

University of Alberta

**Advanced Materials and Detection Methods in Ultrathin-Layer
Chromatography**

by

Anthony James Oko

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

in

Microsystems and Nanodevices

Electrical and Computer Engineering

©Anthony James Oko

Spring 2013

Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms. The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission

Abstract

The performance of highly anisotropic, nanostructured, thin-film ultrathin-layer chromatography (UTLC) media with porosity and architecture engineered using the glancing-angle deposition (GLAD) process is investigated. Anisotropic structures resembling nanoblades are fabricated on the plates, producing channel-like features that partially decouple analyte migration from development direction, and offering new separation behaviours. These GLAD UTLC plates provide channel features that reduce transverse spot broadening while providing the wide pores required for rapid migration and high separation performance with plate numbers approaching 800. The rapid separations typical of these miniaturized plates call for new alternative detection techniques. I have designed, implemented and characterized a measurement system which records UTLC separations in full colour with 32 μm spatial resolution and 33 ms temporal resolution. My code analyzes multiple tracks per plate, filters analyte spots by colour, and automatically generates time-resolved figures of merit. Both absorption and transmission detection modes are examined, each of which offers their own advantages.

Contents

1	INTRODUCTION	1
1.1	PROJECT DESCRIPTION	3
2	BACKGROUND	5
2.1	THIN LAYER CHROMATOGRAPHY.....	5
2.2	ULTRATHIN LAYER CHROMATOGRAPHY	6
2.3	THEORY OF OPERATION	7
2.3.1	Figures of Merit.....	8
2.4	GLANCING ANGLE DEPOSITION	11
2.5	GLAD UTLC.....	15
2.6	PLANAR CHROMATOGRAPHY DETECTION SCHEMES	17
3	ANISOTROPIC GLAD UTLC.....	20
3.1	INTRODUCTION	20
3.1.1	Chapter Overview	21
3.2	GLAD UTLC PLATE FABRICATION.....	21
3.2.1	GLAD System Parameters	21
3.2.2	Nanostructured Thin Film Characteristics	22
3.3	ANALYTE SPOTTING	23
3.4	DEVELOPMENT OF GLAD UTLC PLATE.....	23
3.5	ANISOTROPY MEASUREMENTS	24
3.6	CHROMATOGRAMS.....	25
3.7	PLATE NANOSTRUCTURE.....	26
3.7.1	Deposition Angle Effects	26
3.7.2	Nanoblade Disorder	28
3.8	SPOT DIFFUSION AND PLATE DEVELOPMENT.....	28
3.8.1	Spot Analysis	28

3.8.2	Elution Behavior	29
3.9	DYE SEPARATION ASSESSMENT	31
3.9.1	Effects of Varying Deposition Angle	31
3.9.2	Effects of Modifying Channel Orientation	33
3.10	QUANTIFICATION OF SEPARATION PERFORMANCE.....	33
3.10.1	Migration Distance.....	33
3.10.2	Plate Numbers	34
3.10.3	Retention Factors.....	34
3.11	DIFFUSIVITY RATIO.....	36
3.12	POTENTIAL OPPORTUNITIES.....	36
3.12.1	Limits of Detection.....	37
3.13	CONCLUSION	38
4	TIME-RESOLVED UTLC	40
4.1	INTRODUCTION	40
4.1.1	GLAD UTLC	40
4.1.2	Multiple Images of Single Plate Development.....	41
4.1.3	Quantitative Analysis	41
4.1.4	Chapter Overview	42
4.2	DETECTION METHOD	42
4.3	PREPARATION AND SPOTTING TECHNIQUE.....	44
4.4	UTLC PLATE DEVELOPMENT	44
4.4.1	Camera Settings	45
4.4.2	Frame Extraction.....	45
4.5	IDENTIFICATION AND ANALYSIS OF DYE COMPONENTS.....	46
4.6	VALIDATION OF HD CAMERA COLOR REPRODUCTION	47
4.6.1	Color Standard Evaluation	47
4.6.2	System Isolation from External Lighting	47
4.6.3	Compression Algorithms.....	48
4.7	HUE FILTERS	49

4.7.1	Post Processing	51
4.8	GENERATION OF TIME RESOLVED CHROMATOGRAMS	51
4.8.1	Separation Track Cropping.....	51
4.8.2	Chromatogram Production	52
4.8.3	Background Correction	53
4.8.4	Peak Curve Fitting	53
4.8.5	Color Mixing	55
4.8.6	Time Dependent Analysis	57
4.9	FIGURES OF MERIT	58
4.9.1	Mobile Flow Constant	58
4.9.2	Time Resolved Plate Numbers	59
4.9.3	Time Resolved Separation Numbers.....	60
4.9.4	Time Resolved Resolution	60
4.10	PROSPECTIVE SYSTEM APPLICATIONS OF TR-UTLC	62
4.11	CONCLUSION	63
5	TRANSMISSION MODE VIDEO UTLC	64
5.1	INTRODUCTION	64
5.2	EXPERIMENTAL SETUP	65
5.2.1	Development Chamber Design	65
5.2.2	LED Source Control	66
5.3	CIRCUIT DESIGN.....	69
5.3.1	LED Drivers	70
5.4	MEASUREMENT PRINCIPLE.....	70
5.4.1	Food Dye Analysis	71
5.4.2	LED Selection.....	71
5.5	RESULTS AND DISCUSSIONS: EXPERIMENT ONE	73
5.5.1	Introduction	73
5.5.2	Chromatograms	74
5.5.3	Analyte Response to LEDs.....	74

5.5.4	Spotting Deformities	75
5.6	RESULTS AND DISCUSSIONS: EXPERIMENT TWO	77
5.6.1	Introduction	77
5.6.2	Background Correction	77
5.6.3	Detection Capabilities	78
5.7	CONCLUSION AND FUTURE WORK	79
5.7.1	Source Synchronization	79
5.7.2	Enhanced Spectral Sensitivity	80
5.7.3	Limits of Detection	81
6	CONCLUSION	82
6.1	SUMMARY OF THESIS RESULTS	82
6.2	ONGOING EFFORTS	83
6.2.1	GLAD UTLC	83
6.2.2	Detection and Analysis	84
APPENDIX A	ANALYSIS CODE	92
A.1	MATLAB SCRIPTS	92
A.1.1	Define UTLC Analysis Parameters	92
A.1.1.1	checkHueRangev2.m	93
A.1.1.2	createRecipeUTLCv3	96
A.1.1.3	getChi4.m	114
A.1.1.4	migrationfit3.m	116
A.1.2	Colour-Filtered Densitogram Extraction	117
A.1.2.1	mainUTLCv8.m	117
A.1.2.2	filteredDensitogram.m	123
A.1.2.3	gaussFitUTLC.m	124
A.1.3	Figure of Merit Calculations	126
A.1.3.1	fomUTLC.m	127
A.2	ARDUINO MICROCONTROLLER CODE	136

List of Tables

Table 2-1: A comparison between a Merck UTLC plate and its predecessors, TLC and HPTLC. Data taken from Poole <i>et al</i> ¹⁷	7
--	---

Table 2-2: Advantages and disadvantages of several devices for use as detectors in planar chromatography.	18
--	----

Table 3-1: Morphological properties of SBD GLAD film posts. Channel width and the standard deviation of the nanoblade orientation (major axis of the SBD posts) increases with deposition angle.	27
---	----

Table 5-1: Characteristics of four different colored food dyes	71
--	----

Table 5-2: Characteristics of three different LEDs used as light sources in the transparent mode video UTLC system	72
--	----

List of Figures

Figure 2-1: Schematic of a separation on a TLC plate. 8

Figure 2-2: Schematic of the GLAD system setup. The source material's vapour flux strikes the substrate at an angle, α , relative to the substrate normal. The substrate can also be rotated during the deposition according to ϕ , producing films of varying morphologies. (Reproduced from Steele *et al.*²³ with permission from Elsevier). 12

Figure 2-3: A SiO₂ nanostructured film deposited by GLAD at a deposition angle of 87° and constant azimuthal angle. The scale bar is 2 μ m long. (Reproduced from Vick *et al.*³³ with permission from Cambridge University Press). 13

Figure 2-4: Chiral Alq₃ GLAD film with 328 nm pitch deposited at $\alpha=85^\circ$ and using continuous substrate rotation. The inset shows a top view of the film. (Reproduced from Hruddy *et al.*³⁴ with permission from the American Institute of Physics). 14

Figure 2-5: An oblique view SEM of a serial bi-deposition GLAD UTLC plate. (Reproduced from Jim *et al.*³⁷ with permission from Elsevier). 16

Figure 3-1: Horizontal development chamber used to develop anisotropic GLAD UTLC plate. 24

Figure 3-2: (a) Schematic showing analyte separation and angle definitions. Track angle (θ_T) and channel angle (θ_C) are reported relative to development direction. (b) Image processing masks are created to isolate a separation track. One of the masks is partially transparent for clarity. (c) A representation of the cropped image with all unmasked pixel rows left-aligned. (d) Chromatogram displaying the average darkness (see Section 2.5) along each pixel row. Developed plate images enhanced for presentation..... 26

Figure 3-3: High (top row) and low (bottom row) magnification scanning electron micrographs of 5 μ m thick SiO₂ SBD films. Films deposited at higher deposition angle (α) are more porous. Arrows indicate the along channel direction (θ_C)..... 27

Figure 3-4: (a) Applied spots broaden more in the along channel direction than in the across channel direction forming ellipsoids. (b) Fully developed plate with a small track angle deviation ($\Delta\theta$). Due to the films channel-like structure the migration front is curved. (c) Spot diffusion increases for larger α (greater porosity) and the spots' aspect ratios remain approximately constant. Plate images enhanced for presentation..... 30

Figure 3-5: Chromatograms for varying deposition angles in the along-channel direction ($\theta_C = 0^\circ$). Distances are referred to the initial spot position. The Dimethyl Yellow (DY) peak is clearly distinguishable for each deposition angle, while the Ariabel Red (AR) peak is only resolved at higher deposition angles. Developed plate images enhanced for presentation..... 31

Figure 3-6: Chromatograms from an $\alpha = 87^\circ$ SBD film at varying channel angles. The initial spot is located at $hRF = 0$. Peak broadening is observed for increasing channel angles along with a slight shift in hRF for both the Ariabel Red (AR) and Dimethyl Yellow (DY) peaks. The chromatograms are for each plate's center track. Developed plate images enhanced and peaks offset for presentation..... 32

Figure 3-7: (a) Migration front distances for 4 different deposition angles. (b) The large effect of deposition angle on plate number is observed. The largest plate numbers obtained were for $\alpha = 87^\circ$ with a maximum of just under 800. Open purple triangles located at channel angles of 0° and 45° correspond to the results obtained previously [17]. The Dimethyl Yellow (c) and Ariabel Red (d) retention factors were mostly independent of angle for a given film deposition angle. However, a noticeable change (up to 2x) in the retention factors can be seen for a 4.5° swing in α 35

Figure 3-8: Separation track deviation angle ($\Delta\theta$) as a function of channel angle (θ_C) for films grown at different deposition angles (α). $\Delta\theta$ is independent of deposition angle α . The solid line is a best fit of Eqn. (2) to the entire data set. Inset shows good agreement between model and experimental data at very high channel angles..... 38

Figure 4-1: Camera setup and apparatus used to take high-definition video of UTLC plate during development. A black cloth is draped around the system to control lighting during the development. The camera is isolated from the development chamber in order to protect the camera lens from any potentially corrosive vapours. The arrow indicates the plate development direction..... 43

Figure 4-2: The hue angle reported by our system against a color standard (Munsell ColorChecker Chart). (a) A linear fit for the uncontrolled lighting has a slope of 1.08 ± 0.04 , and intercept of -6 ± 8 . (b) If lighting conditions are controlled, the linear fit has a slope of 0.99 ± 0.02 , and intercept of 2 ± 4 indicating excellent agreement with the color standard..... 48

Figure 4-3: Cropped areas of each analyte from a particular development track (a) are used to generate histograms of sum of saturation signals for given hue (b). Analyte hue range selection enables calculation and analysis of time-resolved UTLC chromatograms. 50

Figure 4-4: A single frame from a separation video (a), showing the test dye mixture separated in six tracks. A three dimensional view (b) of saturation for each pixel in the frame. The black box in (a) and (b) outline the same track. No color (hue) filtering was performed when calculating these signals..... 52

Figure 4-5: (a) A single frame 250 seconds into development is shown with coloured dyes exhibiting different degrees of separation. (b) Chromatograms with (coloured lines) and without (black line) hue filtering. (c) Gaussian fits to the hue filtered chromatogram peaks. Hue filters allow deconvolution of partially separated peaks. The separation track image has been digitally enhanced for clarity. 55

Figure 4-6: Chromatograms from the fitted Gaussian peaks for each analyte at various times throughout the development. Peaks become increasingly well resolved as the development continues. Track images have been digitally enhanced for clarity..... 56

Figure 4-7: Time-dependent figures of merit including retention factor (a), plate number (b), separation number (c), and resolution (d). Resolutions are calculated between adjacent peaks in all cases except for that between DY (eluted first) and SB (eluted third). The signal for the OR (eluted second) was too weak for this analysis. A 2 second (20 frames) moving average is applied to all curves. 58

Figure 4-8: (a) Time-dependent migration distances and (b) peak standard deviations of each analyte are shown. A 2 second (20 frames) moving average was applied to all curves. 61

Figure 5-1: Development chamber used for transparent mode video ultrathin layer chromatography. 65

Figure 5-2: Customized aluminum LED holder consisting of 19 holes. The middle hole is left empty, while the remaining 18 holes are used to evenly distribute 3 different colored LEDs. 66

Figure5-3: Setup schematic of transparent mode TR-UTLC detection system.... 67

Figure 5-4: Schematic of the timing diagram used in the transmission mode video UTLC system. Over each complete period, it is ensured that at least one camera frame fully captures each LED color. 68

Figure 5-5: A schematic of LED drivers controlled by means of an Arduino microcontroller and solid state relays. 69

Figure 5-6: Transmission spectrum for four different colored food dyes. The peak wavelengths (blue – 470 nm, green – 525 nm, red – 660 nm) and spectral line half-widths of each LED are shown by the vertical lines and shaded color regions, respectively. 73

Figure 5-7: Lightness chromatograms of applied food dyes (from left to right: 5 mg/mL New Coccine, 8 mg/mL Indigo, 46 mg/mL Green B, 100 mg/mL Chro fb, 7 mg/mL Sunset Yellow) under various LED colors (from top to bottom: red, green, and blue)..... 76

Figure 5-8: Lightness chromatograms of a single frame from a Merck UTLC plate separation of test dye mixture III under white (top left), red (top right), green (bottom left), and blue (bottom right) LED illumination..... 78

.

