOVING state.',
    'N' : 'Current position out of software limit.',
    'S' : 'Communication Time Out.',
    'U' : 'Error during EEPROM access.',
    'V' : 'Error during command execution.'
}
```

(dict[str]:str) String definitions for given command error codes.

Command errors are given as characters which are the keys of the dictionary.

#### A.33.4.2 msg

srsmicro.control.devices.delaystage.CommandError.msg

The documentation for this class was generated from the following file:

- control/devices/delaystage.py

## A.34 thorlabs::CppKcube Class Reference

```
#include <cppkcube.hpp>
```

### Public Member Functions

- CppKcube (int poll\_time)
- void cpp\_close ()
- int cpp\_move (int newpos)
- int cpp\_home ()
- int cpp\_set\_vel (int newvel)
- int cpp\_set\_accel (int newaccel)
- int cpp\_pos ()
- int cpp\_vel ()
- int cpp\_accel ()

### Private Attributes

- char \_cpp\_serialNo [9]
- char \_cpp\_desc [65]

- int `_cpp_pos`
- int `_cpp_vel`
- int `_cpp_accel`

## A.34.1 Constructor & Destructor Documentation

### A.34.1.1 CppKcube()

```
thorlabs::CppKcube::CppKcube (
    int poll_time = 100 )
```

Class using the thorlabs API for control of a KCube motor. This class is reimplemented and compiled in Cython for direct import into the python-based control software.

Parameters

|                  |   |
|------------------|---|
| <i>poll_time</i> | an interger argument for serial communication polling time. |
|------------------|---|

## A.34.2 Member Function Documentation

### A.34.2.1 `cpp_accel()`

```
int thorlabs::CppKcube::cpp_accel ( )
```

Member function for returning the current motor acceleration.

Returns

`_cpp_accel` an integer representing the motor acceleration.

### A.34.2.2 `cpp_close()`

```
void thorlabs::CppKcube::cpp_close ( )
```

Member function for closing communication with the KCubee.

Returns

void.



### A.34.2.3 `cpp_home()`

```
int thorlabs::CppKcube::cpp_home ( )
```

Member function for homing the motor.

Returns

`_cpp_pos` an integer representing the motor position.

### A.34.2.4 `cpp_move()`

```
int thorlabs::CppKcube::cpp_move (
    int newpos )
```

Member function for moving the motor to a new position.

Parameters

|                            |  |
|----------------------------|--|
| <code><i>newpos</i></code> | an integer for what the new velocity should be set to. |
|----------------------------|--|

Returns

`_cpp_pos` an integer representing the motor position.

### A.34.2.5 `cpp_pos()`

```
int thorlabs::CppKcube::cpp_pos ( )
```

Member function for returning the current motor position.

Returns

`_cpp_pos` an integer representing the motor position.

### A.34.2.6 `cpp_set_accel()`

```
int thorlabs::CppKcube::cpp_set_accel (
    int newaccel )
```

Member function for setting the motor acceleration.

Parameters

|                              |  |
|------------------------------|--|
| <code><i>newaccel</i></code> | an integer for what the new acceleration should be set to. |
|------------------------------|--|

Returns

`_cpp_accel` an integer representing the current acceleration.

#### A.34.2.7 `cpp_set_vel()`

```
int thorlabs::CppKcube::cpp_set_vel (
    int newvel )
```

Member function for setting the motor velocity.

Parameters

|               |  |
|---------------|--|
| <i>newvel</i> | an integer for what the new velocity should be set to. |
|---------------|--|

Returns

`_cpp_vel` an integer representing the current velocity.

#### A.34.2.8 `cpp_vel()`

```
int thorlabs::CppKcube::cpp_vel ( )
```

Member function for returning the current motor velocity.

Returns

`_cpp_vel` an integer representing the motor velocity.

### A.34.3 Member Data Documentation

#### A.34.3.1 `_cpp_accel`

```
int thorlabs::CppKcube::_cpp_accel [private]
```

#### A.34.3.2 `_cpp_desc`

```
char thorlabs::CppKcube::_cpp_desc[65] [private]
```

#### A.34.3.3 `_cpp_pos`

```
int thorlabs::CppKcube::_cpp_pos [private]
```

#### A.34.3.4 `_cpp_serialNo`

```
char thorlabs::CppKcube::_cpp_serialNo[9] [private]
```

#### A.34.3.5 `_cpp_vel`

```
int thorlabs::CppKcube::_cpp_vel [private]
```

The documentation for this class was generated from the following files:

- control/devices/cppkcube.hpp
- control/devices/cppkcube.cpp

### A.35 `optosky::CppSpec` Class Reference

```
#include <cppspec.hpp>
```

#### Public Member Functions

- `CppSpec ()`
- `int cpp_acquire ()`
- `int cpp_close ()`
- `int cpp_set_exposeTime ()`
- `int cpp_set_averages ()`

#### Private Attributes

- `HP2000_wrapper::HP2000Wrapper m_spec`
- `HP2000_wrapper::Spectrum _data`
- `bool _is_open`
- `double _wavelength [2048]`
- `double _spectrum [2048]`
- `int _exposureTime`
- `int _avgs`

#### A.35.1 Constructor & Destructor Documentation

##### A.35.1.1 `CppSpec()`

```
optosky::CppSpec::CppSpec ( )
```

## A.35.2 Member Function Documentation

### A.35.2.1 `cpp_acquire()`

```
int optosky::CppSpec::cpp_acquire ( )
```

### A.35.2.2 `cpp_close()`

```
int optosky::CppSpec::cpp_close ( )
```

### A.35.2.3 `cpp_set_averages()`

```
int optosky::CppSpec::cpp_set_averages ( )
```

### A.35.2.4 `cpp_set_exposeTime()`

```
int optosky::CppSpec::cpp_set_exposeTime ( )
```

## A.35.3 Member Data Documentation

### A.35.3.1 `_avgs`

```
int optosky::CppSpec::_avgs [private]
```

### A.35.3.2 `_data`

```
HP2000_wrapper::Spectrum optosky::CppSpec::_data [private]
```

### A.35.3.3 `_exposureTime`

```
int optosky::CppSpec::_exposureTime [private]
```

### A.35.3.4 `_is_open`

```
bool optosky::CppSpec::_is_open [private]
```

### A.35.3.5 `_spectrum`

```
double optosky::CppSpec::_spectrum[2048] [private]
```

### A.35.3.6 `_wavelength`

```
double optosky::CppSpec::_wavelength[2048] [private]
```

### A.35.3.7 `m_spec`

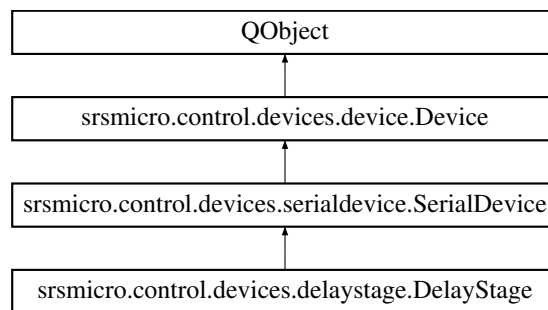
```
HP2000_wrapper::HP2000Wrapper optosky::CppSpec::m_spec [private]
```

The documentation for this class was generated from the following files:

- `control/devices/cppspec.hpp`
- `control/devices/cppspec.cpp`

## A.36 `srsmicro.control.devices.delaystage.DelayStage` Class Reference

The `DelayStage` class for controlling the Newport FCL200 delay stage.  
Inheritance diagram for `srsmicro.control.devices.delaystage.DelayStage`:



### Public Member Functions

- `def __init__ (self, name='Delay Stage')`

*The `DelayStage` class initializer.*

- `def check_errors (self)`

*Check for command errors and also query state.*

- `def parse_cmd (self, param, val)`

*This function is overwritten by subclasses to perform the actions appropriately upon parsing messages from the GUI.*

## Public Attributes

- `comport`

## Static Public Attributes

- `dictionary cmds`

## Private Member Functions

- `def _query_state (self)`  
*Query device state and parse any positioner errors.*
- `def _read_current_conditions (self)`
- `def _move_relative (self, val)`  
*Move the stage to a new absolute position.*
- `def _move_absolute (self, val)`  
*Move the stage to a new absolute position.*

## Private Attributes

- `_cond_vars`  
*(dict[str]:str) Holds current delay stage conditions.*

## Static Private Attributes

- `dictionary _controller_state`  
*(dict[str]:str) Dictionary reference for definitions of two hexadecimal delay stage state responses.*

### A.36.1 Detailed Description

The DelayStage class for controlling the Newport FCL200 delay stage. Extends the SerialDevice class with device specific methods and variables.

### A.36.2 Constructor & Destructor Documentation

#### A.36.2.1 `__init__()`

```
def srsmicro.control.devices.delaystage.DelayStage.__init__ (
    self,
    name = 'Delay Stage' )
```

The DelayStage class initializer.

Initiliazes `comport` (inherited from `SerialDevice`) to `COM7`

Reimplemented from `srsmicro.control.devices.serialdevice.SerialDevice`.

## A.36.3 Member Function Documentation

### A.36.3.1 `_move_absolute()`

```
def srsmicro.control.devices.delaystage.DelayStage._move_absolute (
    self,
    val ) [private]
```

Move the stage to a new absolute position.

Makes use of the `_move_relative` method because obtaining an estimate of the time required to make the movement requires calculating the relative position difference.

Parameters

|            |   |
|------------|---|
| <i>val</i> | (float) The absolute position to move the delay stage to. |
|------------|---|

### A.36.3.2 `_move_relative()`

```
def srsmicro.control.devices.delaystage.DelayStage._move_relative (
    self,
    val ) [private]
```

Move the stage to a new absolute position.

First calculates the time required to move to the new position. This time is used to block communication until the move is completed.

Parameters

|            |                                      |
|------------|--------------------------------------|
| <i>val</i> | (float) The relative motion to make. |
|------------|--------------------------------------|

### A.36.3.3 `_query_state()`

```
def srsmicro.control.devices.delaystage.DelayStage._query_state (
    self ) [private]
```

Query device state and parse any positioner errors.

Errors are accessed through raising a `PositionerError` exception. The class definition for the exception contains the error code definitions.

Reimplemented from `srsmicro.control.devices.device.Device`.

### A.36.3.4 `_read_current_conditions()`

```
def srsmicro.control.devices.delaystage.DelayStage._read_current_conditions (
```

```
self ) [private]
```

### A.36.3.5 check\_errors()

```
def srsmicro.control.devices.delaystage.DelayStage.check_errors (
    self )
```

Check for command errors and also query state.

Errors are accessed through raising a `CommandError` exception. The class definition for the exception contains the error code definitions.

### A.36.3.6 parse\_cmd()

```
def srsmicro.control.devices.delaystage.DelayStage.parse_cmd (
    self,
    param,
    val )
```

This function is overwritten by subclasses to perform the actions appropriately upon parsing messages from the GUI.

Reimplemented from `srsmicro.control.devices.device.Device`.

## A.36.4 Member Data Documentation

### A.36.4.1 \_cond\_vars

```
srsmicro.control.devices.delaystage.DelayStage._cond_vars [private]
```

(dict[str]:str) Holds current delay stage conditions.

Keys: pos, vel, accel, cmd\_err, pos\_err

### A.36.4.2 \_controller\_state

```
srsmicro.control.devices.delaystage.DelayStage._controller_state [static], [private]
```

**Initial value:**

```
= {
    '0A': 'NOT REFERENCED from RESET.',
    '0B': 'NOT REFERENCED from HOMING.',
    '0C': 'NOT REFERENCED from CONFIGURATION.',
    '0D': 'NOT REFERENCED from DISABLE.',
    '0E': 'NOT REFERENCED from READY.',
    '0F': 'NOT REFERENCED from MOVING.',
    '10': 'NOT REFERENCED - NO PARAMETERS IN MEMORY.',
    '14': 'CONFIGURATION',
    '1E': 'HOMING.',
    '28': 'MOVING.',
    '32': 'READY from HOMING.',
    '33': 'READY from MOVING.',
    '34': 'READY from DISABLE.',
    '3C': 'DISABLE from READY.',
    '3D': 'DISABLE from MOVING.' }
```

(dict[str]:str) Dictionary reference for definitions of two hexadecimal delay stage state responses.



### A.36.4.3 cmds

dictionary srsmicro.control.devices.delaystage.DelayStage.cmds [static]

#### Initial value:

```
= { 'pos' : '1TP',  
    'vel' : '1VA',  
    'accel' : '1AC' }
```

### A.36.4.4 comport

srsmicro.control.devices.delaystage.DelayStage.comport

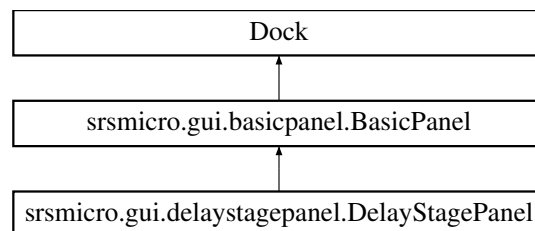
The documentation for this class was generated from the following file:

- control/devices/delaystage.py

## A.37 srsmicro.gui.delaystagepanel.DelayStagePanel Class Reference

The DelayStagePanel provides the GUI elements for interacting with the Newport FCL200 optical delay stage.

Inheritance diagram for srsmicro.gui.delaystagepanel.DelayStagePanel:



### Public Member Functions

- def \_\_init\_\_ (self, \*args, \*\*kwargs)

*The DelayStagePanel constructor.*

### Public Attributes

- funcs

## Private Member Functions

- def `_turnon` (self)
- def `_home` (self)
- def `_abs_mv` (self)

*Function associated to the button and form to move to an absolute Delay Stage position.*

- def `_rel_neg` (self)

*Function associated to the button and form to move the Delay Stage relative to its current position, in the negative direction.*

- def `_rel_pos` (self)

*Function associated to the button and form to move the Delay Stage relative to its current position in the positive direction.*

- def `_set_vel` (self)

*Function associated to the button and form to set the Delay Stage velocity.*

- def `_set_accel` (self)

*Function associated to the button and form to set the Delay Stage acceleration.*

## Additional Inherited Members

### A.37.1 Detailed Description

The DelayStagePanel provides the GUI elements for interacting with the Newport FCL200 optical delay stage.

### A.37.2 Constructor & Destructor Documentation

#### A.37.2.1 `__init__()`

```
def srsmicro.gui.delaystagepanel.DelayStagePanel.__init__(  
    self,  
    * args,  
    ** kwargs )
```

The DelayStagePanel constructor.

Parameters

|                 |   |
|-----------------|---|
| <i>statpath</i> | (str) The path to the text file containing the GUI's status elements. These elements only display information.                      |
| <i>ctrlpath</i> | (str) The path to the text file containing the GUI's control elements. These elements allow interaction with the associated device. |

Reimplemented from `srsmicro.gui.basicpanel.BasicPanel`.

## A.37.3 Member Function Documentation

### A.37.3.1 `_abs_mv()`

```
def srsmicro.gui.delaystagepanel.DelayStagePanel._abs_mv (
    self ) [private]
```

Function associated to the button and form to move to an absolute Delay Stage position.  
Emits the appropriate signal.

### A.37.3.2 `_home()`

```
def srsmicro.gui.delaystagepanel.DelayStagePanel._home (
    self ) [private]
```

### A.37.3.3 `_rel_neg()`

```
def srsmicro.gui.delaystagepanel.DelayStagePanel._rel_neg (
    self ) [private]
```

Function associated to the button and form to move the Delay Stage relative to its current position, in the negative direction.  
Emits the appropriate signal.

### A.37.3.4 `_rel_pos()`

```
def srsmicro.gui.delaystagepanel.DelayStagePanel._rel_pos (
    self ) [private]
```

Function associated to the button and form to move the Delay Stage relative to its current position in the positive direction.  
Emits the appropriate signal.

### A.37.3.5 `_set_accel()`

```
def srsmicro.gui.delaystagepanel.DelayStagePanel._set_accel (
    self ) [private]
```

Function associated to the button and form to set the Delay Stage acceleration.  
Emits the appropriate signal.

### A.37.3.6 `_set_vel()`

```
def srsmicro.gui.delaystagepanel.DelayStagePanel._set_vel (
    self ) [private]
```

Function associated to the button and form to set the Delay Stage velocity.  
Emits the appropriate signal.

### A.37.3.7 `_turnon()`

```
def srsmicro.gui.delaystagepanel.DelayStagePanel._turnon (
    self ) [private]
```

## A.37.4 Member Data Documentation

### A.37.4.1 funcs

`srsmicro.gui.delaystagepanel.DelayStagePanel.funcs`

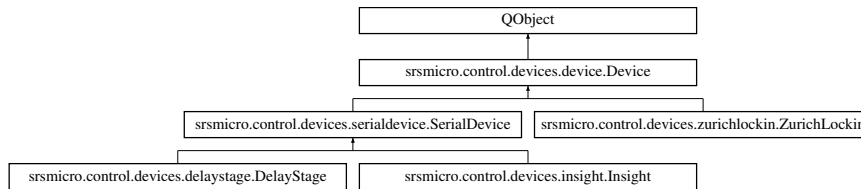
The documentation for this class was generated from the following file:

- `gui/delaystagepanel.py`

## A.38 `srsmicro.control.devices.device.Device` Class Reference

The base device class.

Inheritance diagram for `srsmicro.control.devices.device.Device`:



## Public Member Functions

- `def __init__ (self, name='Device')`  
*The SerialDevice base class initializer.*
- `def open (self)`  
*Open communication with the physical device.*
- `def close (self)`  
*Close communication for the physical device if needed.*
- `def comtime (self)`  
*Property for the default time to wait after writing a command.*
- `def comtime (self, val)`  
*Property setter for changing the default time to wait after writing a command.*
- `def parse_cmd (self, param, val)`  
*This function is overwritten by subclasses to perform the actions appropriately upon parsing messages from the GUI.*

- `def query_state (self)`  
*Performs device specific status queries.*
- `def logs (self)`  
*Property to return device specific log information, such as errors and communication updates over the serial port.*
- `def current_time (self)`  
*Property to return current time.*
- `def exit (self)`  
*Device shutdown method.*

## Public Attributes

- `name`  
*Device identifier for simplifying signal and slot operation.*

## Static Public Attributes

- `state = Signal(object, object)`
- `cmd_result = Signal(str)`  
*(Signal) For relaying device parameters and values.*

## Private Member Functions

- `def _open (self)`  
*Private function overwritten by subclasses for the actual device specific opening communication.*
- `def _close (self)`  
*Private function overwritten by subclasses for the actual device specific closing communication.*
- `def _query_state (self)`

## Private Attributes

- `_isconnected`  
*(Boolean) Is the serial port open or closed.*
- `_comtime`  
*Time (in seconds) to wait before reading back the result of a command.*

### A.38.1 Detailed Description

The base device class.

Provides basic interface for interacting with device parameters, logging, and using Qt Signals and Slots.

## A.38.2 Constructor & Destructor Documentation

### A.38.2.1 `__init__()`

```
def srsmicro.control.devices.device.Device.__init__ (
    self,
    name = 'Device' )
```

The SerialDevice base class initializer.

Reimplemented in `srsmicro.control.devices.delaystage.DelayStage`, `srsmicro.control.devices.insight.Insight`, `srsmicro.control.devices.zurichlockin.ZurichLockin`, and `srsmicro.control.devices.serialdevice.SerialDevice`.

## A.38.3 Member Function Documentation

### A.38.3.1 `_close()`

```
def srsmicro.control.devices.device.Device._close (
    self ) [private]
```

Private function overwritten by subclasses for the actual device specific closing communication.

Reimplemented in `srsmicro.control.devices.serialdevice.SerialDevice`.

### A.38.3.2 `_open()`

```
def srsmicro.control.devices.device.Device._open (
    self ) [private]
```

Private function overwritten by subclasses for the actual device specific opening communication.

Reimplemented in `srsmicro.control.devices.serialdevice.SerialDevice`, and `srsmicro.control.devices.zurichlockin.ZurichLockin`.

### A.38.3.3 `_query_state()`

```
def srsmicro.control.devices.device.Device._query_state (
    self ) [private]
```

Reimplemented in `srsmicro.control.devices.delaystage.DelayStage`, and `srsmicro.control.devices.insight.Insight`.

#### A.38.3.4 close()

```
def srsmicro.control.devices.device.Device.close (
    self )
```

Close communication for the physical device if needed.

#### A.38.3.5 comtime() [1/2]

```
def srsmicro.control.devices.device.Device.comtime (
    self )
```

Property for the default time to wait after writing a command.

Returns

`_comtime` (float) Wait time in seconds.

#### A.38.3.6 comtime() [2/2]

```
def srsmicro.control.devices.device.Device.comtime (
    self,
    val )
```

Property setter for changing the default time to wait after writing a command.

Parameters

|                  |                               |
|------------------|-------------------------------|
| <code>val</code> | (float) Wait time in seconds. |
|------------------|-------------------------------|

#### A.38.3.7 current\_time()

```
def srsmicro.control.devices.device.Device.current_time (
    self )
```

Property to return current time.

Useful for logging purposes.

#### A.38.3.8 exit()

```
def srsmicro.control.devices.device.Device.exit (
    self )
```

Device shutdown method.

Overwritten by subclasses.

Reimplemented in `srsmicro.control.devices.insight.Insight`,

`srsmicro.control.devices.serialdevice.SerialDevice`, and

`srsmicro.control.devices.zurichlockin.ZurichLockin`.

### **A.38.3.9 logs()**

```
def srsmicro.control.devices.device.Device.logs (  
    self )
```

Property to return device specific log information, such as errors and communication updates over the serial port.

There is no setter for this property as the device manages its logs internally.

### **A.38.3.10 open()**

```
def srsmicro.control.devices.device.Device.open (  
    self )
```

Open communication with the physical device.

### **A.38.3.11 parse\_cmd()**

```
def srsmicro.control.devices.device.Device.parse_cmd (  
    self,  
    param,  
    val )
```

This function is overwritten by subclasses to perform the actions appropriately upon parsing messages from the GUI.

Reimplemented in `srsmicro.control.devices.delaystage.DelayStage`, and `srsmicro.control.devices.insight.Insight`.

### **A.38.3.12 query\_state()**

```
def srsmicro.control.devices.device.Device.query_state (  
    self )
```

Performs device specific status queries.

Intended for execution on a separate thread, so it can be run infinitely at given intervals. The associated private method should be overloaded on a per device basis to account for differences in communication syntax. This method has no return value. Instead the associated private method updates the internally managed state and then a signal is emitted with new parameter value pairs.

## **A.38.4 Member Data Documentation**



#### A.38.4.1 `_comtime`

`srsmicro.control.devices.device.Device._comtime` [private]  
Time (in seconds) to wait before reading back the result of a command.

#### A.38.4.2 `_isconnected`

`srsmicro.control.devices.device.Device._isconnected` [private]  
(Boolean) Is the serial port open or closed.

#### A.38.4.3 `cmd_result`

`srsmicro.control.devices.device.Device.cmd_result` = `Signal(str)` [static]  
(Signal) For relaying device parameters and values.  
(Signal) Last message read back from the device.  
Used to report results.

#### A.38.4.4 `name`

`srsmicro.control.devices.device.Device.name`  
Device identifier for simplifying signal and slot operation.

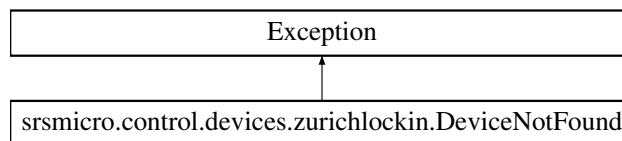
#### A.38.4.5 `state`

`srsmicro.control.devices.device.Device.state` = `Signal(object, object)` [static]  
The documentation for this class was generated from the following file:

- `control/devices/device.py`

## A.39 `srsmicro.control.devices.zurichlockin.DeviceNotFound` Class Reference

Exception class when unable to find Lockin devices connected.  
Inheritance diagram for `srsmicro.control.devices.zurichlockin.DeviceNotFound`:



## Public Member Functions

- `def __init__(self, msg)`  
*DeviceNotFound class initializer.*
- `def __str__(self)`  
*String representation of the PositionError on print.*

## Public Attributes

- `msg`

### A.39.1 Detailed Description

Exception class when unable to find Lockin devices connected.

### A.39.2 Constructor & Destructor Documentation

#### A.39.2.1 `__init__()`

```
def srsmicro.control.devices.zurichlockin.DeviceNotFound.__init__(  
    self,  
    msg )
```

DeviceNotFound class initializer.

Parameters

|            |                                      |
|------------|--------------------------------------|
| <i>msg</i> | (str) Additional message to include. |
|------------|--------------------------------------|

### A.39.3 Member Function Documentation

#### A.39.3.1 `__str__()`

```
def srsmicro.control.devices.zurichlockin.DeviceNotFound.__str__(  
    self )
```

String representation of the PositionError on print.

Returns

(str) `self.msg`

## A.39.4 Member Data Documentation

### A.39.4.1 msg

`srsmicro.control.devices.zurichlockin.DeviceNotFound.msg`

The documentation for this class was generated from the following file:

- `control/devices/zurichlockin.py`

## A.40 `srsmicro.utilities.experimentresult.ExperimentResult` Class Reference

### Public Member Functions

- `def __init__(self, path=None, name='experimentresults')`  
*ExperimentResult class initializer.*
- `def write_logs(self, logs)`  
*Write logs to the appropriate dataset in the file.*
- `def exit(self)`  
*Shutdown procedure.*

### Public Attributes

- `file`  
*An hdf5 File instance for storing logs and experiment results.*

### Private Attributes

- `_path`

### A.40.1 Detailed Description

The class definition of the `ExperimentResult` object which handles I/O of all data, metadata, and experiment logs.

### A.40.2 Constructor & Destructor Documentation

### A.40.2.1 `__init__()`

```
def srsmicro.utilities.experimentresult.ExperimentResult.__init__ (
    self,
    path = None,
    name = 'experimentresults' )
```

ExperimentResult class initializer.

Parameters

|             |   |
|-------------|---|
| <i>path</i> | (str) Destination path for output file. If none is provided (default) the current working directory will be used. |
| <i>name</i> | (str) Output file name. Default is experimentresults.   |

## A.40.3 Member Function Documentation

### A.40.3.1 `exit()`