List of Symbols

TLC	Thin Layer Chromatography	
HPTLC	High Performance Thin Layer Chromatography	
UTLC	Ultrathin Layer Chromatography	
GLAD	Glancing Angle Deposition	
α	Deposition Angle	deg
φ	Azimuthal Angle	deg
$\Delta\theta$	Track Angle Deviation	deg
θ_C	Channel Angle	deg
θ_T	Track Angle	deg
hRf	Retention Factor	%
N	Plate Number	
SN	Separation Number	
RS	Resolution	
Z	Migration Distance	Varies
W	Full Width Half Maximum	Varies
R	Diffusivity Ratio	
χ	Mobile Flow Constant	
σ	Peak Standard Deviation	
CCD	Charge Coupled Device	
LOD	Limit of Detection	ng
CMOS	Complementary Metal Oxide Semiconductor	

PVC	Polyvinyl Chloride	
FOM	Figures of Merit	Varies
LED	Light Emitting Diode	
SBD	Serial Bi-Deposition	
OC	Office Chromatography	
<i>DY</i>	Dimethyl Yellow	
<i>OR</i>	Oracet Red	
<i>SB</i>	Sudan Blue II	
<i>AR</i>	Ariabel Red	
<i>IP</i>	Indophenol	
<i>OV</i>	Oracet Violet	
RGB	Red, Green, Blue	
HSV	Hue, Saturation, Value	
HD	High Definition	
UV	Ultraviolet	
DC	Direct Current	
SSR	Solid State Relay	

1 Introduction

Thin layer chromatography (TLC) is ideal for low budget separation and quantification of compound inorganic and organic mixtures¹. In order to successfully separate the individual components of a mixture, several standard steps are carried out. First, the TLC plate is spotted with a compound mixture. Manual spotting is typically done inexpensively with a capillary tube. However, automated spotting is often used if speed and precision are warranted. Next, the TLC plate is placed in a development chamber so as to initiate the separation (development) process. The development chamber is pre-filled with a solvent (mobile phase) that, given enough time, will fully saturate a closed (covered) chamber. The mobile phase migrates through the stationary phase (porous immobile film on the surface of the TLC plate) through capillary action, and in the process begins to separate the mixture. The final step is to dry the plate.

Once a TLC plate has been completely developed and the individual components comprising the initial sample have separated, analysis can take place. Analysis can be achieved in a variety of ways, with ultraviolet (UV) visualization and mass spectrometry (MS) detection being two of the more popular techniques. Many analytes (components to be separated) will naturally fluoresce under UV radiation. In cases where the analytes do not naturally fluoresce, a fluorescent indicator can be applied to the TLC plate following the drying process. MS, also done post-development, is used to identify the analytes based on their mass-to-charge ratio. This detection flexibility is one of the main benefits of TLC. TLC

also allows for a wide variety of mobile phases, 2-D separations, and multiple separations on a single plate while requiring little in terms of laboratory equipment.

With all of the advantages of TLC, there remain a few drawbacks to overcome in order to for it to remain competitive with other separation techniques in the future, such as high performance liquid chromatography (HPLC). The limited efficiency, reproducibility, and free gas phase are some of the major obstacles inherent to TLC².

To aid in this effort, researchers have examined many exciting alternatives to traditional TLC plates. One of these alternatives, ultrathin layer chromatography (UTLC), is the main focus of this thesis. The characteristics of anisotropic glancing angle deposition (GLAD) UTLC plates are analyzed here, which provides further insight into the separation physics as well as helping to optimize plate performance.

GLAD UTLC plates have shown to provide excellent separation performance, employed with a variety of different film structures. These film's nanostructures allow for engineered porosity, unlike traditional TLC plates where the porosity is difficult to control. Analysis on varying film types, including vertical post, chevron, and serial bi-deposition (SBD) films have been performed recently^{3,4}. Post-separation techniques have also been implemented on GLAD UTLC plates using reactive ion etching, which has the effect of altering the plate's porosity. These plates can be divided into different areas (zones) which are etched for various time durations. This creates zones of tunable porosity, each

of which exhibit unique elution behavior. As the mobile phase migrates from a high to low porous zone, a concentrating effect is observed. This allows users to detect lower concentration sample mixtures⁵. However, with the adoption of these new separation media, reliable detection has become problematic, as the UTLC plates become more sensitive in the development process. The sensitivity is a result of the shorter migration distances evident with the use of GLAD UTLC plates, which is further exaggerated with the use of concentrating zones. It becomes extremely difficult to dry the UTLC plate without affecting the analyte's final separated position.

In this thesis, I have designed and implemented an automated detection scheme for these miniaturized plates to deliver a large quantity of data from a single development while reducing the amount of human intervention, and thus improving the reproducibility of the GLAD UTLC plates.

1.1 Project Description

This thesis is concerned with both the characterization and technological advances in the area of UTLC separations. The benefits of UTLC are exploited, by first developing an understanding of the effect of nanostructured morphology on UTLC separations, in order to design customized detection systems. An outline of the thesis is shown below:

Chapter 2 discusses the necessary background for the thesis. The history and development of TLC and the emergence of GLAD UTLC are detailed.

Chapter 3 deals with the study of anisotropic GLAD UTLC plates. The advantages of the GLAD process for use in UTLC are discussed. The extent of separation anisotropy and UTLC plate performance is shown to be related to specific GLAD parameters. The ability of the GLAD process to dramatically alter the separation media, allowing for much quicker and sensitive separations, further illustrates the need for an improved detection method if UTLC is to be used to its full potential.

Chapter 4 focusses on improving one of the weaknesses with the experimental procedures traditionally used in developing UTLC plates, whereby critical information can be lost due to human handling errors and destructive detection techniques. An innovative approach is undertaken, using a high-definition video camera, which extracts significantly more data during the separation interval, while improving the overall system reliability. Several MATLAB scripts are written that provide automated analysis of the crucial pieces of data extracted from a UTLC plate development.

Chapter 5 outlines a proof-of-concept experiment which attempts to improve upon the *in situ* video detection system used in Chapter 4. A transmission mode video UTLC setup, implementing LED source control, is designed and shown to have the potential to increase the detection sensitivity at early stages in the development process.

Chapter 6 provides a summary of the thesis and expresses possibilities for ongoing efforts in the field of UTLC.

2 Background

2.1 Thin Layer Chromatography

Planar chromatography has been used for over a century to effectively separate mixtures⁶. In planar chromatography, sometimes referred to as open-bed chromatography⁷, the stationary phase exists as a plane surface. It offers a simple, low cost alternative to other separation methods such as column chromatography, first developed by Tswett in 1903⁸.

Thin layer chromatography (TLC), a type of planar chromatography, operates on the theory that the analytes to be separated will have varying affinities between the stationary phase and the mobile phase. Adjustments to both of these phases enable users to establish the separation characteristics they desire. Since the introduction of TLC, researches have continued to implement higher quality planar separation techniques, eventually leading the advent of high performance TLC (HPTLC). HPTLC offers quicker, more efficient separations than TLC is capable.

Izmailov and Shraiber are widely recognized for their pioneering work on thin layer chromatography, and demonstrated separations of plant pigments in 1938⁹. Since then, TLC has since been used in a wide array of applications including identification of toxic substances in biological fluids, verification of pharmaceutical product claims, and even in the monitoring of water supplies to ensure safe drinking water¹⁰. TLC is effective for low-cost analysis of substances,

and can be especially favorable for those substances that are difficult to detect. Post-chromatographic derivatization treatments, easily achievable due to the open nature of the plates, allow for enhanced detection capabilities. In the 1970's HPTLC became popularized, marking a significant improvement in plate performance and efficiency. A smaller particle size is used in the HPTLC stationary phase, and thus greater porosity is attained, resulting in better separation characteristics. HPTLC provides quantification capabilities on par with other more costly chromatographic systems, such as gas chromatography and high performance liquid chromatography¹¹.

The largest manufacturer of TLC plates today is Merck Millipore International with the most commonly used sorbent material being silica gel 60, with a mean particle size of approximately 10 μm ¹². Several industrial pieces of equipment are available from a variety of manufacturers for the spotting, development, and detection of TLC plates. CAMAG remains one of the most popular manufacturers of such equipment. They offer products such as capillary dispensers for small volume spotting applications, vertical and horizontal development chambers, derivatization as well as chromatogram evaluation instruments¹³.

2.2 Ultrathin Layer Chromatography

Ultrathin layer chromatography (UTLC), first carried out by Hauck *et al*¹⁴ in 2001, is a relatively new separation technology that offers many advantages over both TLC and HPTLC. The substantially thinner stationary phase in UTLC

(< 10 μm) allows for lower reagent volumes, faster developments over shorter distances, and higher density parallel separations^{15,16}. UTLC plates are also much smaller than regular TLC and HPTLC plates, and thus require more complex analysis tools and techniques. Table 2-1 provides a comparison between these three planar chromatography modes.

Table 2-1: A comparison between a Merck UTLC plate and its predecessors, TLC and HPTLC. Data taken from Poole *et al*¹⁷.

<i>Mode</i>	<i>TLC</i>	<i>HPTLC</i>	<i>UTLC*</i>
Plate Size (cm)	20 x 20	10 x 10	6 x 3.6
Separation Time (min)	30 – 200	3 – 20	1 – 5
Development Distance (cm)	7 – 15	3 – 7	1 – 3
Layer Thickness (μm)	100 – 250	100 – 200	10
UV-Vis Detection Limit (ng)	1 – 5	0.1 – 0.5	0.5
Application Volume (μL)	1 – 5	0.1 – 0.5	0.01 – 0.1

2.3 Theory of Operation

Thin layer chromatography operates on the principle that each analyte in the mixture (initial spot) will have varying retention strengths to the stationary and mobile phases. The retention mechanism is mainly a function of contrasting polarity between analyte and each of the two phases¹⁸. As the mobile phase (solvent) migrates through the stationary phase, a process that is induced by

capillary action, analytes that are more strongly retained by the stationary phase will not migrate as far. This characteristic of each analyte, known as retardation or retention factor, must vary amongst analytes for them to separate sufficiently. The retention factor is defined as the ratio between analyte and mobile front migration distance from the initial spotting position.

2.3.1 Figures of Merit

There exist several figures of merit in TLC to effectively quantify the quality of separation performance. A schematic indicating the relevant terms involving a TLC separation is shown in Figure 2-1. It is these terms upon which the relevant figures of merit are derived.

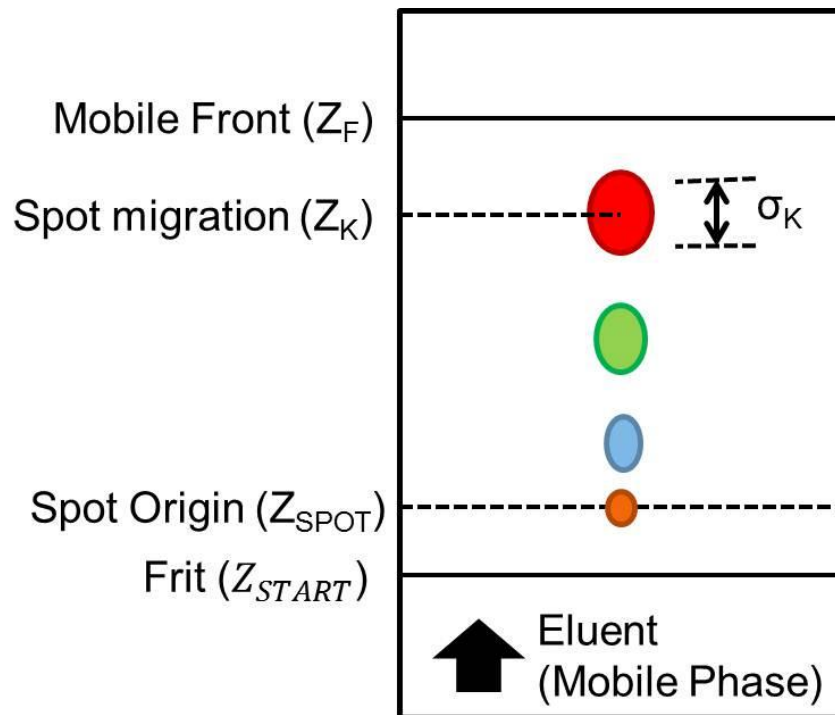


Figure 2-1: Schematic of a separation on a TLC plate.

The retention factor (hR_F) describes how far a particular analyte has migrated (in terms of percentage) with respect to the mobile front position, Z_F , and can be written as

$$hR_F = 100 \frac{Z_F - Z_{SPOT}}{Z_K - Z_{SPOT}} \quad (2-1)$$

Plate number (N), or number of theoretical plates, is often used to describe the efficiency of the separation. This is a concept that is derived from column chromatography. In column chromatography, as the analytes are carried through the column, the analytes are said to go through equilibrations over each ‘plate’ between the stationary and mobile phases. The more ‘plates’ in a column, the better able the column is to separate the compounds. Hence, the separation efficiency can either be improved by increasing the column length or by increasing the density of plates. The length of each plate, known as the plate height (H), is another commonly used term in chromatography. Both plate number and plate height can be used interchangeably as they are simply related by the column length as illustrated in Equations 2-2 and 2-3, respectively.

$$N = \frac{(Z_K - Z_{SPOT})^2}{\sigma^2} \quad (2-2)$$

$$H = \frac{Z_K - Z_{SPOT}}{N} \quad (2-3)$$

In TLC, the column length is analogous to the analyte migration distance and each plate can be thought of as a layer of particles.

Resolution (RS), a unitless quantity, is another figure of merit important to TLC. It describes the overlapping, or lack of, for two analytes. Resolution calculations are reliant upon a gaussian distribution for each separated spot. A

large resolution is ideal and indicates a high degree of separation. The resolution can be mathematically written as

$$RS = \frac{1}{2} \frac{Z_{K,2} - Z_{K,1}}{\sigma_{K,2} - \sigma_{K,1}} \quad (2-4)$$

The maximum number of analytes a TLC plate can theoretically separate is referred to as separation number (SN). The separation number is closely related to plate number and resolution, which must both be large in order for the separation number to be adequate. The separation number is defined as

$$SN = \frac{Z_F}{b_1 + b_0} - 1 \quad (2-5)$$

where b_1 and b_0 are the extrapolated full width half maximum at $hRf = 100$ and $hRf = 0$, respectively.

It is ideal to have as little spot broadening as possible during a separation. This condition ensures that the highest efficiency plate performance is achieved, resulting in small theoretical plate heights and large separation numbers. The van Deemter equation is commonly used in TLC. It relates plate height, H , to the mobile front velocity, u , and can be written as:

$$H = A + \frac{B}{u} + Cu \quad (2-6)$$

where A is the Eddy diffusion coefficient, B is the molecule diffusion coefficient, and C is the mass transfer coefficient.

Eddy and molecule diffusion as well as mass transfer must be optimized to minimize spot broadening¹⁹, according to the van Deemter equation. Eddy diffusion is caused by variances in pore size diameter in the stationary phase while molecule diffusion is a function of temperature, viscosity and particle size. Mass transfer is a measure of how quickly the solute transfers between phases²⁰.

2.4 Glancing Angle Deposition

Glancing angle deposition (GLAD) is a physical vapor deposition technique performed under high vacuum that utilizes substrate tilt and rotation to deposit highly porous thin films with adjustable columnar morphologies²¹. A schematic of the GLAD setup is shown in Figure 2-2.

GLAD was originated by Robbie *et al.* in 1995²². They demonstrated a system capable of depositing highly structured, porous thin films. Unlike previous methods in which films were deposited with a highly oblique incident flux from only one side, the symmetrical deposition process in GLAD allows for vertically oriented columns, while maintaining the high surface area to volume ratio that is desirable amongst many applications. The complex morphologies typical in GLAD are made possible because of the self-shadowing that occurs during the deposition process. GLAD also offers the flexibility of depositing a large range of materials by means of electron beam evaporation.

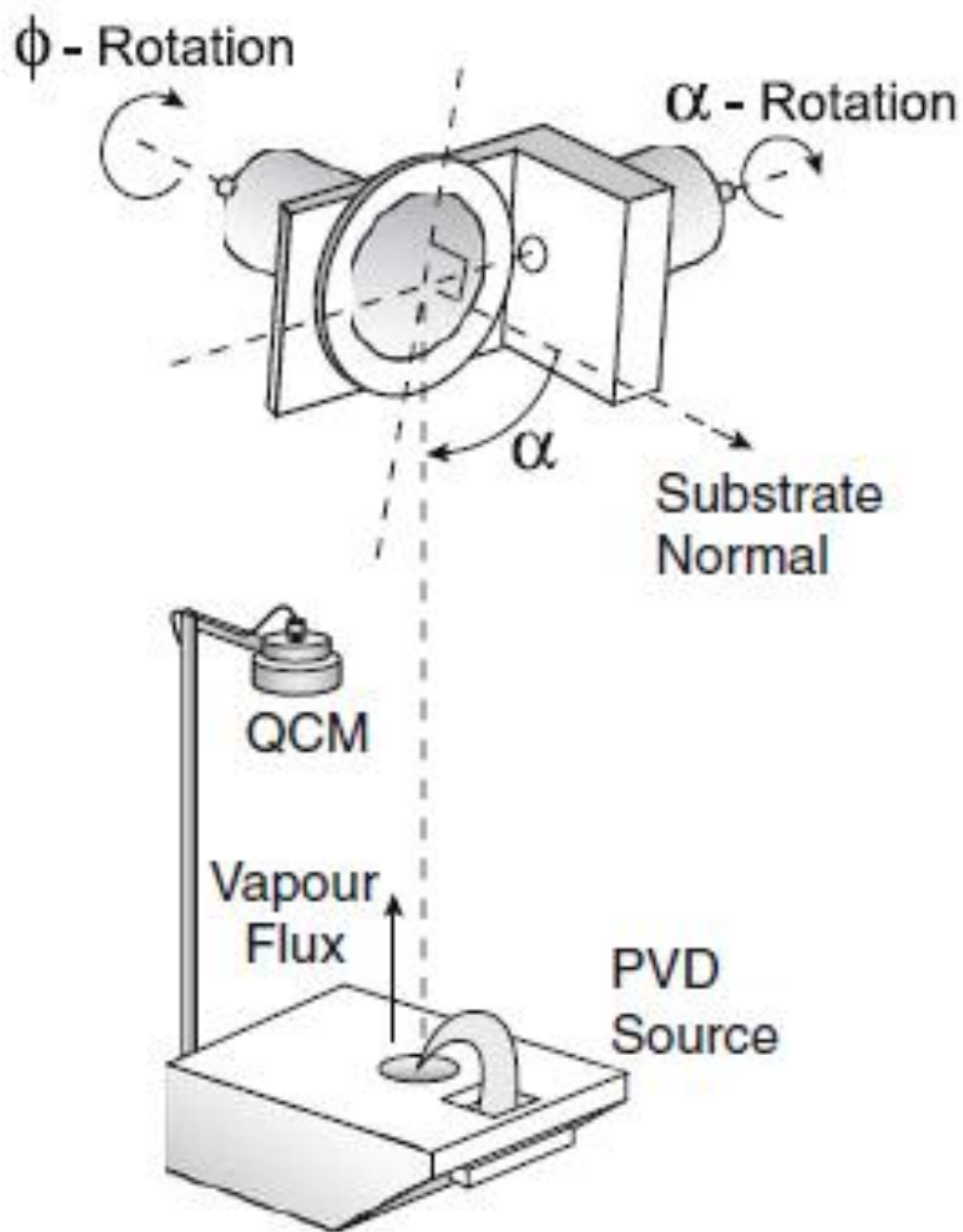


Figure 2-2: Schematic of the GLAD system setup. The source material's vapour flux strikes the substrate at an angle, α , relative to the substrate normal. The substrate can also be rotated during the deposition according to ϕ , producing films of varying morphologies. (Reproduced from Steele *et al.*²³ with permission from Elsevier).

Since this initial work by Robbie *et al.*, advances in GLAD research have increased the capacity of this technique in several different areas. The high surface area and exceptional control over structure of GLAD films has been exploited in applications involving organic photovoltaic devices^{24–26}, photonic crystals^{27–29}, sensors^{28,30,31}, and optical filters³², amongst many others.

Figure 2-3 shows a GLAD film deposited with a constant ϕ at $\alpha=87^\circ$. The smaller, thinner columns in Figure 2-3 are a result of the self-shadowing inherent in GLAD. Also evident, the deposited material tends to grow towards the angle of incident flux due to low diffusion and preferential adsorption to one side of the film.

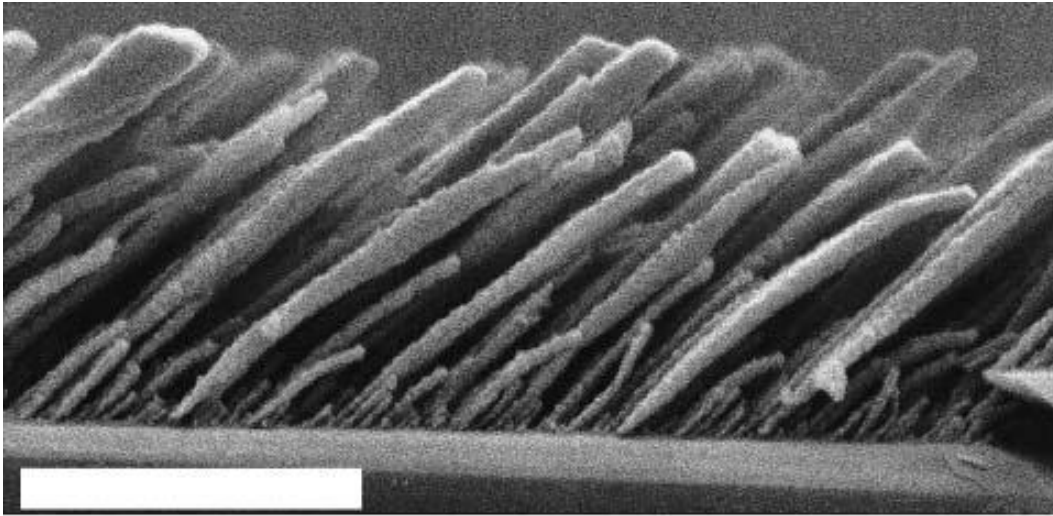


Figure 2-3: A SiO_2 nanostructured film deposited by GLAD at a deposition angle of 87° and constant azimuthal angle. The scale bar is $2\text{ }\mu\text{m}$ long. (Reproduced from Vick *et al.*³³ with permission from Cambridge University Press).

It is also possible to engineer intricate structures using the GLAD process, one of which is demonstrated in Figure 2-4. In this case, a helical Alq_3 GLAD film is fabricated for use as a one-dimensional polarization-dependent photonic

stop band³⁴. In this case, ϕ is rotated at a rate that is directly proportional to the deposition rate. The multivariable motor control (both α and ϕ) offered by GLAD provides the capability to produce many other fascinating structures, such as serial bi-deposition (SBD) films, which is discussed further in Chapter 3.

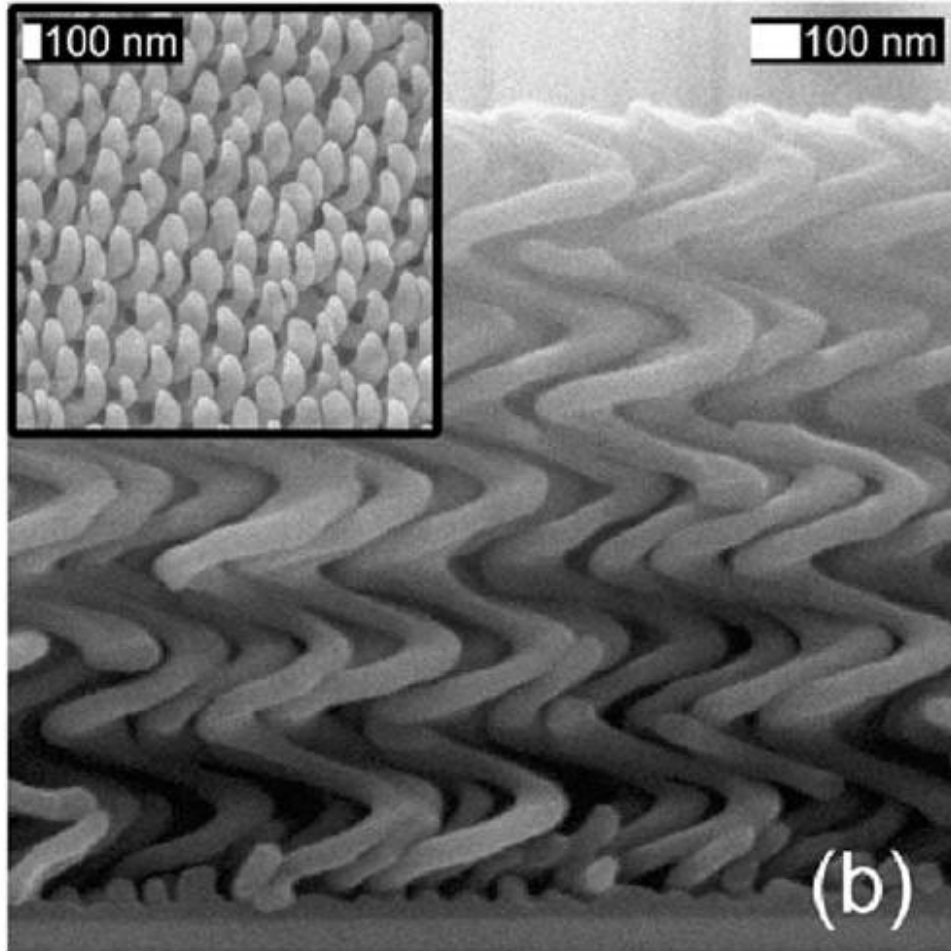


Figure 2-4: Chiral Alq₃ GLAD film with 328 nm pitch deposited at $\alpha=85^\circ$ and using continuous substrate rotation. The inset shows a top view of the film. (Reproduced from Hrudey *et al.*³⁴ with permission from the American Institute of Physics).

2.5 GLAD UTLC

GLAD, as it pertains to UTLC, offers excellent control over the stationary phase media morphology. The specific area and porosity of the stationary phase is manipulated by altering the deposition angle, α . The different film morphologies produced in GLAD are completed by adjusting the substrate rotation, ϕ . It is also possible to grow films consisting of several different materials utilizing the GLAD process³⁵, including SiO₂ and Al₂O₃, both of interest in UTLC media.

Of particular interest to this thesis, Bezuidenhout *et al.* first used GLAD to grow SiO₂ nanostructured UTLC media in 2008³⁶. In their qualitative analysis, Bezuidenhout *et al.* grew two film structures, six-sided spirals and zig-zag, at several different porosities and thicknesses. They showed that the separation quality, determined visually, could be greatly improved as α , the deposition angle, increased over 75°, leading to faster migration front velocities. Thicker films also were shown to increase separation performance.

In 2010 Jim *et al.* demonstrated the capability of anisotropic GLAD UTLC SiO₂ plates to develop in a different direction than the migration of the mobile phase, which required a new analytical approach³⁷. This research exhibited the powerful control over the UTLC separations made possible through GLAD. A relationship between the migration anisotropy and track deviation angle was also presented. Jim *et al.* were able to attain limits of detection of 10 ng and plate heights as low as 12 μm with their GLAD UTLC plates. An SEM image of one of their SBD GLAD UTLC plates is shown in Figure 2-5.

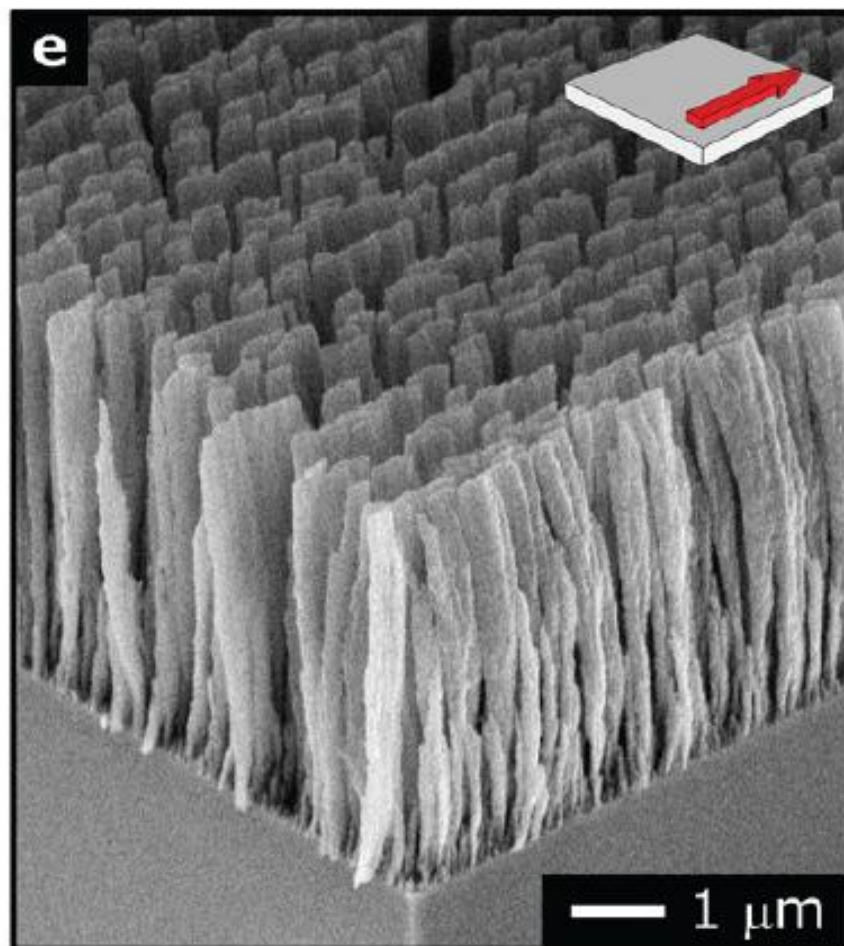


Figure 2-5: An oblique view SEM of a serial bi-deposition GLAD UTLC plate. (Reproduced from Jim *et al.*³⁷ with permission from Elsevier).