```
def srsmicro.utilities.experimentresult.ExperimentResult.exit (
    self )
```

Shutdown procedure.

Properly close the working file.

### A.40.3.2 `write_logs()`

```
def srsmicro.utilities.experimentresult.ExperimentResult.write_logs (
    self,
    logs )
```

Write logs to the appropriate dataset in the file.

## A.40.4 Member Data Documentation

### A.40.4.1 `_path`

`srsmicro.utilities.experimentresult.ExperimentResult._path` [private]

### A.40.4.2 `file`

`srsmicro.utilities.experimentresult.ExperimentResult.file`

An hdf5 File instance for storing logs and experiment results.

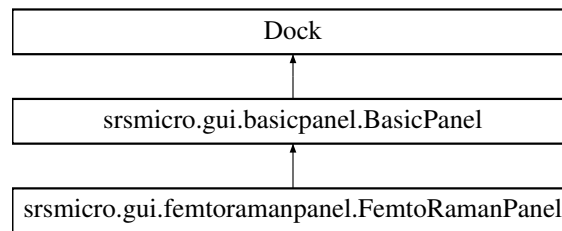
The documentation for this class was generated from the following file:

- utilities/experimentresult.py

## A.41 srmicro.gui.femtoramanpanel.FemtoRamanPanel Class Reference

The FemtoRamanPanel provides the GUI elements needed for performing femtosecond SRS microscopy measurements and displaying the results.

Inheritance diagram for srmicro.gui.femtoramanpanel.FemtoRamanPanel:



### Public Member Functions

- def `__init__` (self, \*args, \*\*kwargs)  
*The FemtoRamanPanel constructor.*

### Public Attributes

- funcs

### Private Member Functions

- def `_set_dwell` (self)  
*Function associated to the button and form to update the pixel dwell time during image raster scanning.*

### Additional Inherited Members

#### A.41.1 Detailed Description

The FemtoRamanPanel provides the GUI elements needed for performing femtosecond SRS microscopy measurements and displaying the results.

#### A.41.2 Constructor & Destructor Documentation

#### A.41.2.1 `__init__()`

```
def srsmicro.gui.femtoramanpanel.FemtoRamanPanel.__init__(
    self,
    * args,
    ** kwargs )
```

The FemtoRamanPanel constructor.

Parameters

|                 |   |
|-----------------|---|
| <i>statpath</i> | (str) The path to the text file containing the GUI's status elements. These elements only display information.                      |
| <i>ctrlpath</i> | (str) The path to the text file containing the GUI's control elements. These elements allow interaction with the associated device. |

Reimplemented from `srsmicro.gui.basicpanel.BasicPanel`.

### A.41.3 Member Function Documentation

#### A.41.3.1 `_set_dwell()`

```
def srsmicro.gui.femtoramanpanel.FemtoRamanPanel._set_dwell (
    self ) [private]
```

Function associated to the button and form to update the pixel dwell time during image raster scanning.

Emits the appropriate signal.

### A.41.4 Member Data Documentation

#### A.41.4.1 `funcs`

```
srsmicro.gui.femtoramanpanel.FemtoRamanPanel.funcs
```

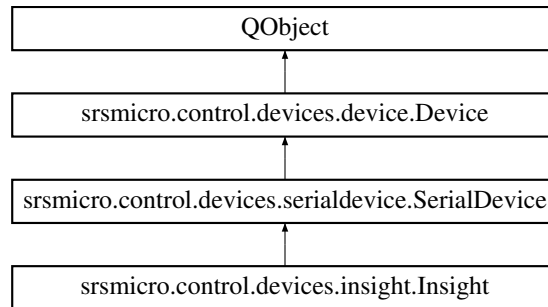
The documentation for this class was generated from the following file:

- `gui/femtoramanpanel.py`

## A.42 `srsmicro.control.devices.insight.Insight` Class Reference

The Insight class for controlling the SpectraPhysics Insight DS+ femtosecond laser/OPO.

Inheritance diagram for `srsmicro.control.devices.insight.Insight`:



## Public Member Functions

- `def __init__ (self, name='Insight')`  
*The Insight class initializer.*
- `def parse_cmd (self, param, val)`  
*This function is overwritten by subclasses to perform the actions appropriately upon parsing messages from the GUI.*
- `def exit (self)`  
*Device shutdown method.*

## Public Attributes

- baudrate
- comport

## Static Public Attributes

- dictionary `cmds`  
*(dict[str]:str) Dictionary reference for the write commands to retrieve or set various laser variables.*
- dictionary `fault_codes`  
*(dict[str]:str) Dictionary reference for numerical fault code meanings when read using the read history command.*
- dictionary `op_errors`  
*(dict[hex]:str) Dictionary reference for operational errors read using the generic status command.*

## Private Member Functions

- `def _query_state (self)`  
*Read the full status including operational state, history of error codes, and any current error codes.*

- def `_check_errors` (self, str state)
- def `_read_history` (self)
  - Reads error code history from the insight startup buffer.*
- str `_parse_op_state` (self, int resp)
  - Return current state in text form.*
- def `_read_current_conditions` (self)

## Private Attributes

- `_cond_vars`

### A.42.1 Detailed Description

The Insight class for controlling the SpectraPhysics Insight DS+ femtosecond laser/OPO. Extends the SerialDevice class with device specific methods and variables.

### A.42.2 Constructor & Destructor Documentation

#### A.42.2.1 `__init__()`

```
def srsmicro.control.devices.insight.Insight.__init__ (
    self,
    name = 'Insight' )
```

The Insight class initializer.

Initiliazes comport (inherited from SerialDevice) to COM6

Reimplemented from srsmicro.control.devices.serialdevice.SerialDevice.

### A.42.3 Member Function Documentation

#### A.42.3.1 `_check_errors()`

```
def srsmicro.control.devices.insight.Insight._check_errors (
    self,
    str state ) [private]
```

#### A.42.3.2 `_parse_op_state()`

```
str srsmicro.control.devices.insight.Insight._parse_op_state (
    self,
    int resp ) [private]
```

Return current state in text form.



## Parameters

|             |   |
|-------------|---|
| <i>resp</i> | (int) Numeric code corresponding to operational state of laser. |
|-------------|---|

## Returns

state (str) Text description of laser state.

### A.42.3.3 `_query_state()`

```
def srsmicro.control.devices.insight.Insight._query_state (
    self ) [private]
```

Read the full status including operational state, history of error codes, and any current error codes. Also reads values such as humidity, current, and diode temperature. Reimplemented from `srsmicro.control.devices.device.Device`.

### A.42.3.4 `_read_current_conditions()`

```
def srsmicro.control.devices.insight.Insight._read_current_conditions (
    self ) [private]
```

### A.42.3.5 `_read_history()`

```
def srsmicro.control.devices.insight.Insight._read_history (
    self ) [private]
```

Reads error code history from the insight startup buffer.

### A.42.3.6 `exit()`

```
def srsmicro.control.devices.insight.Insight.exit (
    self )
```

Device shutdown method.  
Overwritten by subclasses.

Reimplemented from `srsmicro.control.devices.serialdevice.SerialDevice`.

### A.42.3.7 `parse_cmd()`

```
def srsmicro.control.devices.insight.Insight.parse_cmd (
    self,
    param,
    val )
```

This function is overwritten by subclasses to perform the actions appropriately upon parsing messages from the GUI.

Reimplemented from `srsmicro.control.devices.device.Device`.

## A.42.4 Member Data Documentation

### A.42.4.1 `_cond_vars`

`srsmicro.control.devices.insight.Insight._cond_vars` [private]

### A.42.4.2 `baudrate`

`srsmicro.control.devices.insight.Insight.baudrate`

### A.42.4.3 `cmds`

`srsmicro.control.devices.insight.Insight.cmds` [static]

#### Initial value:

```
= {
    'd1_curr': 'READ:PLASer:DIODE1:CURRent' , # Operating current for diode 1
    'd1_hrs': ' READ:PLASer:DIODE1:HOURS', # Operating hours for diode 1
    'd1_temp': 'READ:PLASer:DIODE1:TEMPerature', # Diode 1 Temperature
    'd2_curr': 'READ:PLASer:DIODE2:CURRent', # Operating current for diode 2
    'd2_hrs': 'READ:PLASer:DIODE2:HOURS', # Operating hours for diode 2
    'd2_temp': 'READ:PLASer:DIODE2:TEMPerature', # Diode 2 temperature
    'humidity': 'READ:HUMidity',
    'dsm_max': 'CONT:SLMAX', # Maximum GVD compensation position for insight, given current wavelength
    'dsm_min': 'CONT:SLMIN', # Minimum GVD compensation position for insight, given current wavelength
    'dsm_pos': 'CONT:DSMPOS', # The current GVD compensation position for insight
    'fixed_shutter': 'IRSHUTter', # Open/close 1040 nm shutter
    'main_shutter': 'SHUTter', # Open/close main (OPO) shutter
    'opo_wl': 'WAVelength', # OPO wavelength
    'align': 'MODE'
}
```

(dict[str]:str) Dictionary reference for the write commands to retrieve or set various laser variables.

Keys are the shorthand used in code.

### A.42.4.4 `comport`

`srsmicro.control.devices.insight.Insight.comport`

### A.42.4.5 `fault_codes`

`srsmicro.control.devices.insight.Insight.fault_codes` [static]

#### Initial value:

```
= {
```

```

'000': 'Normal operation.',
'056': 'Fault: Hardware timeout. Notify SpectraPhysics if it continues.',
'066': 'Fault: Software timeout. Speak with system operator.',
'088': 'Fault: Diode thermistor short. Contact SpectraPhysics.',
'089': 'Fault: Diode thermistor open. Contact SpectraPhysics.',
'090': 'Fault: Diodes too hot (T>30). Check cooling system.',
'091': 'Fault: Diodes warm (T>27). Check cooling system.',
'092': 'Fault: Diodes cold (T<17). Check cooling system.',
'117': 'Fault: Internal interlock opened. Contact SpectraPhysics.',
'118': 'Fault: CDRH interlock open.',
'119': 'Fault: Power supply interlock. Check cable.',
'120': 'Fault: Key switch interlock. Turn key.',
'129': 'Fault: Very high humidity. Change purge cartridge.',
'130': 'Warning: High humidity. Change purge cartridge soon.',
'481': 'Fault: Slow diode ramp. Contact SpectraPhysics.',
'482': 'Fault: Low fs oscillator power. Contact SpectraPhysics.',
'483': 'Fault: low FTO power. Try different wavelengths. Contact SpectraPhysics.'
}

```

(dict[str]:str) Dictionary reference for numerical fault code meanings when read using the read history command.

#### A.42.4.6 op\_errors

srsmicro.control.devices.insight.Insight.op\_errors [static]

##### Initial value:

```

= { 0x00000200 : 'User interlock open. Laser forced off.',
    0x00000400 : 'Safety keyswitch interlock open. Laser forced off.',
    0x00000800 : 'Power supply interlock open. Laser forced off.',
    0x00001000 : 'Internal interlock open. Laser forced off.',
    0x00004000 : 'Detecting a warning. Check history for cause.',
    0x00008000 : 'Faul detected. Laser diodes turned off. Check history.'
}

```

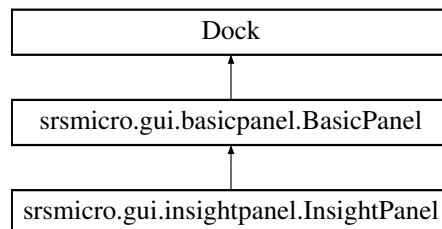
(dict[hex]:str) Dictionary reference for operational errors read using the generic status command. The documentation for this class was generated from the following file:

- control/devices/insight.py

## A.43 srsmicro.gui.insightpanel.InsightPanel Class Reference

The InsightPanel provides the GUI elements for interacting with the Insight DS+.

Inheritance diagram for srsmicro.gui.insightpanel.InsightPanel:



## Public Member Functions

- `def __init__(self, *args, **kwargs)`

*The InsightPanel constructor.*

## Public Attributes

- `funcs`

## Private Member Functions

- `def _turnon(self, *args)`

*Function associated to the button to turn the Insight DS+ on.*

- `def _alignmode(self)`

*Function associated to the button to put the Insight DS+ into alignment mode.*

- `def _tunewl(self)`

*Function associated to the button and form to change the wavelength the Insight DS+ OPO is tuned to.*

- `def _main_shutter(self)`

*Function associated to the button to open or close the main OPO shutter of the Insight DS+.*

- `def _fixed_shutter(self)`

*Function associated to the button to open or close the fixed 1040 nm shutter of the Insight DS+.*

- `def _update_controls(self, param, val)`

*Function to update the control GUI element appearance depending on the Insight DS+ status.*

## Additional Inherited Members

### A.43.1 Detailed Description

The InsightPanel provides the GUI elements for interacting with the Insight DS+.

### A.43.2 Constructor & Destructor Documentation

#### A.43.2.1 `__init__()`

```
def srsmicro.gui.insightpanel.InsightPanel.__init__(
    self,
    * args,
    ** kwargs )
```

The InsightPanel constructor.

## Parameters

|                 |   |
|-----------------|---|
| <i>statpath</i> | (str) The path to the text file containing the GUI's status elements. These elements only display information.                      |
| <i>ctrlpath</i> | (str) The path to the text file containing the GUI's control elements. These elements allow interaction with the associated device. |

Reimplemented from `srsmicro.gui.basicpanel.BasicPanel`.

### A.43.3 Member Function Documentation

#### A.43.3.1 `_alignmode()`

```
def srsmicro.gui.insightpanel.InsightPanel._alignmode (
    self ) [private]
```

Function associated to the button to put the Insight DS+ into alignment mode.  
Emits the appropriate signal.

#### A.43.3.2 `_fixed_shutter()`

```
def srsmicro.gui.insightpanel.InsightPanel._fixed_shutter (
    self ) [private]
```

Function associated to the button to open or close the fixed 1040 nm shutter of the Insight DS+.  
Emits the appropriate signal.

#### A.43.3.3 `_main_shutter()`

```
def srsmicro.gui.insightpanel.InsightPanel._main_shutter (
    self ) [private]
```

Function associated to the button to open or close the main OPO shutter of the Insight DS+.  
Emits the appropriate signal.

#### A.43.3.4 `_tunewl()`

```
def srsmicro.gui.insightpanel.InsightPanel._tunewl (
    self ) [private]
```

Function associated to the button and form to change the wavelength the Insight DS+ OPO is tuned to.  
Emits the appropriate signal.

#### A.43.3.5 `_turnon()`

```
def srsmicro.gui.insightpanel.InsightPanel._turnon (
    self,
    * args ) [private]
```

Function associated to the button to turn the Insight DS+ on.  
Emits the appropriate signal.

#### A.43.3.6 `_update_controls()`

```
def srsmicro.gui.insightpanel.InsightPanel._update_controls (
    self,
    param,
    val ) [private]
```

Function to update the control GUI element appearance depending on the Insight DS+ status.  
The function updates the display color of buttons to indicate laser emission/open shutters.  
Reimplemented from `srsmicro.gui.basicpanel.BasicPanel`.

### A.43.4 Member Data Documentation

#### A.43.4.1 `funcs`

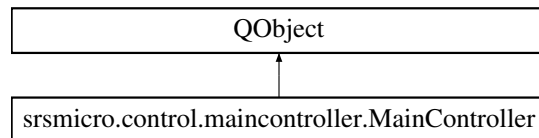
`srsmicro.gui.insightpanel.InsightPanel.funcs`

The documentation for this class was generated from the following file:

- `gui/insightpanel.py`

## A.44 `srsmicro.control.maincontroller.MainController` Class Reference

The `MainController` class for interfacing the GUI elements with device objects.  
Inheritance diagram for `srsmicro.control.maincontroller.MainController`:



### Public Member Functions

- `def __init__ (self)`  
*The MainController class constructor.*
- `def init_devices (self)`  
*Run the startup procedure for the various devices and handle any messages/errors.*
- `def parse_data (self, type, data)`

*Appropriately package data for display as the multiple types of output (images, plots, etc) may be expected.*

- `def distribute_cmd (self, device, param, val)`

*Parse messages from the GUI and pass them to the appropriate devices.*

- `str global_control (self, str parameter, str val)`

*Manage device responses to what are currently called "global" parameters, including COM ports etc.*

- `def return_device_conditions (self)`

*Method will be deprecated.*

- `def exit (self)`

*Run the shutdown procedures for each device in turn.*

## **Public Attributes**

- `cmd_result`

*Holds the result of the last attempted action on any device.*

- `running`

## **Static Public Attributes**

- `device_state = Signal(object, object)`

*(Signal) Emitted with the name of the device, and its settings as the two parameters.*

- `data = Signal(object)`

*(Signal) Emitted with new data for plotting/display.*

- `log = Signal(str)`

*(Signal) Emitted with a message to be recorded on the GUIs log.*

- `insight_cmd = Signal(object, object)`

- `delaystage_cmd = Signal(object, object)`

## **Private Member Functions**

- `def _read_result (self, msg)`

*Receive the result of the last action taken.*

## **Private Attributes**

- `_insight`

*Instance of the Insight device for controlling the laser.*

- `_delaystage`

*Instance of the DelayStage device for controlling the delay stage.*

- `_zi`  
*Instance of the ZurichLockin device for controlling the lock-in amplifier.*
- `_status_thread`  
*A separate thread for running the StatusReporter worker.*
- `_reporter`  
*Instance of a StatusReporter which takes a list of devices and queries their state on a loop.*

## A.44.1 Detailed Description

The MainController class for interfacing the GUI elements with device objects. The MainController holds instances of all the devices and through appropriate Signals and Slots passes information to and from the GUI.

## A.44.2 Constructor & Destructor Documentation

### A.44.2.1 `__init__()`

```
def srsmicro.control.maincontroller.MainController.__init__ (
    self )
```

The MainController class constructor.

## A.44.3 Member Function Documentation

### A.44.3.1 `_read_result()`

```
def srsmicro.control.maincontroller.MainController._read_result (
    self,
    msg ) [private]
```

Receive the result of the last action taken.

### A.44.3.2 `distribute_cmd()`

```
def srsmicro.control.maincontroller.MainController.distribute_cmd (
    self,
    device,
    param,
    val )
```

Parse messages from the GUI and pass them to the appropriate devices.



#### **A.44.3.3 exit()**

```
def srsmicro.control.maincontroller.MainController.exit (
    self )
```

Run the shutdown procedures for each device in turn.  
Handle any errors as needed.

#### **A.44.3.4 global\_control()**

```
str srsmicro.control.maincontroller.MainController.global_control (
    self,
    str parameter,
    str val )
```

Manage device responses to what are currently called "global" parameters, including COM ports etc.  
This method will be deprecated.

#### **A.44.3.5 init\_devices()**

```
def srsmicro.control.maincontroller.MainController.init_devices (
    self )
```

Run the startup procedure for the various devices and handle any messages/errors.

#### **A.44.3.6 parse\_data()**

```
def srsmicro.control.maincontroller.MainController.parse_data (
    self,
    type,
    data )
```

Appropriately package data for display as the multiple types of output (images, plots, etc) may be expected.

#### **A.44.3.7 return\_device\_conditions()**

```
def srsmicro.control.maincontroller.MainController.return_device_conditions (
    self )
```

Method will be deprecated.

### **A.44.4 Member Data Documentation**

#### **A.44.4.1 `_delaystage`**

`srsmicro.control.maincontroller.MainController._delaystage` [private]  
Instance of the DelayStage device for controlling the delay stage.

#### **A.44.4.2 `_insight`**

`srsmicro.control.maincontroller.MainController._insight` [private]  
Instance of the Insight device for controlling the laser.

#### **A.44.4.3 `_reporter`**

`srsmicro.control.maincontroller.MainController._reporter` [private]  
Instance of a StatusReporter which takes a list of devices and queries their state on a loop.

#### **A.44.4.4 `_status_thread`**

`srsmicro.control.maincontroller.MainController._status_thread` [private]  
A separate thread for running the StatusReporter worker.

#### **A.44.4.5 `_zi`**

`srsmicro.control.maincontroller.MainController._zi` [private]  
Instance of the ZurichLockin device for controlling the lock-in amplifier.

#### **A.44.4.6 `cmd_result`**

`srsmicro.control.maincontroller.MainController.cmd_result`  
Holds the result of the last attempted action on any device.

#### **A.44.4.7 `data`**

`srsmicro.control.maincontroller.MainController.data = Signal(object)` [static]  
(Signal) Emitted with new data for plotting/display.

#### **A.44.4.8 `delaystage_cmd`**

`srsmicro.control.maincontroller.MainController.delaystage_cmd = Signal(object, object)`  
[static]

#### A.44.4.9 device\_state

```
srsmicro.control.maincontroller.MainController.device_state = Signal(object, object)
[static]
```

(Signal) Emitted with the name of the device, and its settings as the two parameters.

#### A.44.4.10 insight\_cmd

```
srsmicro.control.maincontroller.MainController.insight_cmd = Signal(object, object)
[static]
```

#### A.44.4.11 log

```
srsmicro.control.maincontroller.MainController.log = Signal(str) [static]
```

(Signal) Emitted with a message to be recorded on the GUIs log.

#### A.44.4.12 running

```
srsmicro.control.maincontroller.MainController.running
```

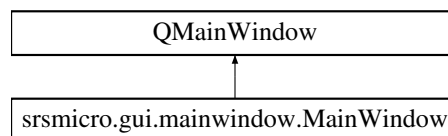
The documentation for this class was generated from the following file:

- control/maincontroller.py

## A.45 srsmicro.gui.mainwindow.MainWindow Class Reference

The main SRS microscope window populated with the necessary elements for various experiment types and loading of data/projects.

Inheritance diagram for srsmicro.gui.mainwindow.MainWindow:



### Public Member Functions

- def \_\_init\_\_(self)  
*The MainWindow constructor.*
- def parse\_state(self, device, params)  
*Slot for updating device specific GUI elements, e.g.*
- def global\_changed(self, obj, param, val)

*Emit a signal to be parsed by the controller.*

- def expmt\_changed (self, device, param, val)
- def statusbar (self)
- def statusbar (self, msg)

*Setter for the statusbar.*

- def update\_log (self, msg)

*Add text to the log widget.*

- def closeEvent (self, event)

*Shutdown routine.*

## Public Attributes

- statusbar

## Private Member Functions

- def \_create\_explorer\_dock (self)

*Populate Dock GUI element containing general experiment parameters and logs.*

- def \_create\_parameters\_widget (self)

*Create a ParameterTree for experiment parameters and add it to the \_explorer\_dock Dock widget.*

- def \_create\_expmt\_tab (self)

*Create the docks to display all widgets and features specific to the selected experiment.*

- def \_update\_expmt (self, obj, param, val)

*Slot for updating the experiment type parameter.*

- def \_create\_console\_dock (self)

*Create the dock to display a console to interact with the program.*

- def \_toggleconsole (self)

*Toggle visibility of the console.*

- def \_create\_menubar (self)

*Create the menubar and populate it.*

- def \_create\_filemenu (self)

*Create the 'File' menu to be added to the menubar.*

- def \_openfile (self)

*Open a file.*

- def \_saveim (self)

*Save an image.*

- def \_savezistate (self)

*Save a configuration/state.*

## Private Attributes

- `_explorer_dock`  
*Left aligned dock for global parameters and logs.*
- `_parameters_widget`  
*ParameterTree widget for global parameters such as experiment type and serial communication ports.*
- `_expmt_type`  
*ListParameter for selecting which experiment dock is displayed.*
- `_insight_com`  
*ListParameter for selecting the physical COM port of the Insight.*
- `_delaystage_com`  
*ListParameter for selecting the physical COM port of the delay stage.*
- `_current_panel`  
*The current panel being displayed.*
- `_insight_panel`  
*Contains Insight specific GUI elements.*
- `_delaystage_panel`  
*Contains delay stage specific GUI elements.*
- `_zi_panel`  
*Contains lock-in amplifier specific GUI elements.*
- `_fssrs_panel`  
*Contains GUI elements for running a simple SRS experiment and displaying the images/results.*
- `_console_dock`  
*Dock GUI element containing the Console Widget.*
- `_console`  
*The actual console widget.*
- `_showconsole`

### A.45.1 Detailed Description

The main SRS microscope window populated with the necessary elements for various experiment types and loading of data/projects.

Contains a DockArea, to allow rearrangement of the various GUI modules.

**Todo** Currently there is a bug requiring that titles of dock elements be hidden, which gets rid of the ability to move docks around and float them.

### A.45.2 Constructor & Destructor Documentation

#### A.45.2.1 `__init__()`

```
def srsmicro.gui.mainwindow.MainWindow.__init__(  
    self )
```

The MainWindow constructor.

### A.45.3 Member Function Documentation

#### A.45.3.1 `_create_console_dock()`

```
def srsmicro.gui.mainwindow.MainWindow._create_console_dock (  
    self ) [private]
```

Create the dock to display a console to interact with the program.

The Console widget contain within the dock inherits the **main** namespace (accesed through 'm.') giving it access to all objects. This allows the user to send commands to specific devices if needed, but should be used sparingly unless the relevant device manuals have been read. The console is hidden by default but can be accessed through the File menu.

#### A.45.3.2 `_create_explorer_dock()`

```
def srsmicro.gui.mainwindow.MainWindow._create_explorer_dock (  
    self ) [private]
```

Populate Dock GUI element containing general experiment parameters and logs.

#### A.45.3.3 `_create_expmt_tab()`

```
def srsmicro.gui.mainwindow.MainWindow._create_expmt_tab (  
    self ) [private]
```

Create the docks to display all widgets and features specific to the selected experiment.

Changes in experiment type will change which dock is displayed in the DockArea; however all docks are created immediately. This allows simple switching between experiment modes without loss of recorded data or GUI state, at the cost of a little more overhead.

#### A.45.3.4 `_create_filemenu()`

```
def srsmicro.gui.mainwindow.MainWindow._create_filemenu (  
    self ) [private]
```

Create the 'File' menu to be added to the menubar.

#### A.45.3.5 `_create_menubar()`