Further control of the elution behavior of GLAD UTLC plates was established in 2011 when Jim *et al.* utilized post-processing to modify GLAD films³⁸. Using reactive ion etching to alter the surface area and ultimately the retention behavior of their thin films, they fabricated concentration zones on the GLAD UTLC plates. These zones exhibited the potential to focus large volume, dilute mixtures resulting in theoretical plate number up to 1200.

Continued UTLC stationary phase engineering efforts are motivated by the enhanced separation performance possible with controlled microstructure. Electrospinning offers an interesting approach to produce nanofibrous reversed-phase polymer³⁹ and glassy carbon⁴⁰ UTLC stationary phases. Other methods include physical vapour deposition of macroporous thin films.

2.6 Planar Chromatography Detection Schemes

Many different detection schemes are currently used in planar chromatography. These include slit and video densitometry, flatbed scanners, digital cameras, and industrial charge coupled device (CCD) cameras^{41–45}. Each of these detection devices offers particular benefits over their competitors and also has their drawbacks, as illustrated in Table 2-2.

Table 2-2: Advantages and disadvantages of several devices for use as detectors in planar chromatography.

<i>Instrument</i>	<i>Advantages</i>	<i>Disadvantages</i>
Slit Densitometer	-Well researched -Simple setup	-Slow -Expensive
Video Densitometer	-Captures entire plate -Commercially available software	-Expensive
Flatbed Scanner	-High resolution images -Fast image capture -Inexpensive	-Increased human handling of plates
Digital Camera	-Inexpensive -Fast Image Capture	-Images are compressed -Complex setup
CCD Camera	-High resolution images -Fast image capture -Ultraviolet response	-Expensive -Complex setup

Once an image of the complete plate is gathered, with the exception of the slit densitometer, the image's data can then be analyzed. In many cases, commercially available software is used to extract the necessary information from the post-developed plate. This information is obtained from chromatograms produced by the software package and typically includes parameters such as plate number, plate height, and limits of detection. However, most of these software packages are designed for larger plate sizes, such as in TLC or HPTLC. With UTLC being a relatively new technology, commercial detection solutions have yet to become commonplace. Obtaining the appropriate figures of merit (FOM) from the extracted post-developed plate images can become a challenging

undertaking. Variations in lighting as well as plate defects must be considered and taken into account by executing plate background corrections^{46,47}. Due to the unpredictable separation behavior observed at times in planar chromatography, automatic detection of initial spots, migration fronts, and peak widths can be difficult. To aid in this task, efforts have recently been made to improve detection sensitivity or reliability by taking multiple images via digital or CCD cameras^{48,49}. This thesis research presents the implementation of a new detection technique using a digital high-definition video camera. Videos taken from a UTLC plate development are subsequently analyzed by use of custom-written MATLAB scripts to generate time-resolved figures of merit.

3 ^{*}Anisotropic GLAD UTLC

3.1 Introduction

GLAD is an established vacuum deposition technique for engineering porous columnar thin films in many useful materials and morphologies^{50–52}, and is a flexible platform for macroporous UTLC stationary phase study and design^{36,37,53}. The first study of normal-phase silica GLAD-UTLC thin film media investigated elution dependence on porosity and microstructure³⁶. Isotropic and anisotropic chromatographic media were comprised of hexagonal helices and chevrons, respectively. Rapid separations performed over short distances demonstrated that nanostructured GLAD-UTLC plates have a place within the new “Office Chromatography” (OC) concept⁵³. Sample mixtures applied with an inkjet printer were separated on GLAD thin film UTLC plates in less than 60 seconds over ~10 mm and subsequently imaged with a flatbed scanner. Office peripheral performance exceeded that of dedicated TLC equipment, providing sufficient spotting and imaging precision to take advantage of miniaturised UTLC plates. The most recent investigation quantified GLAD UTLC plate performance with isotropic and anisotropic macropores using lipophilic dye separations³⁷. Analyte migration and plate performance depended strongly on UTLC plate anisotropies, motivating a more thorough investigation.

^{*} This chapter has been modified from Oko *et al.*⁴⁵ with permission from Elsevier

3.1.1 Chapter Overview

In this chapter, we report detailed investigations and modeling of anisotropic blade-like thin film chromatography media. UTLC plates of varied porosity and channel feature orientation were fabricated using GLAD and characterized using lipophilic dye separations. Migration distances up to 22 mm and plate numbers approaching 800 were obtained. In addition, we present a theoretical model describing mobile phase flow in anisotropic media. The effects of film microstructure on UTLC figures of merit are discussed. Engineered GLAD-UTLC plates taking advantage of the unique elution behaviours studied here may improve performance over conventional planar chromatography methods.

3.2 GLAD UTLC Plate Fabrication

3.2.1 GLAD System Parameters

The GLAD technique has been described previously^{36,37,50–52}, so only a brief review is given here. In GLAD, a substrate located within a high-vacuum physical vapour deposition system is maneuvered to achieve several thin film morphologies^{50,51}. Manipulation of the film deposition angle (α) changes porosity⁵⁴ and surface area⁵⁵ while modulation of the azimuthal angle (φ) enables a variety of architectures including slanted posts, helices, vertical posts, chevrons, and anisotropic nanoblade structures. System base pressures were less than 100 μPa ; deposition pressures varied between 460 μPa – 660 μPa . This variation in

deposition pressure was due to the evaporation of the source material during the deposition, resulting in increased pressures. The deposition rate was held at approximately 8 \AA s^{-1} – 10 \AA s^{-1} , low enough to minimize φ -position errors.

3.2.2 Nanostructured Thin Film Characteristics

We deposited 5 μm thick (nominal) GLAD SiO_2 thin films on 1 inch square B270 glass substrates (Schott B270, S. I. Howard Glass, Worcester, Massachusetts, USA). Four deposition angles ($\alpha = 82.5^\circ, 84.0^\circ, 85.5^\circ$, and 87.0°) were chosen to produce films with different porosities; porosity increases with deposition angle^{50,51,54}. A serial bideposition (SBD) film structure with a 24 nm period was grown^{37,56}. The 24 nm period consists of four stages: 10 nm growth at $\varphi = 0^\circ$, 2 nm uniform growth as φ changes from 0° to 180° , another 10 nm growth at $\varphi = 180^\circ$, and finally 2 nm uniform growth as φ changes from 180° to 0° .

SBD structures are anisotropic in the substrate plane, resulting in well-defined channel-like features. Such channels may be oriented at an arbitrary angle with respect to the development direction (perpendicular to the top edge of the GLAD-UTLC plate). We define this angle as the “channel angle” (θ_c , shown in Figure 3-2a). Nominal channel angles from 0° to 65° inclusive and 90° were obtained by orienting the edge of each glass substrate at the desired angle relative to the evaporant source direction during deposition. All samples were thoroughly cleaned with a dilute solution of Citrinex (Alconox, White Plains, New York, USA) and rinsed with deionized water prior to deposition. Films were also

deposited on silicon wafers for scanning electron microscopy (SEM) characterization.

3.3 Analyte Spotting

Test Dye Mixture III (CAMAG, Muttenz, Switzerland) was diluted in toluene ($\geq 99.5\%$, Sigma-Aldrich Canada, Oakville, Ontario, Canada) to 50%. This diluted mixture contained 1.5 mg of Dimethyl Yellow, 0.25 mg of Oracet Red G, 1 mg of Sudan Blue II, 0.75 mg of Ariabel Red, 1 mg of Oracet Violet 2R, and 2 mg of Indophenol per mL of toluene (listed in descending hR_F). Spots were applied with a desktop dispensing robot (I&J 2200, I&J Fisnar, Wayne, New Jersey, USA). We manually dipped a 1/4 inch, 32 gauge blunt end stainless steel tip needle (I&J Fisnar) into a small volume ($\sim 500\ \mu\text{L}$) of diluted test dye mixture before securing it onto the robot arm. Five spots were applied 4 mm apart along a line 3 mm from the bottom edge of the UTLC plate. At each spot the needle was rested against the plate for 1 second, allowing the diluted dye to wick into the film through capillary forces. This procedure resulted in estimated spot volumes between 2 nL and 10 nL.

3.4 Development of GLAD UTLC Plate

We used a horizontal development chamber (Desaga H-Chamber, 50 mm x 50 mm, Sarstedt, Nümbrecht, Germany) similarly modified to that in ³⁷, and shown in Figure 3-1. The reservoir and conditioning trough were filled with the mobile phase (4:3 toluene:*n*-hexane; 95% *n*-hexane, Fisher Scientific, Ottawa, Ontario, Canada) at least one hour prior to development to saturate the vapour

phase. Spotted UTLC plates were loaded face-down into the chamber and aligned against the back glass spacer to control placement and angular orientation. Mobile phase in the reservoir continuously wicked up the porous glass frit and onto the face-down thin film stationary phase. Each plate was developed for 90 seconds and then removed from the chamber. An 1875 W hot air dryer was used to evaporate residual solvent by directing air opposite to the development direction at $\sim 45^\circ$ with respect to the plate normal.

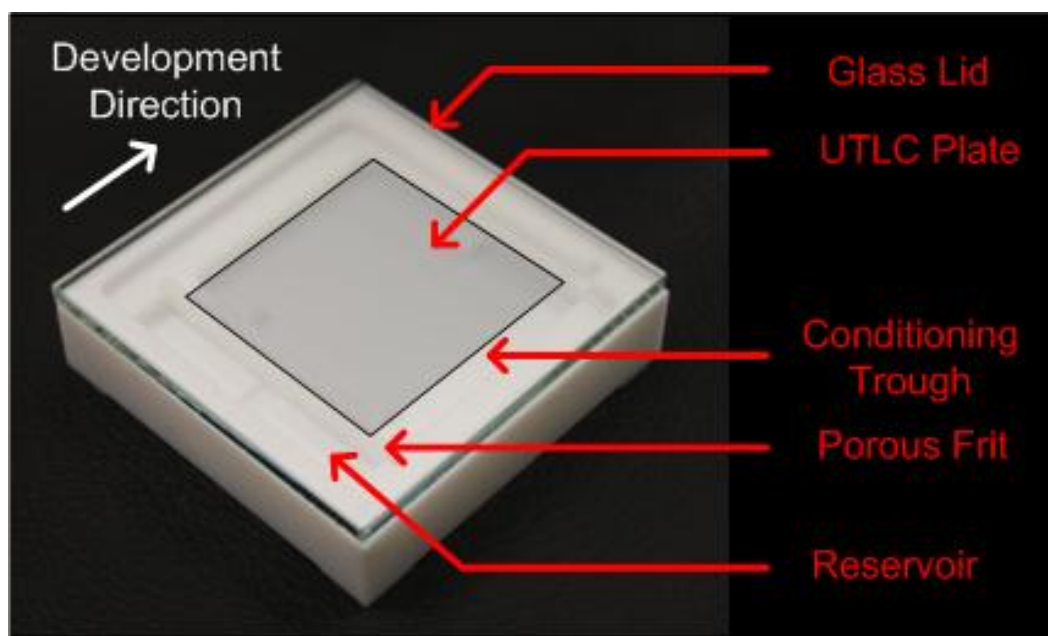


Figure 3-1: Horizontal development chamber used to develop anisotropic GLAD UTLC plate.

3.5 Anisotropy Measurements

Samples were scanned before and after development using a flatbed film scanner (CanoScan 5600F, Canon, Mississauga, Ontario, Canada) in transmission

mode. Photoshop (Photoshop CS3 Extended, version 10.0.1, Adobe Systems, San Jose, USA) was used to determine the channel and separation track angles from the high resolution (1200 dpi) scanned images. The channel orientation angle (θ_C) was measured from scans of undeveloped plates using an elliptical best fit to applied spots using ImageJ⁵⁷. We used the Dimethyl Yellow spot to obtain accurate separation track measurements as this analyte travelled furthest. The angles reported here are an average and standard deviation for each sample. Some tracks at channel angles greater than 60° were distorted by our hot air drying technique; these tracks were excluded.

3.6 Chromatograms

Photoshop was also used to isolate each separation track. To clearly locate each track, scanned images were enhanced using a similar procedure to that described in³⁷. Each track was masked to a constant pixel width along the development direction as seen in Figure 3-2b. Track widths were chosen to match the analyte that broadened most during development, with typical values of 0.85 mm (40 pixels). The unenhanced, isolated tracks were saved as JPEG images (it was not possible to obtain a lossless image from the flatbed scanner), as in Figure 3-2b, and imported into a custom script written for the MATLAB mathematical analysis software platform (MATLAB R2009a, version 7.8.0.347, MathWorks Inc., Natick, Massachusetts, USA) which calculates the average darkness value for each pixel row of the isolated track image³⁷. Chromatograms were produced by plotting average darkness against development direction. Output from this analysis is shown in Figure 3-2d.

3.7 Plate Nanostructure

3.7.1 Deposition Angle Effects

Top down SEM images of our UTLC films are shown in Figure 3-3 at different magnifications. The nanoblade SBD structure is clearly visible for all deposition angles. Image processing using ImageJ⁵⁷ was used to quantify some aspects of film morphology. Combined with a custom Python code⁵⁸, we were able to determine average edge-to-edge nanoblade spacing (channel width) and standard deviation of nanoblade orientation. This was determined via the top view images of the SEM images. Dr. Michael Taschuk was a significant help in obtaining these results, which are given in Table 3-1. The average channel width

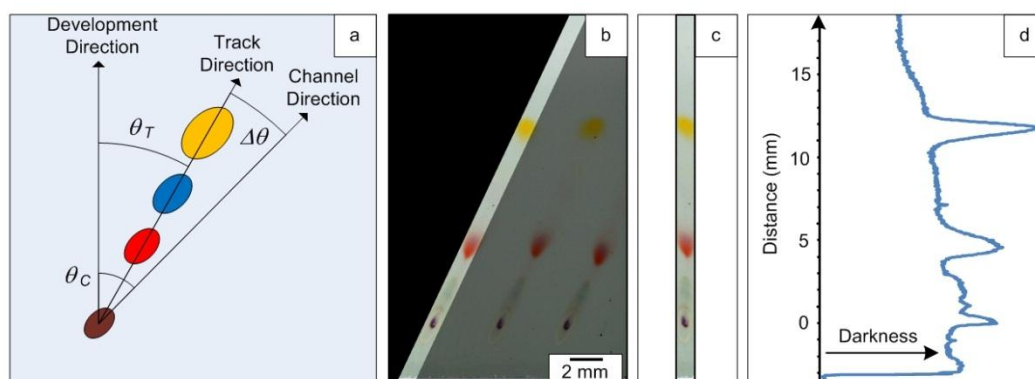


Figure 3-2: (a) Schematic showing analyte separation and angle definitions. Track angle (θ_T) and channel angle (θ_C) are reported relative to development direction. (b) Image processing masks are created to isolate a separation track. One of the masks is partially transparent for clarity. (c) A representation of the cropped image with all unmasked pixel rows left-aligned. (d) Chromatogram displaying the average darkness (see Section 2.5) along each pixel row. Developed plate images enhanced for presentation.

increases with deposition angle, reflecting the expected porosity dependence on deposition angle for GLAD films. However, spacing in the across-channel direction is poorly defined, and could not be quantified using the same analysis.