```
def srsmicro.gui.mainwindow.MainWindow._create_menubar (
    self ) [private]
```

Create the menubar and populate it.

#### A.45.3.6 `_create_parameters_widget()`

```
def srsmicro.gui.mainwindow.MainWindow._create_parameters_widget (
    self ) [private]
```

Create a ParameterTree for experiment parameters and add it to the `_explorer_dock` Dock widget.

#### A.45.3.7 `_openfile()`

```
def srsmicro.gui.mainwindow.MainWindow._openfile (
    self ) [private]
```

Open a file.

Not currently in use.

#### A.45.3.8 `_saveim()`

```
def srsmicro.gui.mainwindow.MainWindow._saveim (
    self ) [private]
```

Save an image.

Not currently in use.

#### A.45.3.9 `_savezistate()`

```
def srsmicro.gui.mainwindow.MainWindow._savezistate (
    self ) [private]
```

Save a configuration/state.

Not currently in use.

#### A.45.3.10 `_toggleconsole()`

```
def srsmicro.gui.mainwindow.MainWindow._toggleconsole (
    self ) [private]
```

Toggle visibility of the console.

Accessed through File menu actions.

#### A.45.3.11 `_update_expmt()`

```
def srsmicro.gui.mainwindow.MainWindow._update_expmt (
    self,
    obj,
    param,
```

`val ) [private]`

Slot for updating the experiment type parameter.

Only the final `val` parameter is used; however, the other two are needed to match the emitted signal pattern.

Parameters

|              |  |
|--------------|--|
| <i>obj</i>   | Reference to the GUI element.                  |
| <i>param</i> | The parameter that was changed.                |
| <i>val</i>   | The value to which the experiment was changed. |

#### A.45.3.12 `closeEvent()`

```
def srsmicro.gui.mainwindow.MainWindow.closeEvent (  
    self,  
    event )
```

Shutdown routine.

On clicking the 'X' for closing a message will appear asking for confirmation. Acceptance closes the application and emits signals where needed to allow proper shutdown and logging.

#### A.45.3.13 `expmt_changed()`

```
def srsmicro.gui.mainwindow.MainWindow.expmt_changed (  
    self,  
    device,  
    param,  
    val )
```

#### A.45.3.14 `global_changed()`

```
def srsmicro.gui.mainwindow.MainWindow.global_changed (  
    self,  
    obj,  
    param,  
    val )
```

Emit a signal to be parsed by the controller.

#### A.45.3.15 `parse_state()`

```
def srsmicro.gui.mainwindow.MainWindow.parse_state (  
    self,  
    device,  
    params )
```

Slot for updating device specific GUI elements, e.g.



to display a parameter change. This slot parses the signal and relays a separate signal to the appropriate GUI elements corresponding to the device.

Parameters

|               |   |
|---------------|---|
| <i>device</i> | (str) The device whose state was updated.                   |
| <i>params</i> | (dict) The parameters/settings for the device that changed. |

#### A.45.3.16 `statusbar()` [1/2]

```
def srsmicro.gui.mainwindow.MainWindow.statusbar (
    self )
```

#### A.45.3.17 `statusbar()` [2/2]

```
def srsmicro.gui.mainwindow.MainWindow.statusbar (
    self,
    msg )
```

Setter for the statusbar.

Change the message that is displayed.

#### A.45.3.18 `update_log()`

```
def srsmicro.gui.mainwindow.MainWindow.update_log (
    self,
    msg )
```

Add text to the log widget.

Parameters

|            |                                   |
|------------|-----------------------------------|
| <i>msg</i> | The text to be added to the logs. |
|------------|-----------------------------------|

### A.45.4 Member Data Documentation

#### A.45.4.1 `_console`

```
srsmicro.gui.mainwindow.MainWindow._console [private]
```

The actual console widget.

It inherits the **main** namespace as `m`. E.g. Access the controller like this: `m.controller`

#### **A.45.4.2 `_console_dock`**

`srsmicro.gui.mainwindow.MainWindow._console_dock` [private]  
Dock GUI element containing the Console Widget.

#### **A.45.4.3 `_current_panel`**

`srsmicro.gui.mainwindow.MainWindow._current_panel` [private]  
The current panel being displayed.  
Initialized to a BasicPanel but updating the experiment type updates this variable to point to a specific experiment panel

#### **A.45.4.4 `_delaystage_com`**

`srsmicro.gui.mainwindow.MainWindow._delaystage_com` [private]  
ListParameter for selecting the physical COM port of the delay stage.

#### **A.45.4.5 `_delaystage_panel`**

`srsmicro.gui.mainwindow.MainWindow._delaystage_panel` [private]  
Contains delay stage specific GUI elements.

#### **A.45.4.6 `_explorer_dock`**

`srsmicro.gui.mainwindow.MainWindow._explorer_dock` [private]  
Left aligned dock for global parameters and logs.

#### **A.45.4.7 `_expmt_type`**

`srsmicro.gui.mainwindow.MainWindow._expmt_type` [private]  
ListParameter for selecting which experiment dock is displayed.

#### **A.45.4.8 `_fssrs_panel`**

`srsmicro.gui.mainwindow.MainWindow._fssrs_panel` [private]  
Contains GUI elements for running a simple SRS experiment and displaying the images/results.

#### A.45.4.9 `_insight_com`

`srsmicro.gui.mainwindow.MainWindow._insight_com` [private]  
ListParameter for selecting the physical COM port of the Insight.

#### A.45.4.10 `_insight_panel`

`srsmicro.gui.mainwindow.MainWindow._insight_panel` [private]  
Contains Insight specific GUI elements.

#### A.45.4.11 `_parameters_widget`

`srsmicro.gui.mainwindow.MainWindow._parameters_widget` [private]  
ParameterTree widget for global parameters such as experiment type and serial communication ports.

#### A.45.4.12 `_showconsole`

`srsmicro.gui.mainwindow.MainWindow._showconsole` [private]

#### A.45.4.13 `_zi_panel`

`srsmicro.gui.mainwindow.MainWindow._zi_panel` [private]  
Contains lock-in amplifier specific GUI elements.

#### A.45.4.14 `statusbar`

`srsmicro.gui.mainwindow.MainWindow.statusbar`  
The documentation for this class was generated from the following file:

- `gui/mainwindow.py`

## A.46 `srsmicro.control.olympusappcontroller.OlympusAppController` Class Reference

### Public Member Functions

- `def __init__(self)`  
*The OlympusAppController constructor.*
- `def toggle_fvbutton(self, x, y)`

*Function to click buttons on FV1200 software GUI.*

- `def toggle_brightfield (self)`

*Function to toggle between brightfield and laser scanning modes.*

- `def set_scan_type (self, mode, n=1)`

*Function to toggle between different laser scanning modes.*

## Private Attributes

- `_brightfield`
- `_pixeldwell`
- `_zoom`
- `_mode`
- `_nframes`
- `_focus_on`

## A.46.1 Detailed Description

The `OlympusAppController` class provides tools for controlling the proprietary Olympus FV1200 software without user input. The software must be running, but this class facilitates its blind manipulation.

## A.46.2 Constructor & Destructor Documentation

### A.46.2.1 `__init__()`

```
def srsmicro.control.olympusappcontroller.OlympusAppController.__init__ (
    self )
```

The `OlympusAppController` constructor.

## A.46.3 Member Function Documentation

### A.46.3.1 `set_scan_type()`

```
def srsmicro.control.olympusappcontroller.OlympusAppController.set_scan_type (
    self,
    mode,
    n = 1 )
```

Function to toggle between different laser scanning modes.

Parameters

|                   |   |
|-------------------|---|
| <code>mode</code> | (str) Indicates the type of scanning mode to use. (xy, xyz xyt, xyzt) |
|-------------------|---|

### A.46.3.2 `toggle_brightfield()`

```
def srsmicro.control.olympusappcontroller.OlympusAppController.toggle_brightfield (
    self )
```

Function to toggle between brightfield and laser scanning modes.

### A.46.3.3 `toggle_fvbutton()`

```
def srsmicro.control.olympusappcontroller.OlympusAppController.toggle_fvbutton (
    self,
    x,
    y )
```

Function to click buttons on FV1200 software GUI.

Controls switching between open applications to select the FV1200 GUI and then return to the calling software.

Parameters

|          |   |
|----------|---|
| <i>x</i> | (int) x position of the button on screen. |
| <i>y</i> | (int) y position of the button on screen. |

## A.46.4 Member Data Documentation

### A.46.4.1 `_brightfield`

```
srsmicro.control.olympusappcontroller.OlympusAppController._brightfield [private]
```

### A.46.4.2 `_focus_on`

```
srsmicro.control.olympusappcontroller.OlympusAppController._focus_on [private]
```

### A.46.4.3 `_mode`

```
srsmicro.control.olympusappcontroller.OlympusAppController._mode [private]
```

#### A.46.4.4 `_nframes`

`srsmicro.control.olympusappcontroller.OlympusAppController._nframes` [private]

#### A.46.4.5 `_pixeldwell`

`srsmicro.control.olympusappcontroller.OlympusAppController._pixeldwell` [private]

#### A.46.4.6 `_zoom`

`srsmicro.control.olympusappcontroller.OlympusAppController._zoom` [private]

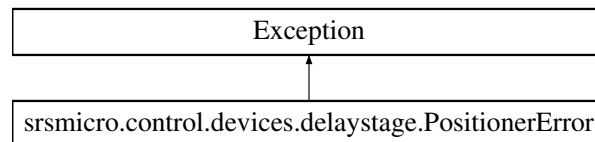
The documentation for this class was generated from the following file:

- `control/olympusappcontroller.py`

## A.47 `srsmicro.control.devices.delaystage.PositionerError` Class Reference

Exception class for positioner errors for the Newport FCL200 delay stage.

Inheritance diagram for `srsmicro.control.devices.delaystage.PositionerError`:



### Public Member Functions

- `def __init__(self, error_code)`  
*PositionError class initializer.*
- `def __str__(self)`  
*String representation of the PositionError on print.*

### Public Attributes

- `msg`

### Static Private Attributes

- `list _pos_errors`  
*(list[str]:str) String definitions for given integer error codes.*

## A.47.1 Detailed Description

Exception class for positioner errors for the Newport FCL200 delay stage.

## A.47.2 Constructor & Destructor Documentation

### A.47.2.1 `__init__()`

```
def srsmicro.control.devices.delaystage.PositionerError.__init__(
    self,
    error_code )
```

PositionError class initializer.

Parameters

|                   |  |
|-------------------|--|
| <i>error_code</i> | (list[int]) List of error codes. Definitions are accessed through class array <code>_pos_errors</code> . |
|-------------------|--|

## A.47.3 Member Function Documentation

### A.47.3.1 `__str__()`

```
def srsmicro.control.devices.delaystage.PositionerError.__str__(
    self )
```

String representation of the PositionError on print.

Returns

`self.msg (str)`

## A.47.4 Member Data Documentation

### A.47.4.1 `_pos_errors`

`srsmicro.control.devices.delaystage.PositionerError._pos_errors` [static], [private]

**Initial value:**

```
= ['Not used.',
   'Not used.',
   'Not used.',
   'Driver overheating.',
   'Driver fault.',
   'Not used.',
```

```
'Not used.',
'No parameters in memory.',
'Homing time out.',
'Not used.',
'Newport reserved.',
'RMS current limit.',
'Not used.',
'Positive end of run.',
'Negative end of run.']
```

(list[str]:str) String definitions for given integer error codes.

The definition is returned by using the error code as the index on the list.

#### A.47.4.2 msg

srsmicro.control.devices.delaystage.PositionerError.msg

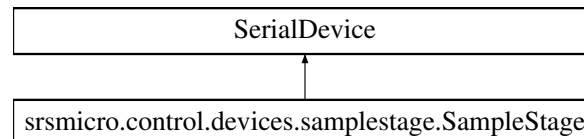
The documentation for this class was generated from the following file:

- control/devices/delaystage.py

## A.48 srsmicro.control.devices.samplestage.SampleStage Class Reference

Facilitates serial communication with Prior Scientific sample stage.

Inheritance diagram for srsmicro.control.devices.samplestage.SampleStage:



### Public Member Functions

- def \_\_init\_\_(self)

### Private Attributes

- \_baudrate

#### A.48.1 Detailed Description

Facilitates serial communication with Prior Scientific sample stage.

Sample stage is needed for Y-dimension motion while acquiring hyperspectral images, to avoid scanning spectrum across Y.

Args:



## A.48.2 Constructor & Destructor Documentation

### A.48.2.1 `__init__()`

```
def srsmicro.control.devices.samplestage.SampleStage.__init__(  
    self )
```

## A.48.3 Member Data Documentation

### A.48.3.1 `_baudrate`

```
srsmicro.control.devices.samplestage.SampleStage._baudrate [private]
```

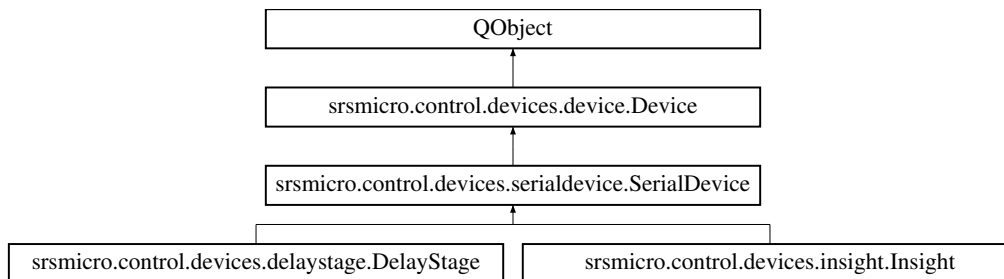
The documentation for this class was generated from the following file:

- control/devices/samplestage.py

## A.49 `srsmicro.control.devices.serialdevice.SerialDevice` Class Reference

The base class for serial devices.

Inheritance diagram for `srsmicro.control.devices.serialdevice.SerialDevice`:



## Public Member Functions

- `def __init__(self, name='Serial Device')`  
*The SerialDevice base class constructor.*
- `def write(self, cmd, waittime)`  
*Write a command to the serial device.*
- `def read(self)`  
*Read back an answer from the serial device.*

- def baudrate (self)  
*Property for the communication baud rate.*
- def baudrate (self, val)  
*Property setter for the communication baud rate.*
- def comport (self)  
*Property for the communication port.*
- def comport (self, val)  
*Property setter for changing the communication port.*
- def exit (self)  
*Device shutdown method.*

## Private Member Functions

- def \_open (self)  
*Private function which opens communication over a serial port.*
- def \_close (self)  
*Private function which closes communication over a serial port.*

## Private Attributes

- \_sercom  
*Serial object to manage communication.*
- \_isconnected  
*(Boolean) Is the serial port open or closed.*
- \_baudrate  
*Baud rate for serial communication.*
- \_comport  
*Serial communication port.*

## Additional Inherited Members

### A.49.1 Detailed Description

The base class for serial devices.

Defines the base class used by devices whose serial communication is managed directly.

Currently includes:

- Prior sample stage
- Insight DS+ femtosecond laser/OPO
- Newport FCL200 optical delay stage

## A.49.2 Constructor & Destructor Documentation

### A.49.2.1 `__init__()`

```
def srsmicro.control.devices.serialdevice.SerialDevice.__init__ (
    self,
    name = 'Serial Device' )
```

The SerialDevice base class constructor.

Reimplemented from `srsmicro.control.devices.device.Device`.

Reimplemented in `srsmicro.control.devices.delaystage.DelayStage`, and `srsmicro.control.devices.insight.Insight`.

## A.49.3 Member Function Documentation

### A.49.3.1 `_close()`

```
def srsmicro.control.devices.serialdevice.SerialDevice._close (
    self ) [private]
```

Private function which closes communication over a serial port.

Reimplemented from `srsmicro.control.devices.device.Device`.

### A.49.3.2 `_open()`

```
def srsmicro.control.devices.serialdevice.SerialDevice._open (
    self ) [private]
```

Private function which opens communication over a serial port.

Reimplemented from `srsmicro.control.devices.device.Device`.

### A.49.3.3 `baudrate()` [1/2]

```
def srsmicro.control.devices.serialdevice.SerialDevice.baudrate (
    self )
```

Property for the communication baud rate.

Returns

`_baudrate` (int) Communication baud rate.

#### A.49.3.4 `baudrate()` [2/2]

```
def srsmicro.control.devices.serialdevice.SerialDevice.baudrate (
    self,
    val )
```

Property setter for the communication baud rate.

Parameters

|            |                  |
|------------|------------------|
| <i>val</i> | (int) Baud rate. |
|------------|------------------|

#### A.49.3.5 `comport()` [1/2]

```
def srsmicro.control.devices.serialdevice.SerialDevice.comport (
    self )
```

Property for the communication port.

Returns

`_comport (str)` Communication port. E.g. 'COM1'

#### A.49.3.6 `comport()` [2/2]

```
def srsmicro.control.devices.serialdevice.SerialDevice.comport (
    self,
    val )
```

Property setter for changing the communication port.

Parameters

|            |   |
|------------|---|
| <i>val</i> | (str) New communication port. E.g. 'COM1' |
|------------|---|

#### A.49.3.7 `exit()`

```
def srsmicro.control.devices.serialdevice.SerialDevice.exit (
    self )
```

Device shutdown method.

Overwritten by subclasses.

Reimplemented from `srsmicro.control.devices.device.Device`.

Reimplemented in `srsmicro.control.devices.insight.Insight`.

### A.49.3.8 read()

```
def srsmicro.control.devices.serialdevice.SerialDevice.read (
    self )
```

Read back an answer from the serial device.

Returns

(str) The ASCII decoded result from reading back from the device.

### A.49.3.9 write()

```
def srsmicro.control.devices.serialdevice.SerialDevice.write (
    self,
    cmd,
    waittime )
```

Write a command to the serial device.

Parameters

|                 |   |
|-----------------|---|
| <i>cmd</i>      | (str) The command to be written   |
| <i>waittime</i> | (float) The time to wait after writing a command. Blocks communication, preventing writing multiple commands before the device can respond. |

## A.49.4 Member Data Documentation

### A.49.4.1 `_baudrate`

`SerialDevice._baudrate` [private]

Baud rate for serial communication.

### A.49.4.2 `_comport`

`SerialDevice._comport` [private]

Serial communication port.

E.g. 'COM1'

### A.49.4.3 `_isconnected`

`SerialDevice._isconnected` [private]

(Boolean) Is the serial port open or closed.

#### A.49.4.4 `_sercom`

`SerialDevice._sercom` [private]

Serial object to manage communication.

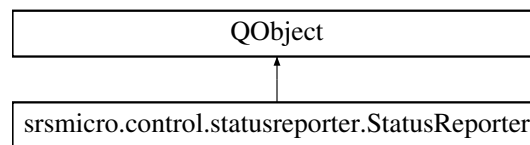
The documentation for this class was generated from the following file:

- `control/devices/serialdevice.py`

## A.50 `srsmicro.control.statusreporter.StatusReporter` Class Reference

Worker class for device status querying.

Inheritance diagram for `srsmicro.control.statusreporter.StatusReporter`:



### Public Member Functions

- `def __init__(self, devices)`  
*The StatusReporter constructor.*
- `def query_state(self)`  
*Method to query device state.*
- `def add_device(self, device)`  
*Method to add a device to the query list if it was not included during object initialization.*

### Public Attributes

- `devices`  
*(list) List of device references.*
- `pause`  
*(bool) Variable to block device status queries.*

### Static Public Attributes

- `shutdown = Signal()`

## A.50.1 Detailed Description

Worker class for device status querying.

Executes query functions for every device reference it is passed.

## A.50.2 Constructor & Destructor Documentation

### A.50.2.1 `__init__()`

```
def srsmicro.control.statusreporter.StatusReporter.__init__(
    self,
    devices )
```

The StatusReporter constructor.

Parameters

|                |   |
|----------------|---|
| <i>devices</i> | (List) A list of all devices to have their state queried. |
|----------------|---|

## A.50.3 Member Function Documentation

### A.50.3.1 `add_device()`

```
def srsmicro.control.statusreporter.StatusReporter.add_device (
    self,
    device )
```

Method to add a device to the query list if it was not included during object initialization.

Parameters

|               |   |
|---------------|---|
| <i>device</i> | (Device) The device object to be added to the list. |
|---------------|---|

### A.50.3.2 `query_state()`

```
def srsmicro.control.statusreporter.StatusReporter.query_state (
    self )
```

Method to query device state.

Runs infinitely, unless an interrupt signal is received, asking devices to emit their state conditions.

## A.50.4 Member Data Documentation

### A.50.4.1 devices

`srsmicro.control.statusreporter.StatusReporter.devices`

(list) List of device references.

Used only for querying status.

### A.50.4.2 pause

`srsmicro.control.statusreporter.StatusReporter.pause`

(bool) Variable to block device status queries.

Set to true while the controller/device object is communicating with the physical device.

### A.50.4.3 shutdown

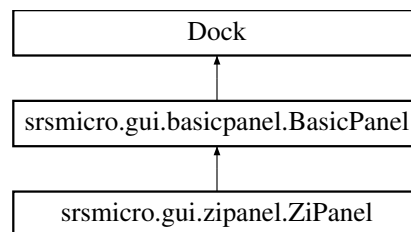
`srsmicro.control.statusreporter.StatusReporter.shutdown = Signal()` [static]

The documentation for this class was generated from the following file:

- `control/statusreporter.py`

## A.51 srsmicro.gui.zipanel.ZiPanel Class Reference

Inheritance diagram for `srsmicro.gui.zipanel.ZiPanel`:



### Public Member Functions

- `def __init__(self, *args, **kwargs)`

*The ZiPanel constructor.*

### Public Attributes

- `funcs`



## Private Member Functions

- `def _update_cond_vars (self)`

## Additional Inherited Members

### A.51.1 Constructor & Destructor Documentation

#### A.51.1.1 `__init__()`

```
def srsmicro.gui.zipanel.ZiPanel.__init__ (
    self,
    * args,
    ** kwargs )
```

The ZiPanel constructor.

Parameters

|                 |   |
|-----------------|---|
| <i>statpath</i> | (str) The path to the text file containing the GUI's status elements. These elements only display information.                      |
| <i>ctrlpath</i> | (str) The path to the text file containing the GUI's control elements. These elements allow interaction with the associated device. |

Reimplemented from `srsmicro.gui.basicpanel.BasicPanel`.

### A.51.2 Member Function Documentation

#### A.51.2.1 `_update_cond_vars()`

```
def srsmicro.gui.zipanel.ZiPanel._update_cond_vars (
    self ) [private]
```

### A.51.3 Member Data Documentation

#### A.51.3.1 `funcs`

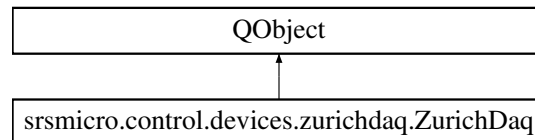
`srsmicro.gui.zipanel.ZiPanel.funcs`

The documentation for this class was generated from the following file:

- `gui/zipanel.py`

## A.52 srsmicro.control.devices.zurichdaq.ZurichDaq Class Reference

The class for managing the Lock-in amplifier's data acquisition module.  
Inheritance diagram for srsmicro.control.devices.zurichdaq.ZurichDaq:



### Public Member Functions

- `def __init__ (self, daq, devname, path)`

*The ZurichDaq constructor.*

- `def setup_scan (self, rows=512, cols=512, repetitions=1, interp=2, delay=0)`
- `def setup_trigger (self, type=1, node='auxin0', edge=1, level=2.5)`

*The trigger inputs.*

- `def start (self)`
- `def read_daq (self)`
- `def stop (self)`
- `def stop (self, val)`
- `def exit (self)`

### Static Public Attributes

- `data = Signal(object)`
- `shutdown = Signal()`

### Private Attributes

- `_daq`
- `_devname`
- `_path`
- `_pixeldwell`
- `_stop`
- `_parameters`

*(list[list[str]]) list containing lists with the path to a parameter and its setting*

## A.52.1 Detailed Description

The class for managing the Lock-in amplifier's data acquisition module. Intended for management on a background thread. Multiple ZurichDaqs can be created to sample from different data streams. Wraps the DAQ module from the ZI API for smoother integration with the Qt framework.

## A.52.2 Constructor & Destructor Documentation

### A.52.2.1 `__init__()`

```
def srsmicro.control.devices.zurichdaq.ZurichDaq.__init__ (
    self,
    daq,
    devname,
    path )
```

The ZurichDaq constructor.

Parameters

|                |  |
|----------------|--|
| <i>devname</i> | (str) The lock-in device name, needed for the parameter hierarchy. |
| <i>daq</i>     | (dataAcquisitionModule) A lock-in dataAcquisitionModule.           |
|                |  |

## A.52.3 Member Function Documentation

### A.52.3.1 `exit()`

```
def srsmicro.control.devices.zurichdaq.ZurichDaq.exit (
    self )
```

### A.52.3.2 `read_daq()`

```
def srsmicro.control.devices.zurichdaq.ZurichDaq.read_daq (
    self )
```

### A.52.3.3 `setup_scan()`

```
def srsmicro.control.devices.zurichdaq.ZurichDaq.setup_scan (
    self,
    rows = 512,
```

```

    cols = 512,
    repetitions = 1,
    interp = 2,
    delay = 0 )

```

#### A.52.3.4 setup\_trigger()

```

def srsmicro.control.devices.zurichdaq.ZurichDaq.setup_trigger (
    self,
    type = 1,
    node = 'auxin0',
    edge = 1,
    level = 2.5 )

```

The trigger inputs.

On object construction the DAQ is set to run without trigger. This method allows setting an input trigger from, e.g., the Olympus. Defaults parameter values are set for the most common configuration used currently.

Parameters

|              |  |
|--------------|--|
| <i>type</i>  | (int) Trigger type. See <code>_parameters</code> documentation. Default: 1 (edge)      |
| <i>node</i>  | (str) Input node for the trigger. Default: 'auxin1' (Auxillary input 1 on the lock-in) |
| <i>edge</i>  | (int) Which edge of the signal to trigger on. Default: 1 (positive)                    |
| <i>level</i> | (float) Voltage level for the trigger. Default: 2.5 (V)                                |

#### A.52.3.5 start()