Table 3-1: Morphological properties of SBD GLAD film posts. Channel width and the standard deviation of the nanoblade orientation (major axis of the SBD posts) increases with deposition angle.

Deposition Angle (α) (degrees)	Average Channel Width (nm)	Nanoblade Orientation Standard Deviation (degrees)
82.5	400 ± 20	15
84.0	670 ± 20	19
85.5	690 ± 20	21
87.0	840 ± 20	26

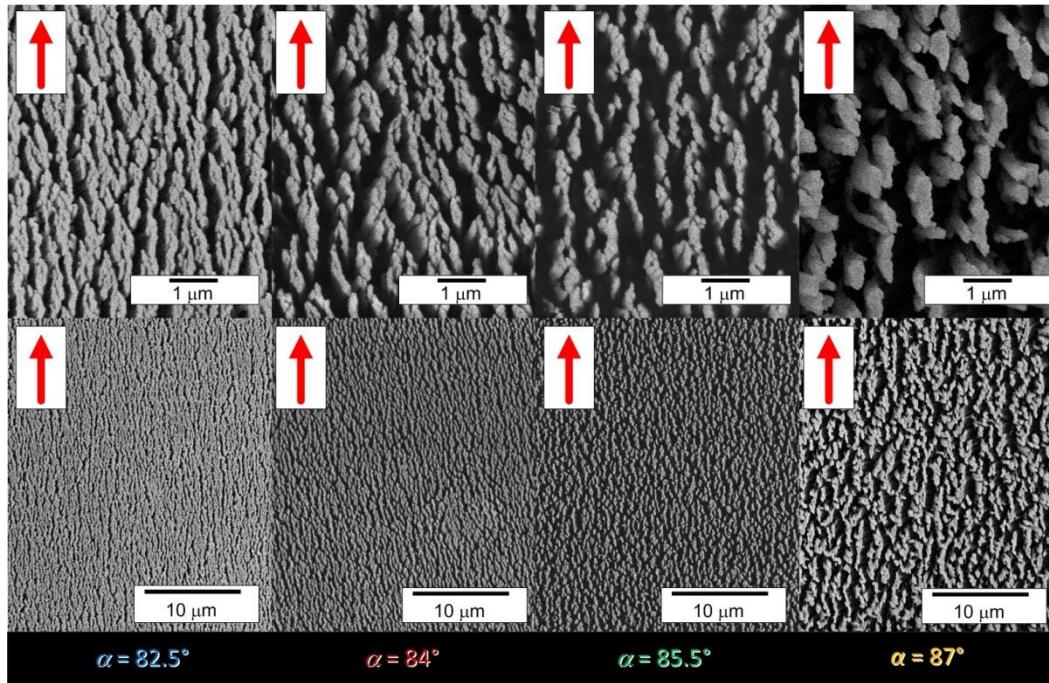


Figure 3-3: High (top row) and low (bottom row) magnification scanning electron micrographs of 5μm thick SiO₂ SBD films. Films deposited at higher deposition angle (α) are more porous. Arrows indicate the along channel direction (θ_c).

3.7.2 Nanoblade Disorder

Also interesting is the increase in disorder with deposition angle. This value is reported as the standard deviation of the SBD nanoblade's orientation; a higher value indicates greater disorder. The post's preferential orientation decreases with increasing deposition angle. Increased variability in the channel walls may act to impede fluid flow, competing with the increased channel width. Fluid flow simulations may shed some light on the relative contribution of these effects.

3.8 Spot Diffusion and Plate Development

3.8.1 Spot Analysis

Initial analyte spots were elliptical due to preferential diffusion in the along-channel direction, as seen in Figure 3-4. As expected, spot diffusion increases with increasing porosity (deposition angle) and is clearly visible in Figure 3-4c. However, the aspect ratio (major to minor axis length; ~ 2 to 3) of the initial spots remained approximately constant with deposition angle. This suggests that the film's across-channel and along-channel diffusivities are changing at approximately the same rate, and thus the macropore anisotropy does not depend on deposition angle.

3.8.2 Elution Behavior

During chromatographic separation, the analytes migrated at a track angle that deviated a few degrees ($\Delta\theta$) from the channel angle (Figure 3-4b). The separated analyte spots remained elliptical during development (Figure 3-2a). The curved migration front seen on dried developed plates (Figure 3-4b) is characteristic of differing channel and development directions and was consistent with that observed in ³⁷. The migration front is recorded in GLAD-UTLC plates through capillary-induced clumping of the film's nanostructure ⁵⁹⁻⁶¹. Although this front was visible in the GLAD thin films deposited at the higher angles ($\alpha = 85.5^\circ$ and $\alpha = 87.0^\circ$), it was difficult to accurately determine its location in those deposited at the lower angle ($\alpha = 82.5^\circ$ and $\alpha = 84^\circ$). This suggests that the effect of column clumping on the optical properties in wetted regions is suppressed in the denser films. When visible, the migration front distance was taken as the furthest distance the migration front travelled from the plate's bottom.

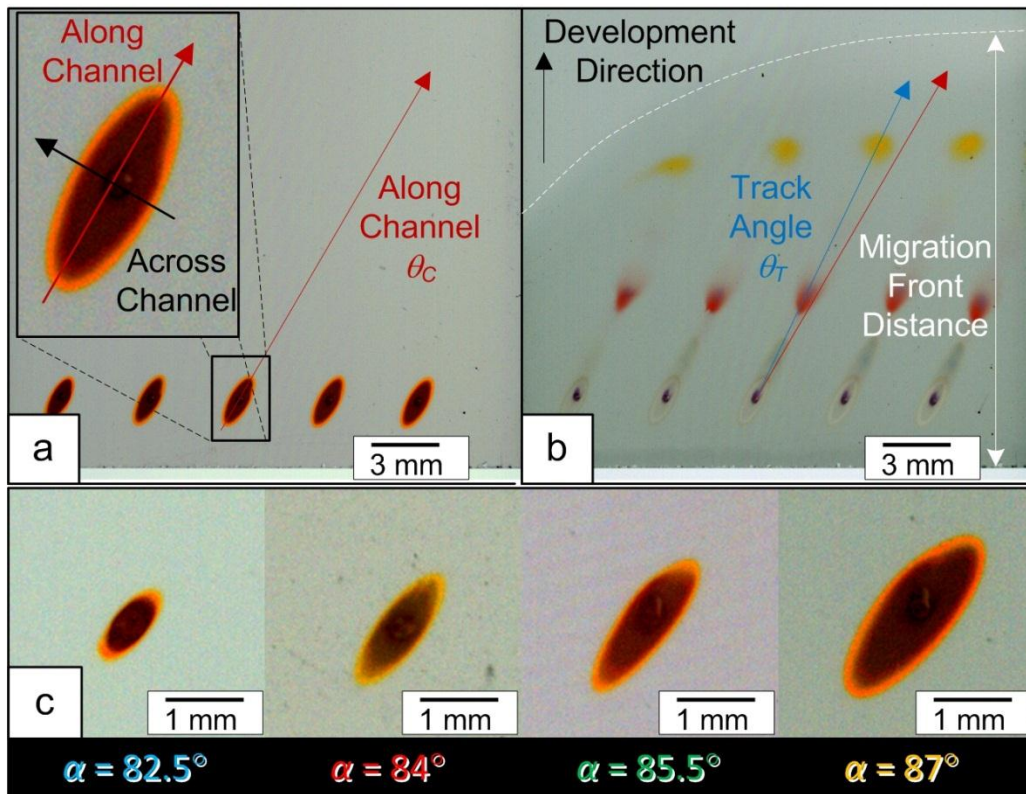


Figure 3-4: (a) Applied spots broaden more in the along-channel direction than in the across-channel direction forming ellipses. (b) Fully developed plate with a small track angle deviation ($\Delta\theta$). Due to the film's channel-like structure the migration front is curved. (c) Spot diffusion increases for larger α (greater porosity) and the spots' aspect ratios remain approximately constant. Plate images enhanced for presentation.

3.9 Dye Separation Assessment

3.9.1 Effects of Varying Deposition Angle

To evaluate our nanostructured UTLC plate's utility in the anisotropic mode, we performed trial separations using the dye solution. Figure 3-5 shows chromatograms and developed plates at all deposition angles. Porosity and

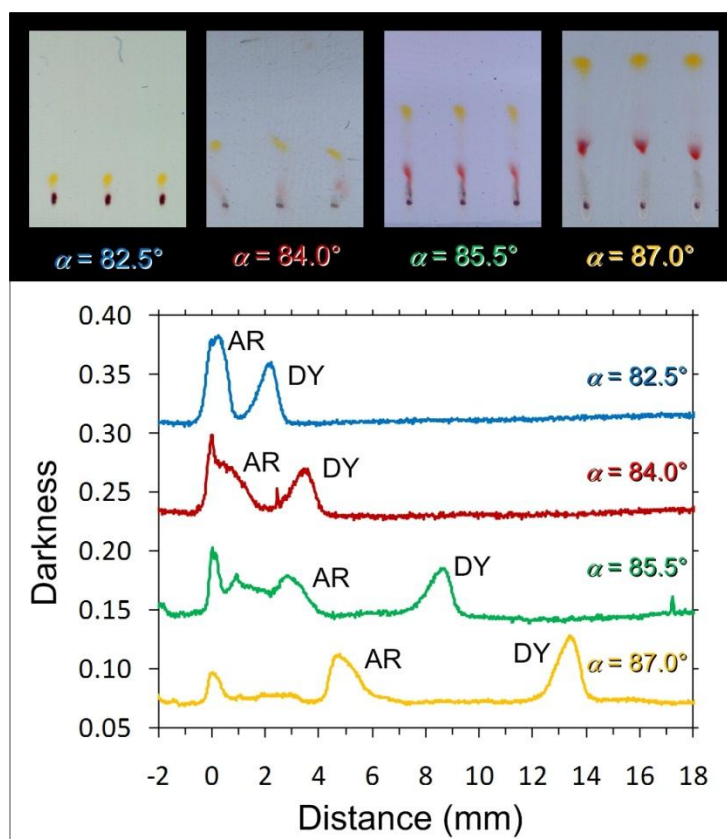


Figure 3-5: Chromatograms for varying deposition angles in the along-channel direction ($\theta_c = 0^\circ$). Distances are referred to the initial spot position. The Dimethyl Yellow (DY) peak is clearly distinguishable for each deposition angle, while the Ariabel Red (AR) peak is only resolved at higher deposition angles. Developed plate images enhanced for presentation.

channel widths increase with deposition angle, providing a more permeable stationary phase. This is reflected in the increased migration distances and improved separations observed at higher deposition angles.

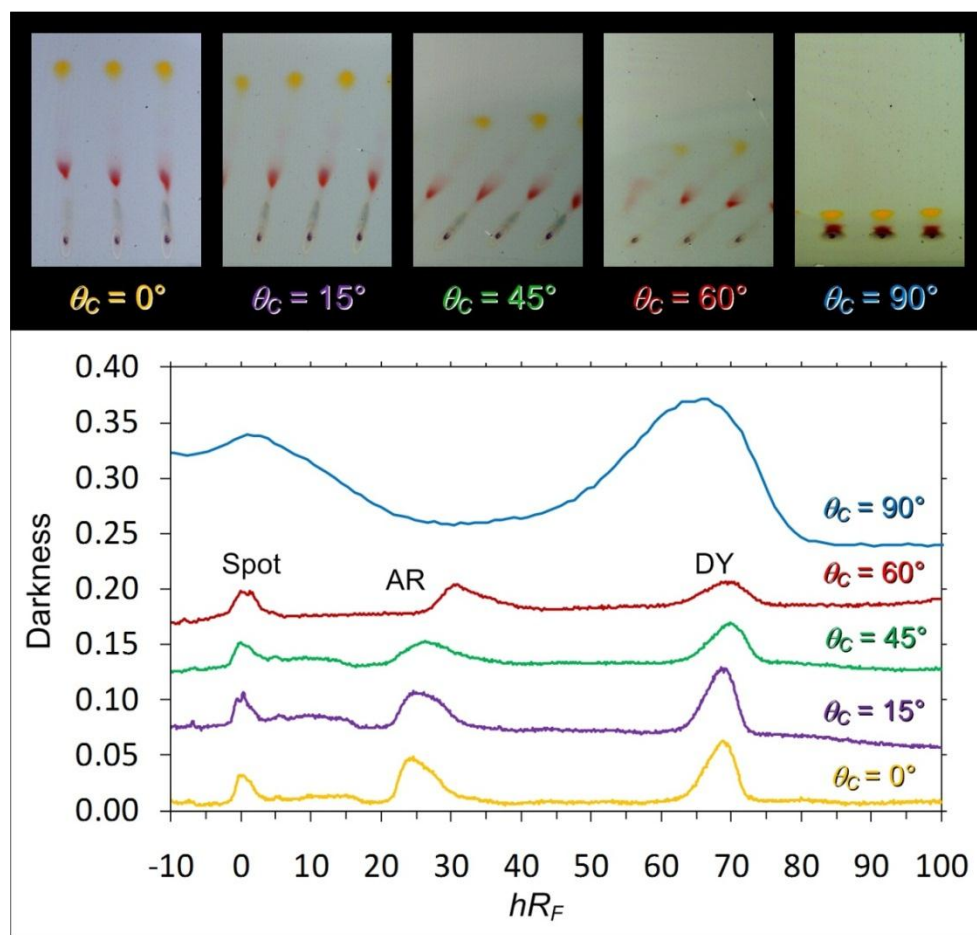


Figure 3-6: Chromatograms from an $\alpha = 87^\circ$ SBD film at varying channel angles. The initial spot is located at $hR_f = 0$. Peak broadening is observed for increasing channel angles along with a slight shift in hR_f for both the Ariabel Red (AR) and Dimethyl Yellow (DY) peaks. The chromatograms are for each plate's center track. Developed plate images enhanced and peaks offset for presentation.

3.9.2 Effects of Modifying Channel Orientation

The impact of developing on plates with different channel orientations is shown in Figure 3-6 for a deposition angle of $\alpha = 87^\circ$. Separation track angle increases with channel angle, while migration distances decrease. The dye spots are still resolvable up to channel angles of $\theta_C \sim 60^\circ$; beyond this point all analytes remain too close to the original application spot to be resolved. Figure 3-5 and Figure 3-6 qualitatively demonstrate the range of separations made available by the GLAD UTLC technique.

3.10 Quantification of Separation Performance

We have quantified our UTLC plate performance using migration distance, plate number and hR_F for Dimethyl Yellow (DY) and Ariabel Red (AR). The other dyes were too faint for this analysis.

3.10.1 Migration Distance

Analytes travelled further in the more porous (higher α) films. Figure 3-7a shows migration distance increasing nearly by a factor of two for all channel angles as deposition angle increased from $\alpha = 82.5^\circ$ to 87.0° . The largest increase was from 10 mm ($\alpha = 82.5^\circ$) to 22 mm ($\alpha = 87.0^\circ$) in the along channel direction ($\theta_C = 0^\circ$).

3.10.2 Plate Numbers

Plate number (N_i) was determined from the darkness chromatograms using the following equation [2]:

$$N_i = 4 \left(\frac{Z_i}{W_i} \right)^2 \quad (3-1)$$

where Z_i and W_i are the peak migration distance and full width half maximum of the i^{th} analyte. Plate number for DY as a function of deposition angle and channel angle is shown in Figure 3-7b. Plate numbers for DY approached 800 with AR substantially lower at well below 100. As expected, plate number improves with the increasing migration distances on plates with higher porosity. A decrease in plate number is observed with channel angle. Previously reported results for a deposition angle of $\alpha = 84^\circ$ are also given in Figure 3-7 for comparison³⁷. The results reported here are consistent with these previous results. However, the higher deposition angle used here produces a factor of two improvement in plate number.

3.10.3 Retention Factors

The DY and AR were easily distinguishable in the chromatograms shown in Figure 3-7c and Figure 3-7d, respectively. The hR_F values for DY ranged between 20 to 70, whereas the AR varied by only 10 (from 20 to 30). Retention decreases with increasing deposition angle, most dramatically at a deposition angle of 82.5° . It was not possible to determine the AR hR_F for $\alpha = 84^\circ$ and $\alpha =$

82.5° since this analyte failed to separate from the initial spot over the 90 second development time.

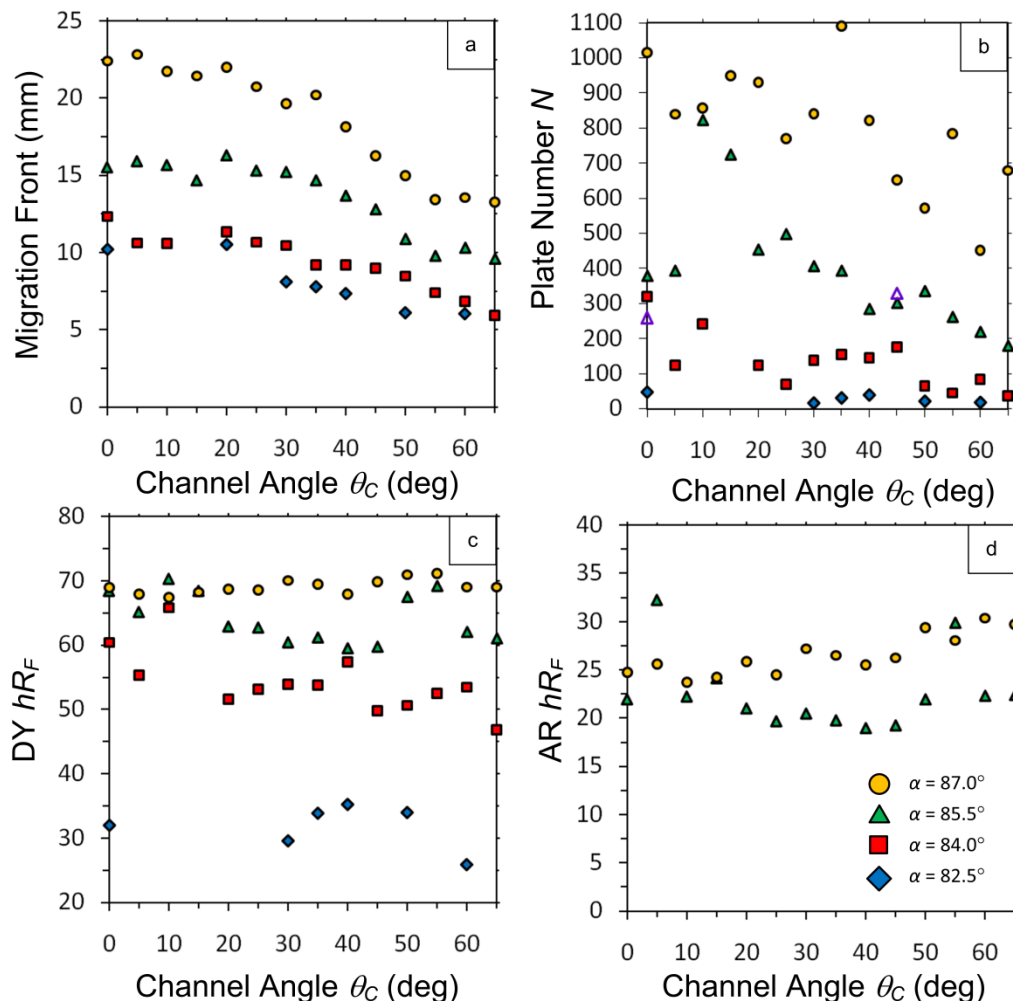


Figure 3-7: (a) Migration front distances for 4 different deposition angles. (b) The large effect of deposition angle on plate number is observed. The largest plate numbers obtained were for $\alpha = 87^\circ$ with a maximum of just under 800. Open purple triangles located at channel angles of 0° and 45° correspond to the results obtained previously [17]. The Dimethyl Yellow (c) and Ariabel Red (d) retention factors were mostly independent of angle for a given film deposition angle. However, a noticeable change (up to 2x) in the retention factors can be seen for a 4.5° swing in α .

3.11 Diffusivity Ratio

We define θ_C and θ_T as the channel and track angles, respectively. By projecting the development direction onto the channel and separation track vectors, the track deviation angle ($\Delta\theta$) can be shown to be:

$$\Delta\theta = \theta_C - \theta_T = \tan^{-1}\left(\frac{1}{\sqrt{R}} \tan \theta_C\right) \quad (3-2)$$

where R is the diffusivity ratio between the along and across channel directions, respectively. A fit of this model to all $\Delta\theta$ data is given in Figure 3-8, yielding a value for R of ~ 200 . No dependence on deposition angle is observed, consistent with the spot diffusion observations reported in Section 3.1. While it is difficult to obtain precise data for the higher channel angles due to the short migration distances, the inset shows good agreement between the model and the track deviations observed at higher angles. Previously reported $\Delta\theta$ values are consistent with those reported here³⁷. For additional verification of the model, testing very porous films at $\theta_C > 60^\circ$ or increasing development time would be required.

3.12 Potential Opportunities

The anisotropic GLAD-UTLC plates described in this chapter outperform those that were previously reported³⁷. The increase in film deposition angle from $\alpha = 84.0^\circ$ to $\alpha = 87.0^\circ$ produces wider channels (from 400 nm to 840 nm) that produce faster migration velocities enabling higher separation performance.

Although it is tempting to further increase the deposition angle, complications may arise as α approaches 90° . At extremely oblique deposition angles, the reduction in column density and surface area ⁵⁵ will likely reduce analyte retention and overall separating power. Other potential disadvantages include further increase in column orientation disorder (as noted above) and impractically low deposition rates due to the extreme angle.

3.12.1 Limits of Detection

One promising result of the work reported here is the clear observation of DY quantities well below the limit of detection (LOD) reported previously [Jim, 2010]. In that work, we estimated our LOD for DY on GLAD UTLC plates to be ~ 10 ng. The present paper demonstrates that DY masses as small as ~ 3 ng may be separated and detected on GLAD-UTLC media with similar equipment. The improved LOD is attributed to the use of smaller initial spots and suggests that sensitivity is currently limited by sample application methods. Further improvements in LOD and overall separation performance can be expected if the analytes were applied with an inkjet printer. The smaller spots from inkjet application motivate continued work in coupling GLAD UTLC media with the Office Chromatography concept ⁵³.

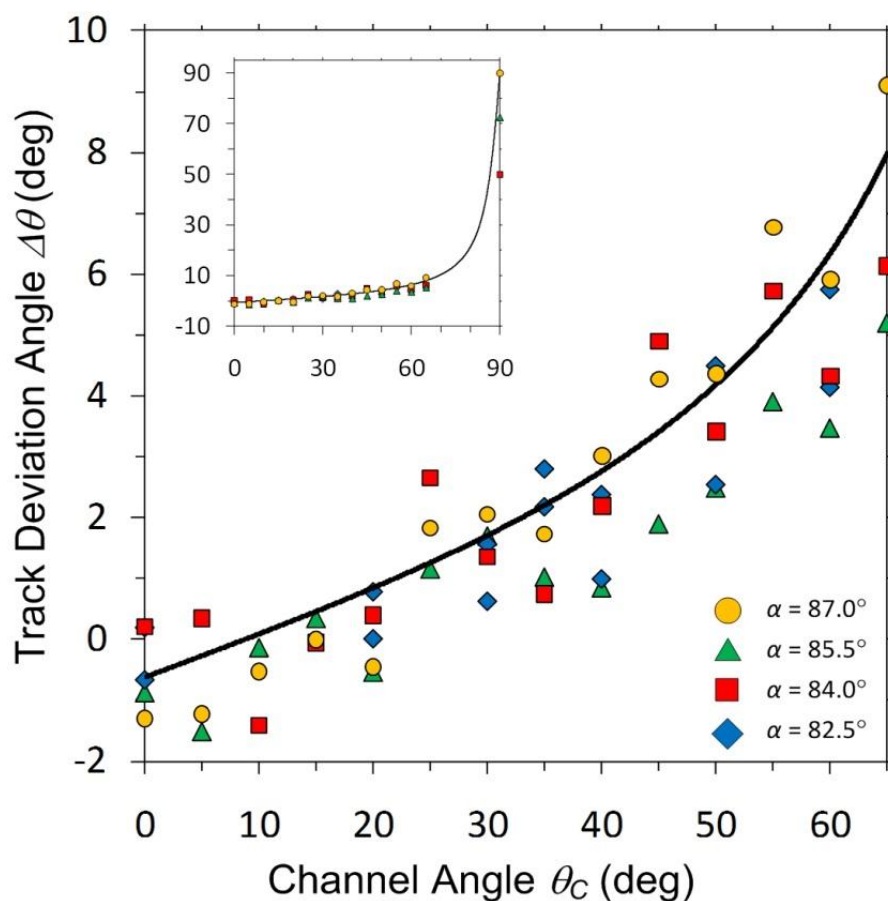


Figure 3-8: Separation track deviation angle ($\Delta\theta$) as a function of channel angle (θ_C) for films grown at different deposition angles (α). $\Delta\theta$ is independent of deposition angle α . The solid line is a best fit of Eqn. (2) to the entire data set. Inset shows good agreement between model and experimental data at very high channel angles.

3.13 Conclusion

We investigated the channel-like features characteristic of anisotropic GLAD-UTLC media and their significant effects on chromatographic behaviour. We quantified GLAD-UTLC plate performance by characterizing migration distance, plate number, retention factor, and separation track deviation angle

($\Delta\theta$)—a figure of merit specific to anisotropic GLAD-UTLC. Migration distance and plate number increase with thin film deposition angle (film porosity) but decrease with channel angle (θ_C). We proposed a theoretical model describing $\Delta\theta$ that relates analyte migration to anisotropic film microstructure. While not a focus of this study, we estimate Dimethyl Yellow LOD to be better than the ~ 10 ng value previously reported, likely due to the smaller initial analyte spots in this work. Application of sample spots with smaller volumes is expected to further improve LOD.

Channel features reduce transverse (across-channel) spot broadening during analyte application and elution while providing the wide (along-channel) pores required for rapid migration and high separation performance. As the channel structures reduce the interference between adjacent separation tracks, they enable the parallel analysis of several sample mixtures on the miniaturized GLAD-UTLC plate formats. Coupling the Office Chromatography (OC) suite, which uses precise inkjet analyte application and high resolution chromatogram imaging, with our GLAD-UTLC plates may significantly enhance performance. An optimized OC-GLAD-UTLC system that combines the benefits of office peripherals with the high performance of anisotropic nanostructured chromatographic media may become a part of the analytical chemist's toolset.

4 * Time-Resolved UTLC

4.1 Introduction

UTLC offers many potential advantages over more traditional planar separation methods such as TLC and HPTLC. Separations on monolithic silica gel UTLC plates (Merck) can occur quickly (< 7 min) and over short distances (< 30 mm)⁶². Such separations use lower analyte and reagent volumes and can be more sensitive^{15,63}. UTLC enables parallel separation of densely spaced sample spots, making it ideal in areas such as food chemistry⁶⁴.

4.1.1 GLAD UTLC

Excellent control over stationary phase morphology and compatibility with post-processing techniques make GLAD a powerful platform for engineering high-performance UTLC separation media. However, full utilization of GLAD-UTLC microstructures requires a new perspective on the fundamental characterization of chromatographic physics. Manual plate operation and post-separation characterization may not always achieve the best possible chromatographic result given the enhanced speed of these separation media. Performance improvements may be achieved by incorporating real-time video analysis.

* This chapter has been modified from Oko *et al.*⁷⁹ with permission from Elsevier

4.1.2 Multiple Images of Single Plate Development

The concept of real-time analysis in TLC has been examined in recent studies in which multiple still images were obtained at defined times during the development. Lancaster *et al.*⁴⁹ demonstrated an improved signal to noise ratio of a single test dye by amalgamating snapshot images taken every 6 s on a TLC plate. In another study, Berezkin and Chausov⁴⁸ recorded and analyzed separations on circular TLC plates for every 1 cm movement of the mobile phase front. Zhang *et al.*⁶⁵ analyzed digital images taken several seconds apart during capillary-driven coloured dye separations performed on particulate-packed microcapillaries to study dye movement and confirm TLC-like elution behaviours. However, these techniques have not yet been applied to miniaturized UTLC plates that achieve faster separations over shorter distances. Automated rapid development data collection provides more insight into plate performance and could eventually lead to analyte quantification improvements.

4.1.3 Quantitative Analysis

Quantitative TLC using charged coupled device sensors has been studied extensively^{41,42,44,66–69}, in which a single plate image is taken at the end of the development. In much of this work, analytes were detected by measuring the intensity of UV fluorescence, UV absorption, visible light reflection, or visible light absorption. TLC performance quantification has also been performed using color model analysis, in particular chromaticity (saturation) measurement of analytes imaged under white light⁷⁰.