```

def srsmicro.control.devices.zurichdaq.ZurichDaq.start (
    self )

```

#### A.52.3.6 stop() [1/2]

```

def srsmicro.control.devices.zurichdaq.ZurichDaq.stop (
    self )

```

#### A.52.3.7 stop() [2/2]

```

def srsmicro.control.devices.zurichdaq.ZurichDaq.stop (
    self,
    val )

```

## A.52.4 Member Data Documentation

### A.52.4.1 `_daq`

`srsmicro.control.devices.zurichdaq.ZurichDaq._daq` [private]

### A.52.4.2 `_devname`

`srsmicro.control.devices.zurichdaq.ZurichDaq._devname` [private]

### A.52.4.3 `_parameters`

`srsmicro.control.devices.zurichdaq.ZurichDaq._parameters` [private]

(list[list[str]]) list containing lists with the path to a parameter and its setting

This list is continually rewritten to as it is only used to pass the parameters to the DAQ; maintaining a copy allows retrieving the last settings that were attempted for debugging. Key parameters:

- `type` - indicates triggering type, each trigger has its own settings. Refer to the Zurich API
  - 0: Trigger off
  - 1: Analog edge trigger on source
  - 2: Digital trigger on DIO
  - 3: Analog pulse trigger on source
  - 4: Analog tracking trigger on source
  - 5: Change trigger
  - 6: Hardware trigger on trigger line source
  - 7: Tracking edge trigger on source
  - 8: Event count trigger on counter source
- `duration` - The duration of one row of the scan (`dwelldwell*num_pixels`)

### A.52.4.4 `_path`

`srsmicro.control.devices.zurichdaq.ZurichDaq._path` [private]

### A.52.4.5 `_pixeldwell`

`srsmicro.control.devices.zurichdaq.ZurichDaq._pixeldwell` [private]

#### A.52.4.6 `_stop`

```
srsmicro.control.devices.zurichdaq.ZurichDaq._stop [private]
```

#### A.52.4.7 `data`

```
srsmicro.control.devices.zurichdaq.ZurichDaq.data = Signal(object) [static]
```

#### A.52.4.8 `shutdown`

```
srsmicro.control.devices.zurichdaq.ZurichDaq.shutdown = Signal() [static]
```

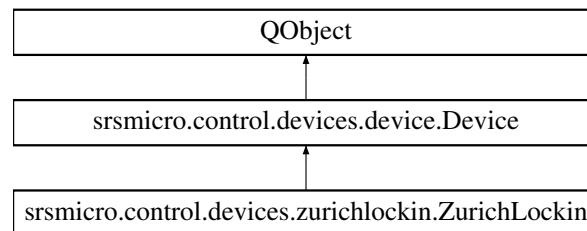
The documentation for this class was generated from the following file:

- `control/devices/zurichdaq.py`

## A.53 `srsmicro.control.devices.zurichlockin.ZurichLockin` Class Reference

The base class for the ZI HF2LI lock-in amplifier.

Inheritance diagram for `srsmicro.control.devices.zurichlockin.ZurichLockin`:



### Public Member Functions

- `def __init__ (self, name='Lockin')`  
*The ZurichLockin base class initializer.*
- `def start_daq (self)`
- `def stop_daq (self)`
- `def exit (self)`  
*Device shutdown method.*
- `def scope (self)`
- `def scope_time (self)`
- `def scope_time (self, val)`
- `def tc (self)`

*Property and setter functions for lockin time constant, modulation frequency and sampling rate.*

- def tc (self, val)
- def freq (self)
- def freq (self, val)
- def rate (self)
- def rate (self, val)

## Public Attributes

- last\_action

## Static Public Attributes

- data = Signal(object, object)

## Private Member Functions

- def \_load\_variables (self)

*Load in the parameter hierarchy from the included configuration file.*

- def \_open (self)

*Run the ZI API discovery routine and attempt to start the lock-in amplifier server.*

- def \_enable\_demod (self, demod=0, sign=0, freq=1.028e7, harm=1, tc=3e-6, order=4, osc=0, rate=100000)

*Enable a demodulator for signal processing.*

- def \_configure\_sign (self, sign=0, ac=1, imp50=1, diff=0, range=.01)

*Configure signal in parameters.*

- def \_configure\_sigout (self, sigout=0, on=1, add=0, range=10)

*Configure signal out parameters.*

- def \_get\_config (self)

*Return current lock-in parameter configuration.*

- def \_image\_data (self, data)

- def \_poll (self, demod, sign, poll\_length=0.05, timeout=500, tc=1e-3)

*Poll a demodulator and record the data.*

- def \_poll\_scope (self, channel)

- def \_check\_api\_errors (self)

*Property and setter functions for signal input/outputs.*

## Private Attributes

- `_cond_vars_list`  
*(list[list]) All possible ZI parameters in list of list format.*
- `_cond_vars`  
*(dict[str]:int/float/long) `_cond_vars_list` converted to dictionary format for easier parameter accession and to match the format used by other devices.*
- `_devname`  
*Actual hardware device name for internal use with the server API.*
- `_server`
- `_daq_thread`
- `_daq`
- `_scope`
- `_tc`
- `_freq`
- `_rate`
- `_api_error`

### A.53.1 Detailed Description

The base class for the ZI HF2LI lock-in amplifier.

Uses a separate server for communication through ZI's provided API. For simplicity, and reproducibility, ZI API calls are handled behind the scenes, so the end user experience mimics using the `SerialDevice` class.

### A.53.2 Constructor & Destructor Documentation

#### A.53.2.1 `__init__()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.__init__(  
    self,  
    name = 'Lockin' )
```

The `ZurichLockin` base class initializer.

Reimplemented from `srsmicro.control.devices.device.Device`.

### A.53.3 Member Function Documentation



### A.53.3.1 `_check_api_errors()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin._check_api_errors (
    self ) [private]
```

Property and setter functions for signal input/outputs.

API errors

/dev1292/status/flags

Check if any API errors were found

### A.53.3.2 `_configure_sigin()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin._configure_sigin (
    self,
    sigin = 0,
    ac = 1,
    imp50 = 1,
    diff = 0,
    range = .01 ) [private]
```

Configure signal in parameters.

Parameters

|              |  |
|--------------|--|
| <i>sigin</i> | (int) Index of signal input to configure. Default: 0             |
| <i>ac</i>    | (int) Whether to enable AC coupling. Default: 1 (Enable)         |
| <i>imp50</i> | (int) Whether input impedance is set to 50 ohm. Default: 1 (Yes) |
| <i>diff</i>  | (int) Whether using differential input. Default: 0 (No)          |
| <i>range</i> | (float) Voltage range of signal in V (0.0001). Default: 0.01     |

### A.53.3.3 `_configure_sigout()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin._configure_sigout (
    self,
    sigout = 0,
    on = 1,
    add = 0,
    range = 10 ) [private]
```

Configure signal out parameters.

Defaults to beginning output, so should only be called to begin output, or then to disable it afterwards.

Parameters

|               |   |
|---------------|---|
| <i>sigout</i> | (int) Index of signal output to configure. Default: 0 |
|---------------|---|

## Parameters

|              |   |
|--------------|---|
| <i>on</i>    | (int) Toggle signal output on or off. Default: 1 (On)     |
| <i>add</i>   | (int) Toggle the signal adder on or off. Default: 0 (Off) |
| <i>range</i> | (float) Output range [0.01, 0.1, 1, 10]. Default: 10      |

### A.53.3.4 `_enable_demod()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin._enable_demod (
    self,
    demod = 0,
    sigin = 0,
    freq = 1.028e7,
    harm = 1,
    tc = 3e-6,
    order = 4,
    osc = 0,
    rate = 100000 ) [private]
```

Enable a demodulator for signal processing.

## Parameters

|              |  |
|--------------|--|
| <i>demod</i> | (int) Index of demodulator, [0, 5]. Default: 0   |
| <i>sigin</i> | (int) Index of signal input, [0, 5]. Default: 0 (Signal In 1)  |
| <i>freq</i>  | (float) Demodulation frequency (Hz). Default: 10280000   |
| <i>harm</i>  | (int) Demodulation harmonic. Default: 1  |
| <i>tc</i>    | (float) Lock-in time constant in seconds. Default: 3e-6  |
| <i>order</i> | (int) Low-pass filter order [1,8]. Default 4 (24 dB/oct slope)   |
| <i>osc</i>   | (int) Oscillator to use 0 or 1. Default 0  |
| <i>rate</i>  | (int) Data transfer rate (Hz), may be approximated by LIA. Default: 10000 (>2 samples/pixel at 512 pixels with dwell=tc) |

### A.53.3.5 `_get_config()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin._get_config (
    self ) [private]
```

Return current lock-in parameter configuration.

**Todo** Define scope of return from tree hierarchy (i.e. which node to start at)

Parse dictionary to more readable form

Actually implement above. Add to logs? Or have a separate output?

### A.53.3.6 `_image_data()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin._image_data (
    self,
    data ) [private]
```

### A.53.3.7 `_load_variables()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin._load_variables (
    self ) [private]
```

Load in the parameter hierarchy from the included configuration file.

One copy is loaded into a list-of-lists format which can be passed directly to ZI API objects for setting parameters. A separate copy is converted to a dictionary which matches the format used by other devices for accessing parameters. The config file is also used by ZI gui element.

### A.53.3.8 `_open()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin._open (
    self ) [private]
```

Run the ZI API discovery routine and attempt to start the lock-in amplifier server.

Reimplemented from `srsmicro.control.devices.device.Device`.

### A.53.3.9 `_poll()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin._poll (
    self,
    demod,
    sigin,
    poll_length = 0.05,
    timeout = 500,
    tc = 1e-3 ) [private]
```

Poll a demodulator and record the data.

#### Args:

`poll_length` (float): how long to poll. Units: (s)  
`timeout` (int): timeout period for response from server. Units (ms)  
`tc` (float): demodulator time constant with which to poll. Units (s)

#### Returns:

`x` (np array): demodulator x values over polling period.  
`y` (np array): demodulator y values over polling period.  
`frame` (np array): auxiliary in 0 values. Currently configured to olympus frame clock.  
`line` (np array): auxiliary in 1 values. Currently configured to olympus line clock.

#### A.53.3.10 `_poll_scope()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin._poll_scope (
    self,
    channel ) [private]
```

Poll the oscilloscope. Not currently in use.

#### A.53.3.11 `exit()`

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.exit (
    self )
```

Device shutdown method.

Overwritten by subclasses.

Reimplemented from `srsmicro.control.devices.device.Device`.

#### A.53.3.12 `freq()` [1/2]

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.freq (
    self )
```

Property to return current oscillator frequency.

#### A.53.3.13 `freq()` [2/2]

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.freq (
    self,
    val )
```

Oscillator frequency setter.

Args:

```
val (float): oscillator frequency. Units (Hz)
```

#### A.53.3.14 `rate()` [1/2]

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.rate (
    self )
```

Property to return current sampling rate of demodulated signal.

#### **A.53.3.15 rate() [2/2]**

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.rate (
    self,
    val )
```

Sampling rate setter.

Args:

```
val (float): sampling rate of demodulated signal. Units (Sa/s)
```

#### **A.53.3.16 scope()**

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.scope (
    self )
```

Return the oscilloscope trace.