4.1.4 Chapter Overview

We use a customized video analysis approach to investigate time-resolved performance of a Merck UTLC plate. A consumer high definition video camera (1080p, 30 frames per second) provided the frame rates and image resolutions necessary to characterize the UTLC's short development times and distances. A custom MATLAB script analyzes the video and reports figures of merit as functions of elution time.

4.2 Detection Method

Our detection method is *in-situ*, using a high-definition video camera (Canon EOS Rebel T2i digital single lens reflex; capable of capturing 1920 x 1080 pixel frames at 30 frames per second) with a dedicated macro lens (Canon EF 100 mm f/2.8 USM) operating in the visible spectrum. The camera utilizes a CMOS image sensor and applies variable bit rate MPEG-4 AVC/H.264 video compression with 4:2:0 subsampling. The camera is oriented normal to the UTLC plates and contained in a custom PVC enclosure to protect the plastic camera body from organic solvent vapour (Figure 4-1). A 10 inch square glass window was sealed to the bottom of the enclosure and placed directly between the camera and the horizontal development chamber (Desaga H-Chamber, 50 mm x 50 mm, Sarstedt, Nümbrecht, Germany). An adjustable stand for the enclosure was used to obtain the highest possible quality spatial resolution (32 μm ,



Figure 4-1: Camera setup and apparatus used to take high-definition video of UTLC plate during development. A black cloth is draped around the system to control lighting during the development. The camera is isolated from the development chamber in order to protect the camera lens from any potentially corrosive vapours. The arrow indicates the plate development direction.

measured with 1951 Air Force Test Pattern, Thorlabs R3L3S1N), in which all UTLC separation tracks could be contained within the camera's field of view. A black cloth was draped around the enclosure to prevent exterior lighting from entering the camera. A direct USB feed-through connection was used to transfer video data to a computer's hard drive without breaking the seal. Two white LED strips (5500 K, theledlight.com, Inc., Carson City, NV) illuminated the

development chamber from opposite sides of the enclosure, perpendicular to the elution direction at $\sim 45^\circ$ to plate normal. Each strip consisted of 12 LED's with luminous intensity of 5000 mcd per LED. Illuminating from only these two sides reduced any unwanted reflections and shadows from the horizontal development chamber surface.

4.3 Preparation and Spotting Technique

To evaluate our detection method, dye separations performed on a UTLC plate (Merck, Darmstadt, Germany) were analyzed. The analyte was undiluted Test Dye Mixture III (CAMAG, Muttenz, Switzerland). In order of elution, this undiluted mixture comprised of 3 mg of N,N-dimethylaminobenzene (*DY* – dimethyl yellow), 0.5 mg of sudan IV (*OR* – oracet red), 2 mg of sudan blue II (*SB*), 1.5 mg of ariabel red (*AR*), 4 mg of indophenol (*IP*), and 2 mg of ciba F-II (*OV* – oracet violet) per mL of toluene. All plates were hand-spotted using a 30 gauge blunt end stainless steel needle (I&J Fisnar, Wayne, New Jersey, USA) dipped into the diluted dye mixture. The needle was pressed against the UTLC plate for approximately 1 second.

4.4 UTLC Plate Development

Toluene ($\geq 99.5\%$, Sigma-Aldrich Canada, Oakville, Canada) was used as the mobile phase for developing the Merck UTLC plate. We filled the conditioning trough with toluene to improve vapour phase saturation within the horizontal development chamber. A porous glass frit dipped into the chamber reservoir supplied mobile phase to the sorbent layer.

4.4.1 Camera Settings

The macro lens was focussed before development by placing into the chamber a glass piece with markings scratched into its underside. The camera lens' focus and depth of field were manually adjusted through the Canon EOS Utility computer interface (Version 2.8.1.0, Canon Inc.) to ensure that the image of the scratched markings appeared clear and sharp. System parameters such as lighting intensity, camera aperture size (F3.2), ISO setting (400), and shutter speed (1/60) were all optimized and kept constant during the course of the video. After starting the video recording, the spotted UTLC plate was developed in the horizontal chamber until the mobile phase front reached the end of the plate, removed, and then dried under a warm air stream. This procedure records the full UTLC development, capturing the set of all possible development times and ensuring the best separation performance is recorded for each analyte component.

4.4.2 Frame Extraction

Following development, the video data was downloaded directly to the computer's hard drive. Ten full-resolution (1920 x 1080 pixels) video frames per second were extracted from the separation video using ZoomBrowser EX software (Version 6.5.1.15, Canon Inc.). In this case (Merck UTLC plate with development duration in the minutes), we deemed it unnecessary to extract the full 30 frames per second. Any time-dependent features of interest could easily be obtained using 10 frames per second. The first extracted frame marked the instant that the mobile phase entered the stationary phase; the last extracted frame

was taken as the moment before the plate was manually removed from the chamber.

4.5 Identification and Analysis of Dye Components

Custom programming code written for MATLAB (Version 7.8 R2009a, MathWorks Inc.) was used to generate time-resolved chromatograms from each chromatography video. Analysis involved four main steps, outlined briefly here and described in full detail in the upcoming sections. Prior to analysis, all pixels within each extracted frame were converted from RGB (red, green, blue) to HSV (hue, saturation, value of brightness); we use the HSV colour model [Marcu, 1995] throughout this work. First, each analyte was assigned a particular hue range, where hue is constrained over the interval of 1 to 360 degrees (see Appendix A.1.1.1 for code). Second, a recipe file was created that contains development information from the UTLC video such as mobile front flow constant, analyte hue ranges, and coordinates of the separation tracks and initial spot locations (see Appendix A.1.1.2 for code). Third, color-filtered chromatograms were generated and Gaussian fits applied for each analyte (see Appendix A.1.2 for code). These chromatograms were produced for every track from each extracted video frame. Migration distances and peak widths of each analyte were calculated from the fits. Lastly, all relevant figures of merit (retention factors, theoretical plate numbers, separation numbers, and resolution between zones) were generated, and reported as functions of development time (see Appendix A.1.3 for code).

4.6 Validation of HD Camera Color Reproduction

4.6.1 Color Standard Evaluation

We used a color standard (Munsell ColorChecker Chart, ColorAccuracy.com) to validate the HD video camera's color fidelity. The color chart was placed below the camera enclosure, in place of the development chamber, using the same system conditions used during UTLC plate development discussed earlier. A video of the color chart was obtained and a single frame extracted (no image enhancements were performed). MATLAB was used to crop each of the non-neutral colors swatches (hue of a neutral color is undefined). The average hue angle of each cropped image was compared to hue values calculated from the color standard's certified RGB values (Figure 4-2).

4.6.2 System Isolation from External Lighting

The importance of preventing exterior lighting from entering our imaging system can be seen by comparing Figure 4-2a with Figure 4-2b. Without the black cloth enclosing our system (Figure 4-2a), our device's calibration curve shows significant nonlinearities, poorly reproducing the color. Covering the imaging system with an opaque cloth improved control over lighting conditions and hue value consistency. Moreover, color saturation values are strongly influenced by lighting conditions and camera settings. We found it vital to ensure the system parameters listed above were held constant during UTLC development. With the black cloth enclosing our system, the hue angle reported

by our device was found to agree well with the Munsell color standard with a slope of 0.99 ± 0.02 and an intercept of $2^\circ \pm 4^\circ$.

4.6.3 Compression Algorithms

The digital camera's video compression algorithms (MPEG-4) may have introduced noise and color distortion effects. While we believe these influences

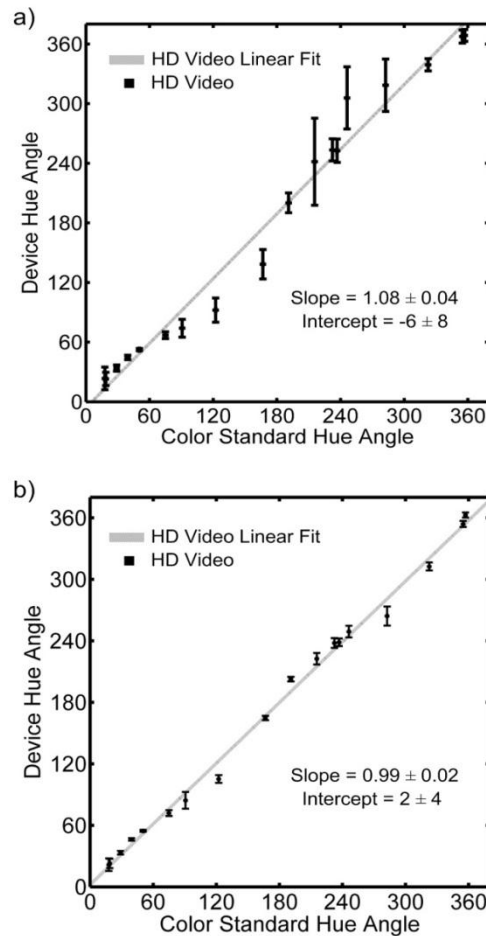


Figure 4-2: The hue angle reported by our system against a color standard (Munsell ColorChecker Chart). (a) A linear fit for the uncontrolled lighting has a slope of 1.08 ± 0.04 , and intercept of -6 ± 8 . (b) If lighting conditions are controlled, the linear fit has a slope of 0.99 ± 0.02 , and intercept of 2 ± 4 indicating excellent agreement with the color standard.

to be slight, several methods of reducing these artifacts currently exist. Recent higher-quality compression extensions of the H.264 standard do not subsample color information and preserve full color resolution ⁷¹. More expensive higher performing video recording devices can also output raw sensor data without introducing additional compression noise or color alterations.

4.7 Hue Filters

Automated color-filtered chromatogram analysis required selection of an appropriate hue range (color filter) for each dye component (Figure 4-3). We manually cropped regions corresponding to each analyte from a video frame in which all dyes can be distinguished. The hue values for each pixel in these regions were binned into a histogram to produce the hue range for each analyte component. The calculated saturation values were summed over all pixels in the cropped region that had a given binned hue value. This produced a signal that incorporates the frequency of pixels with a given hue and the saturation magnitude of pixels with that hue. For example, in the case of the DY signal, a peak at hue $H = 65^\circ$ suggests that the cropped region has many pixels with that hue and that these pixels have large saturation values. The summed saturation signal for given hue can be written as

$$A(j) = \sum_{i=1}^{W \cdot L} S_i \delta_{Hj} ; \quad \{j \in \mathbb{Z} | 1 \leq j \leq 360\} \quad (4-1)$$

$$\text{where } \delta_{Hj} = \begin{cases} 1 & \text{if } H = j \\ 0 & \text{else} \end{cases}$$

where A is the sum of saturation signal, i is the pixel number, W and L are the width and length in pixels of the cropped image, j is the bin number, S is the saturation value, and H is the hue value. The hue range for each analyte is manually determined as shown by the dashed lines in Figure 4-3b. Although *OR* is below our system's detection limit, it was faint but clearly visible by eye. No summed saturation plots are shown for this component.

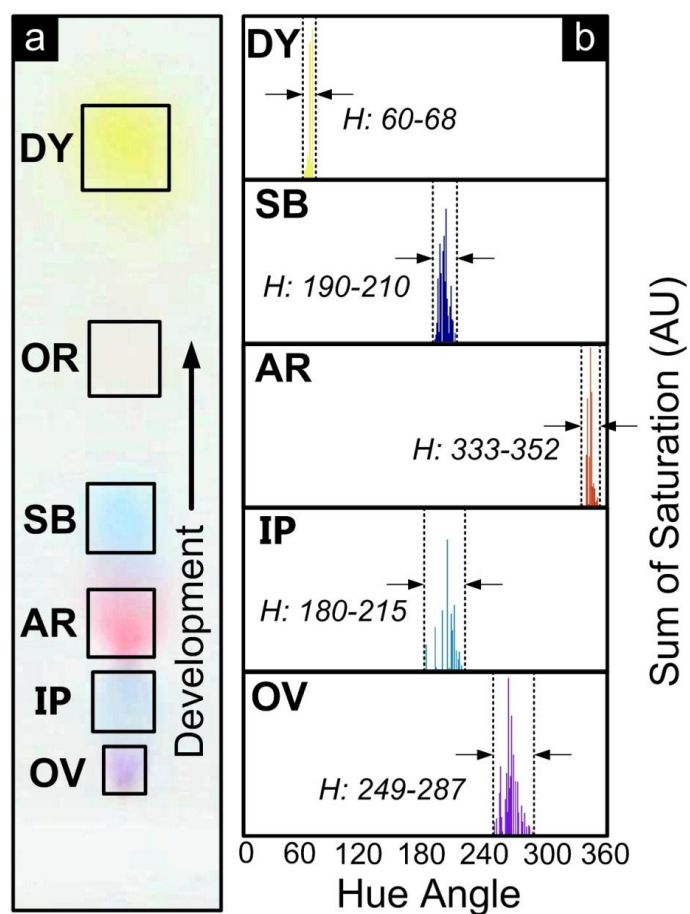


Figure 4-3: Cropped areas of each analyte from a particular development track (a) are used to generate histograms of sum of saturation signals for given hue (b). Analyte hue range selection enables calculation and analysis of time-resolved UTLC chromatograms.

4.7.1 Post Processing

Image post-processing techniques such as background and noise correction^{46,47,72} could enable identification and further analysis of the *OR*. Whole-image denoising techniques incorporating multiple frame averaging could also be exploited as a method to improve detection⁷³. However, here we have chosen to avoid any image post-processing as it is currently unclear whether these non-linear techniques would affect the validity of our results.

4.8 Generation of Time Resolved Chromatograms

4.8.1 Separation Track Cropping

Figure 4-4a show how a track is cropped from one of the extracted frames in the UTLC video. A surface plot (Figure 4-4b) of the same extracted frame gives a visual representation of spot intensity (saturation), which is used for subsequent analysis. The cropped track was made sufficiently wide as to fully contain the largest spot (typically the *DY*). Each track extended the length of the video frame (1080 pixels). The track's position and width were held constant throughout the development.