#### **A.53.3.17 scope\_time() [1/2]**

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.scope_time (
    self )
```

#### **A.53.3.18 scope\_time() [2/2]**

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.scope_time (
    self,
    val )
```

#### **A.53.3.19 start\_daq()**

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.start_daq (
    self )
```

#### **A.53.3.20 stop\_daq()**

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.stop_daq (
    self )
```

#### A.53.3.21 `tc()` [1/2]

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.tc (
    self )
```

Property and setter functions for lockin time constant, modulation frequency and sampling rate.

Property to return current demodulator time constant.

#### A.53.3.22 `tc()` [2/2]

```
def srsmicro.control.devices.zurichlockin.ZurichLockin.tc (
    self,
    val )
```

Time constant setter.

Args:

```
val (float): demodulator time constant. Units (s)
```

### A.53.4 Member Data Documentation

#### A.53.4.1 `_api_error`

```
srsmicro.control.devices.zurichlockin.ZurichLockin._api_error [private]
```

#### A.53.4.2 `_cond_vars`

```
srsmicro.control.devices.zurichlockin.ZurichLockin._cond_vars [private]
```

(dict[str]:int/float/long) `_cond_vars_list` converted to dictionary format for easier parameter accession and to match the format used by other devices.

#### A.53.4.3 `_cond_vars_list`

```
srsmicro.control.devices.zurichlockin.ZurichLockin._cond_vars_list [private]
```

(list[list]) All possible ZI parameters in list of list format.  
Each list is of the format [str, int/float/long] where the first entry is a parameter path, and the second is a parameter setting.

#### A.53.4.4 `_daq`

```
srsmicro.control.devices.zurichlockin.ZurichLockin._daq [private]
```

#### **A.53.4.5** `_daq_thread`

`srsmicro.control.devices.zurichlockin.ZurichLockin._daq_thread` [private]

#### **A.53.4.6** `_devname`

`srsmicro.control.devices.zurichlockin.ZurichLockin._devname` [private]  
Actual hardware device name for internal use with the server API.

#### **A.53.4.7** `_freq`

`srsmicro.control.devices.zurichlockin.ZurichLockin._freq` [private]

#### **A.53.4.8** `_rate`

`srsmicro.control.devices.zurichlockin.ZurichLockin._rate` [private]

#### **A.53.4.9** `_scope`

`srsmicro.control.devices.zurichlockin.ZurichLockin._scope` [private]

#### **A.53.4.10** `_server`

`srsmicro.control.devices.zurichlockin.ZurichLockin._server` [private]

#### **A.53.4.11** `_tc`

`srsmicro.control.devices.zurichlockin.ZurichLockin._tc` [private]

#### **A.53.4.12** `data`

`srsmicro.control.devices.zurichlockin.ZurichLockin.data = Signal(object, object)`  
[static]

#### **A.53.4.13** `last_action`

`srsmicro.control.devices.zurichlockin.ZurichLockin.last_action`  
The documentation for this class was generated from the following file:

- `control/devices/zurichlockin.py`

## A.54 File Documentation

### A.55 `__main__.py` File Reference

#### Namespaces

- namespace `srsmicro`
- namespace `srsmicro.__main__`

*The main GUI application entry point managing the top level GUI, controller, and data object instances and their Signal/Slot connections.*

#### Variables

- `srsmicro.__main__.app = QtWidgets.QApplication(sys.argv)`
- `srsmicro.__main__.mw = MainWindow()`
- `srsmicro.__main__.controller = MainController()`

### A.56 `_version.py` File Reference

#### Namespaces

- namespace `srsmicro`
- namespace `srsmicro._version`

#### Variables

- `string srsmicro._version._version__ = '0.1'`

### A.57 `control/devices/cppkcube.cpp` File Reference

```
#include "cppkcube.hpp"
```

### A.58 `control/devices/cppkcube.hpp` File Reference

```
#include <stdlib.h>  
#include <stdio.h>  
#include <conio.h>  
#include "Thorlabs.MotionControl.KCube.DCServo.h"
```



## Classes

- class thorlabs::CppKcube

## Namespaces

- namespace thorlabs

## A.59 cppkcube.hpp

Go to the documentation of this file.

```
1 #ifndef KCUBE_HPP_INCLUDED
2 #define KCUBE_HPP_INCLUDED
3
4 #include <stdlib.h>
5 #include <cstdio>
6 #include <conio.h>
7 #include "Thorlabs.MotionControl.KCube.DCServo.h"
8
9 namespace thorlabs
10 {
11     class CppKcube
12     {
13     public:
14         // Methods
15         CppKcube(int poll_time);
16         void cpp_close();
17
18         int cpp_move(int newpos);
19         int cpp_home();
20         int cpp_set_vel(int newvel);
21         int cpp_set_accel(int newaccel);
22
23         int cpp_pos();
24         int cpp_vel();
25         int cpp_accel();
26
27     private:
28         // Attributes
29         char _cpp_serialNo[9];
30         char _cpp_desc[65];
31         int _cpp_pos;
32         int _cpp_vel;
33         int _cpp_accel;
34     };
35 }
36
37 #endif
```

## A.60 control/devices/cppspec.cpp File Reference

```
#include "cppspec.hpp"
```

## A.61 control/devices/cppspec.hpp File Reference

```
#include <stdlib.h>
#include <cstdio>
#include <conio.h>
```

### Classes

- class optosky::CppSpec

### Namespaces

- namespace optosky

## A.62 cppspec.hpp

Go to the documentation of this file.

```
1 #ifndef SPEC_HPP_INCLUDED
2 #define SPEC_HPP_INCLUDED
3
4 #include <stdlib.h>
5 #include <cstdio>
6 #include <conio.h>
7
8 namespace optosky
9 {
10     class CppSpec
11     {
12     public:
13         // Methods
14         CppSpec();
15         int cpp_acquire();
16         int cpp_close();
17
18         int cpp_set_exposeTime();
19         int cpp_set_averages();
20
21     private:
22         // Attributes
23         HP2000_wrapper::HP2000Wrapper^ m_spec;
24         HP2000_wrapper::Spectrum _data;
25         bool _is_open;
26         double _wavelength[2048];
27         double _spectrum[2048];
28         int _exposureTime; // In ms
29         int _avgs;
30     };
31 }
32
33 #endif
```

## A.63 control/devices/delaystage.py File Reference

### Classes

- class srsmicro.control.devices.delaystage.DelayStage  
*The DelayStage class for controlling the Newport FCL200 delay stage.*
- class srsmicro.control.devices.delaystage.PositionerError  
*Exception class for positioner errors for the Newport FCL200 delay stage.*
- class srsmicro.control.devices.delaystage.CommandError  
*Exception class for command errors for the Newport FCL200 delay stage.*

### Namespaces

- namespace srsmicro
- namespace srsmicro.control
- namespace srsmicro.control.devices
- namespace srsmicro.control.devices.delaystage  
*Definition of the DelayStage class for interaction with the Newport FCL200 delay stage as well as custom exception classes PositionerError and CommandError for parsing the corresponding error codes.*

## A.64 control/devices/device.py File Reference

### Classes

- class srsmicro.control.devices.device.Device  
*The base device class.*

### Namespaces

- namespace srsmicro
- namespace srsmicro.control
- namespace srsmicro.control.devices
- namespace srsmicro.control.devices.device  
*Definition of base class for communicating with physical devices.*

## A.65 control/devices/insight.py File Reference

### Classes

- class srsmicro.control.devices.insight.Insight  
*The Insight class for controlling the SpectraPhysics Insight DS+ femtosecond laser/OPO.*

## Namespaces

- namespace srsmicro
- namespace srsmicro.control
- namespace srsmicro.control.devices
- namespace srsmicro.control.devices.insight

*Definition of the Insight class for interacting with the Insight DS+ femtosecond laser.*

## A.66 control/devices/samplestage.py File Reference

### Classes

- class srsmicro.control.devices.samplestage.SampleStage

*Facilitates serial communication with Prior Scientific sample stage.*

## Namespaces

- namespace srsmicro
- namespace srsmicro.control
- namespace srsmicro.control.devices
- namespace srsmicro.control.devices.samplestage

*Definition of the SampleStage class for interaction with the Prior Scientific sample XY stage.*

## A.67 control/devices/serialdevice.py File Reference

### Classes

- class srsmicro.control.devices.serialdevice.SerialDevice

*The base class for serial devices.*

## Namespaces

- namespace srsmicro
- namespace srsmicro.control
- namespace srsmicro.control.devices
- namespace srsmicro.control.devices.serialdevice

*Definition of base class for devices using serial communication.*

## A.68 control/devices/zurichdaq.py File Reference

### Classes

- class srsmicro.control.devices.zurichdaq.ZurichDaq

*The class for managing the Lock-in amplifier's data acquisition module.*

### Namespaces

- namespace srsmicro
- namespace srsmicro.control
- namespace srsmicro.control.devices
- namespace srsmicro.control.devices.zurichdaq

*Definition of the ZurichDaq class for managing the dataAcquisitionModule for the ZI API asynchronously.*

## A.69 control/devices/zurichlockin.py File Reference

### Classes

- class srsmicro.control.devices.zurichlockin.ZurichLockin

*The base class for the ZI HF2LI lock-in amplifier.*

- class srsmicro.control.devices.zurichlockin.DeviceNotFound

*Exception class when unable to find Lockin devices connected.*

- class srsmicro.control.devices.zurichlockin.APIError

*Exception class for errors due to ZI API.*

### Namespaces

- namespace srsmicro
- namespace srsmicro.control
- namespace srsmicro.control.devices
- namespace srsmicro.control.devices.zurichlockin

*Definition of the ZurichLockin class for interaction with the Zurich Instruments HF2LI lock-in amplifier as well as custom exception classes APIError and DeviceNotFoundError for zhinst API errors.*

## A.70 control/maincontroller.py File Reference

### Classes

- class srsmicro.control.maincontroller.MainController

*The MainController class for interfacing the GUI elements with device objects.*

## Namespaces

- namespace srsmicro
- namespace srsmicro.control
- namespace srsmicro.control.maincontroller

## A.71 control/olympusappcontroller.py File Reference

### Classes

- class srsmicro.control.olympusappcontroller.OlympusAppController

### Namespaces

- namespace srsmicro
- namespace srsmicro.control
- namespace srsmicro.control.olympusappcontroller

*Definition of the OlympusAppController class which has hard-coded routines for managing the proprietary FV1200 Olympus software, without user input.*

## A.72 control/statusreporter.py File Reference

### Classes

- class srsmicro.control.statusreporter.StatusReporter

*Worker class for device status querying.*

### Namespaces

- namespace srsmicro
- namespace srsmicro.control
- namespace srsmicro.control.statusreporter

*Definition of the StatusReporter class which is intended as a worker on a separate thread for reporting device status updates.*

## A.73 gui/basicpanel.py File Reference

### Classes

- class srsmicro.gui.basicpanel.BasicPanel

*The BasicPanel serves as a template class for device specific GUI elements.*

## Namespaces

- namespace srsmicro
- namespace srsmicro.gui
- namespace srsmicro.gui.basicpanel

*Definition of the BasicPanel class for providing the parsing functions to populate device specific GUI panels.*

## A.74 gui/delaystagepanel.py File Reference

### Classes

- class srsmicro.gui.delaystagepanel.DelayStagePanel

*The DelayStagePanel provides the GUI elements for interacting with the Newport FCL200 optical delay stage.*

### Namespaces

- namespace srsmicro
- namespace srsmicro.gui
- namespace srsmicro.gui.delaystagepanel

*Definition of the DelayStagePanel class for providing the necessary GUI elements for controlling the Newport FCL200 delay stage's parameters.*

## A.75 gui/femtoramanpanel.py File Reference

### Classes

- class srsmicro.gui.femtoramanpanel.FemtoRamanPanel

*The FemtoRamanPanel provides the GUI elements needed for performing femtosecond SRS microscopy measurements and displaying the results.*

### Namespaces

- namespace srsmicro
- namespace srsmicro.gui
- namespace srsmicro.gui.femtoramanpanel

*Definition of the FemtoRamanPanel class for providing the necessary GUI elements for controlling a fsSRS microscopy experiment.*

## **A.76** `__init__.py` File Reference

## **A.77** `control/__init__.py` File Reference

## **A.78** `control/devices/__init__.py` File Reference

## **A.79** `gui/__init__.py` File Reference

## **A.80** `gui/format/__init__.py` File Reference

## **A.81** `utilities/__init__.py` File Reference

## **A.82** `gui/insightpanel.py` File Reference

### **Classes**

- class `srsmicro.gui.insightpanel.InsightPanel`

*The `InsightPanel` provides the GUI elements for interacting with the `Insight DS+`.*

### **Namespaces**

- namespace `srsmicro`
- namespace `srsmicro.gui`
- namespace `srsmicro.gui.insightpanel`

*Definition of the `InsightPanel` class for providing the necessary GUI elements for controlling the `Insight DS+` parameters and device state.*

## **A.83** `gui/mainwindow.py` File Reference

### **Classes**

- class `srsmicro.gui.mainwindow.MainWindow`

*The main SRS microscope window populated with the necessary elements for various experiment types and loading of data/projects.*

### **Namespaces**

- namespace `srsmicro`
- namespace `srsmicro.gui`
- namespace `srsmicro.gui.mainwindow`

*Definition of the `MainWindow` class for organization and arrangement of all GUI elements.*



## A.84 gui/zipanel.py File Reference

### Classes

- class srsmicro.gui.zipanel.ZiPanel

### Namespaces

- namespace srsmicro
- namespace srsmicro.gui
- namespace srsmicro.gui.zipanel

*Definition of the ZiPanel class for providing the necessary GUI elements for interacting with the Zurich Instruments HF2LI lock-in amplifier.*

## A.85 README.md File Reference

## A.86 setup.py File Reference

### Namespaces

- namespace srsmicro
- namespace srsmicro.setup

### Variables

- srsmicro.setup.name
- srsmicro.setup.version
- srsmicro.setup.packages
- srsmicro.setup.entry\_points

## A.87 utilities/conversions.py File Reference

### Namespaces

- namespace srsmicro
- namespace srsmicro.utilities
- namespace srsmicro.utilities.conversions

## Functions

- `def srsmicro.utilities.conversions.calc_omega (w_p, w_s=1040)`  
*Calculate the wavenumber given a pump and Stokes wavelength in nm.*
- `def srsmicro.utilities.conversions.logstotext (h5, output)`  
*Produces a readable text file of recorded experimental logs.*
- `def srsmicro.utilities.conversions.load_zi_yaml (self, path)`  
*Load in the parameter hierarchy from the included configuration file.*

## A.88 utilities/experimentresult.py File Reference

### Classes

- `class srsmicro.utilities.experimentresult.ExperimentResult`

### Namespaces

- namespace `srsmicro`
- namespace `srsmicro.utilities`
- namespace `srsmicro.utilities.experimentresult`

*Definition of `ExperimentResult` class which manages all data generated, useful metadata such as experimental parameters, and any log files generated.*