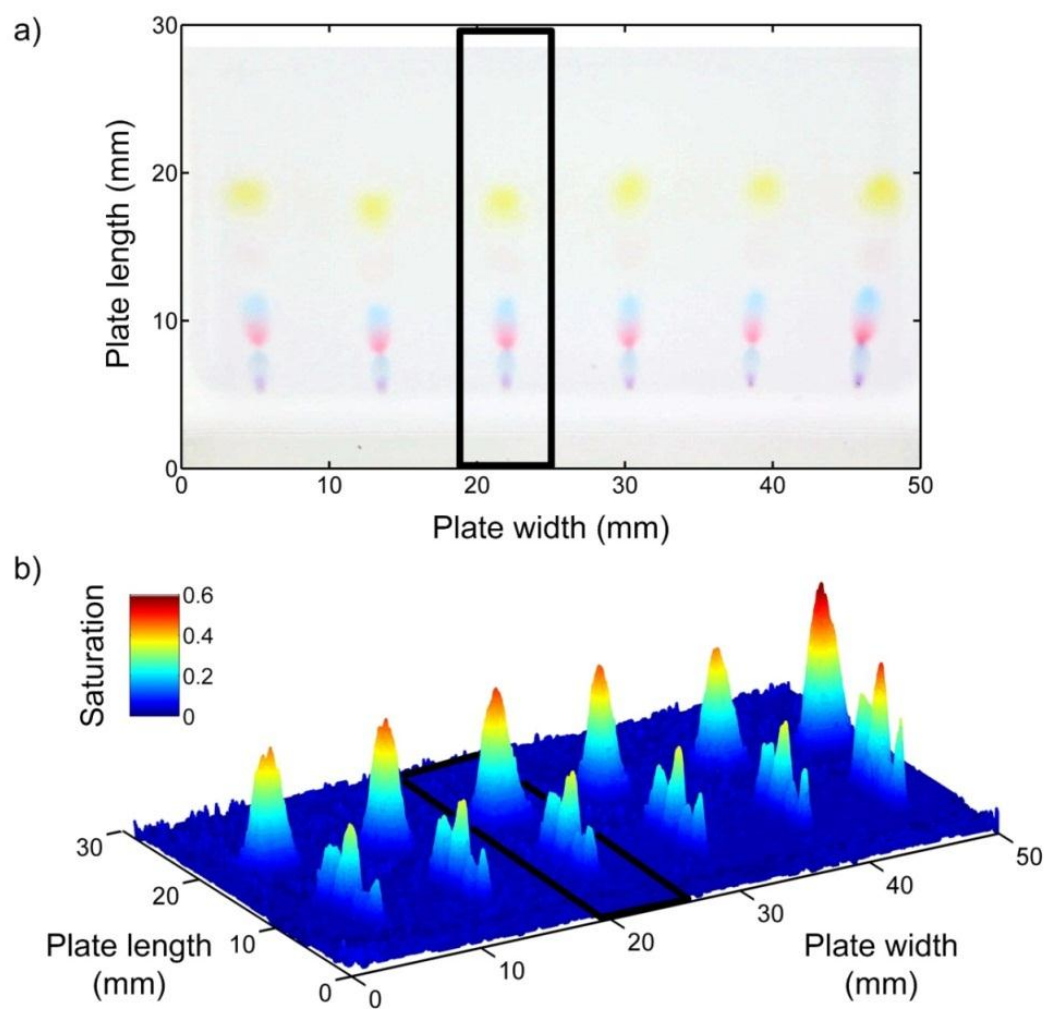


Figure 4-4: A single frame from a separation video (a), showing the test dye mixture separated in six tracks. A three dimensional view (b) of saturation for each pixel in the frame. The black box in (a) and (b) outline the same track. No color (hue) filtering was performed when calculating these signals.

4.8.2 Chromatogram Production

A raw chromatogram for each analyte was produced by summing the saturation values of each pixel for every row of the specified cropped track. Only pixels with hue values satisfying each analyte's hue filter were considered. This

allowed us to track individual analytes that were poorly separated; essentially a method of deconvolution (see Figure 4-5). However, analytes that are completely overlapped (e.g., during the initial stages of development), cannot be deconvoluted using hue filters. Colour mixing of two dyes may also cause pixel hues to lie outside of the hue filters defined for these dyes. The saturation values of such pixels would therefore be excluded from the hue-filtered signal and subsequent peak fitting. The raw chromatogram can be mathematically described as

$$D_k(t, x) = \sum_{y=1}^W S(x, y) ; \{H_{k,min} \leq H(x, y) \leq H_{k,max}\} ; \{1 \leq x \leq L\} \quad (4-2)$$

where x is the distance along the track length, y is the distance along the track width, k is the analyte, t is the time into development, W and L are the width and length of the separation track, S is the saturation, and H is the hue.

4.8.3 Background Correction

The noisy background observed in the unfiltered ($0 \leq H \leq 360$) chromatogram in Figure 4-5 can be attributed to pixels of a characteristic hue range ($110 \leq H \leq 160$) outside any of the defined analyte hue filters. It was therefore unnecessary to correct for this background. Conversely, correction methods would have been required if analyte detection within this background hue range was desired.

4.8.4 Peak Curve Fitting

Retention factor bounds for each analyte were defined in the ‘recipe file’ to ensure that Gaussian peak fitting was performed only over regions of D_k known to contain a spot. This was necessary as background noise or plate surface defects

(such as scratches) created spurious peaks. An unconstrained nonlinear optimization technique (least squared errors using the 'fminunc' function in MATLAB) was used to apply Gaussian fits to each color-filtered D_k signal within the defined migration bounds. Figure 4-6 is the result of superimposing these Gaussian fits for all analytes on a particular track. In this instance, the symmetric peaks in each hue filtered signal were well represented by Gaussian peak fits. However, our method could easily be adapted to accommodate non-Gaussian (asymmetric) peaks better described by alternative models ⁷⁴.

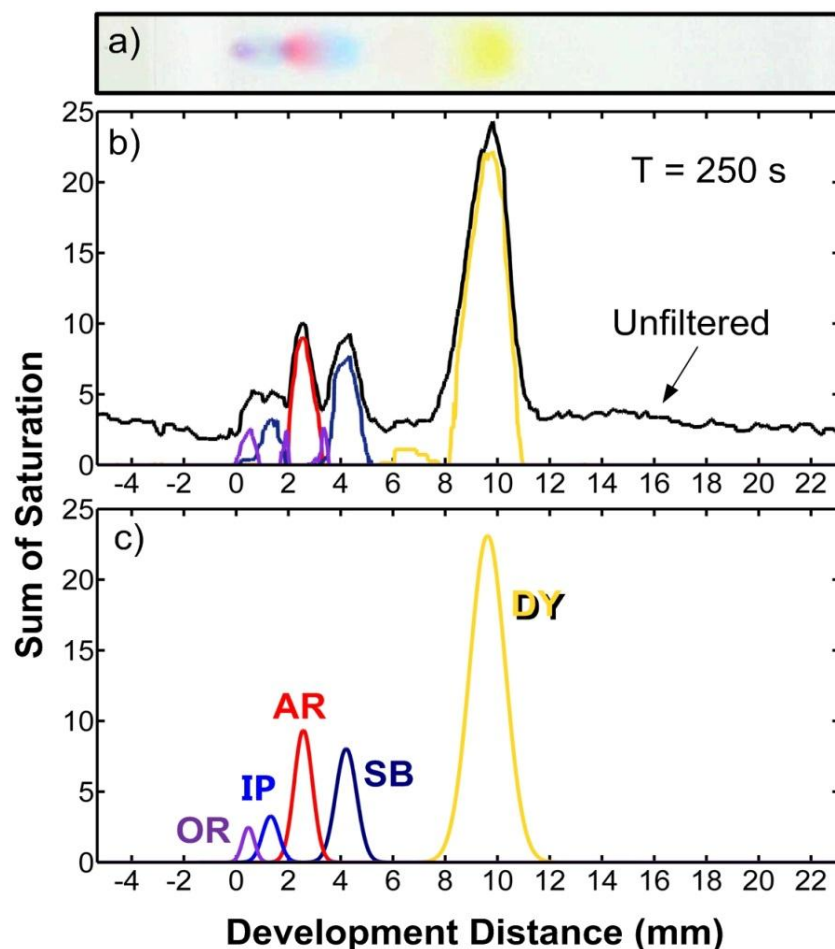


Figure 4-5: (a) A single frame 250 seconds into development is shown with coloured dyes exhibiting different degrees of separation. (b) Chromatograms with (coloured lines) and without (black line) hue filtering. (c) Gaussian fits to the hue-filtered chromatogram peaks. Hue filters allow deconvolution of partially separated peaks. The separation track image has been digitally enhanced for clarity.

4.8.5 Color Mixing

In the initial stages of development, poor dye separation makes analyte tracking difficult, especially for those that are strongly retained. Early on in the development (Figure 4-6a, $t = 45$ s), our automated system is deceived by this color mixing and overestimates the peak width of the *OV* while overlooking other

analytes. During the middle of the development (Figure 4-6b, $t = 150$ s), all of the dye components have separated sufficiently and can be fit more reliably. By the end of the development (Figure 4-6c, $t = 415$ seconds), every analyte is easily discernible and can be fit accurately with a Gaussian curve.

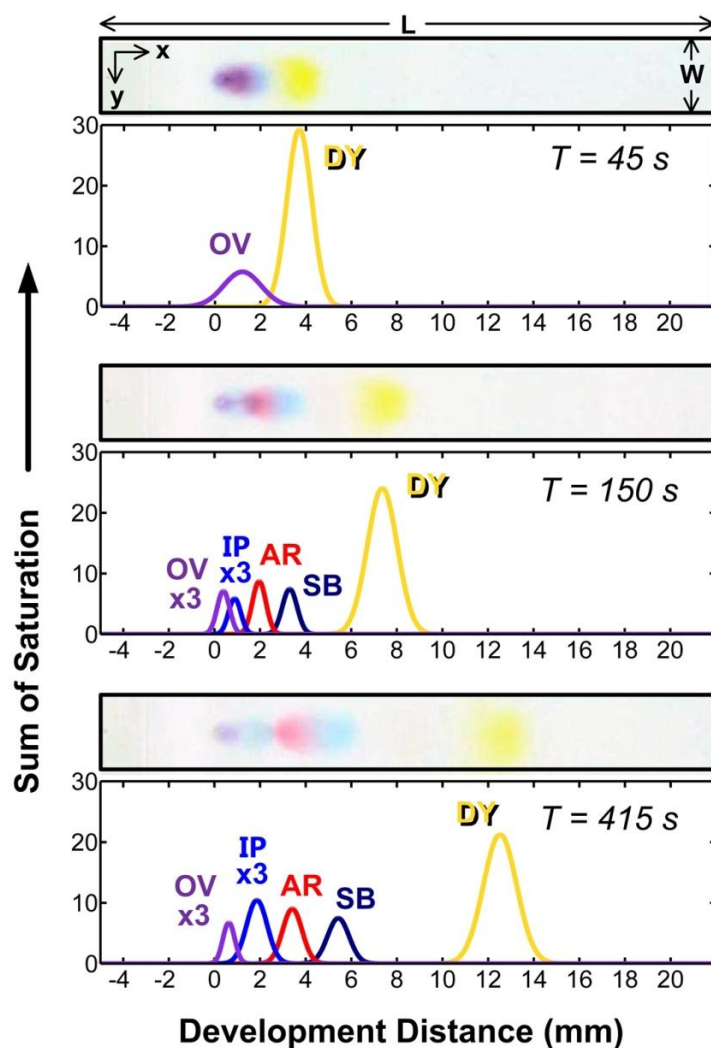


Figure 4-6: Chromatograms from the fitted Gaussian peaks for each analyte at various times throughout the development. Peaks become increasingly well resolved as the development continues. Track images have been digitally enhanced for clarity.

4.8.6 Time Dependent Analysis

Time-dependent peak migration distances (Z_k) and peak standard deviations (σ_k) calculated from the Gaussian fits are shown in Figure 4-7. Our automated system does not generate valid data during the first stages of the development, especially for analytes with high retention. The first peak for the *OV* in Figure 4-7b can again be explained by color mixing with the *IP*. The slope of each analyte's peak width curve can be related to its corresponding diffusion coefficient. The *IP* is shown to diffuse the quickest while *OV* exhibits the least broadening.

Due to circular chromatography effects, first studied by Rutter⁷⁵, occurring during spotting, our analytes experience a virtual focussing effect before broadening in the development direction. Low retention analytes reside near the outside rim of the initial spot. As mobile phase is introduced to the spot, the back end elutes first and is drawn closer to the front end, giving rise to a virtual focussing effect. The effect, although not as evident, can also be observed for analytes with higher retention.

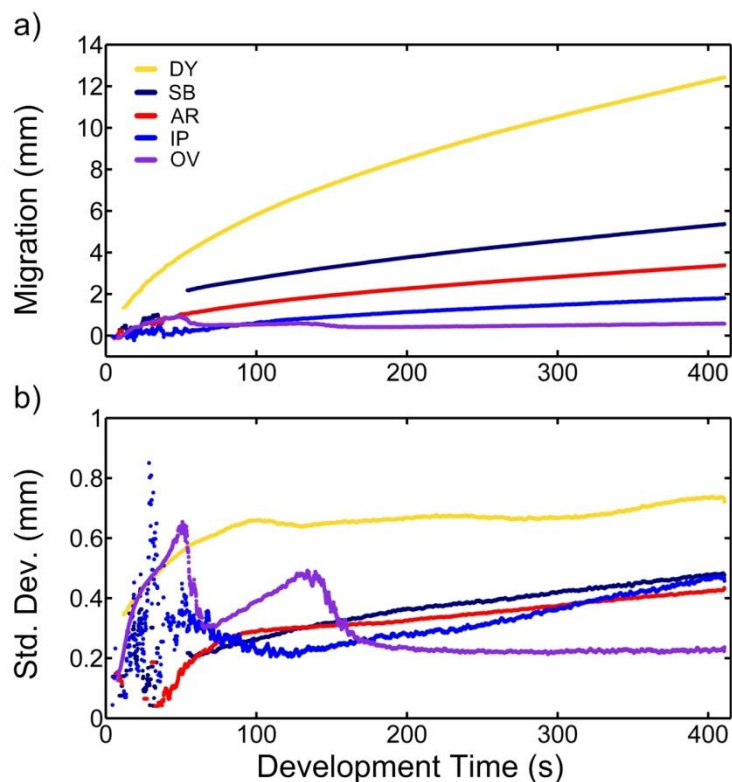


Figure 4-7: Time-dependent figures of merit including retention factor (a), plate number (b), separation number (c), and resolution (d). Resolutions are calculated between adjacent peaks in all cases except for that between DY (eluted first) and SB (eluted third). The signal for the OR (eluted second) was too weak for this analysis. A 2 second (20 frames) moving average is applied to all curves.

4.9 Figures of Merit

4.9.1 Mobile Flow Constant

Incomplete vapour-phase saturation is known to cause uneven solvent fronts ⁷⁶, which is particularly noticeable on UTLC plates ⁴⁵ due to their small size. A bowed mobile phase front position is observed in our work which may be attributed to this insufficient saturation. Thus, determining a single mobile flow

constant for all development tracks on a plate is inappropriate. Rather, a local flow constant is determined and defined in the recipe file for each track before automated analysis. The relationship between front position (Z_F) and the mobile flow constant (χ) is

$$Z_F = \sqrt{\chi \cdot t} + Z_{start} - Z_{spot} \quad (4-3)$$

where t is the time into development, Z_{start} is the distance from the image's edge in which the mobile phase first enters the stationary phase via the porous frit, and Z_{spot} is the distance from the image's edge at which the initial spot is applied. By accounting for Z_{start} we obtain more accurate front locations for early stages in development. Once they are separated, each of the analytes exhibits a nearly constant retention factor, $hR_F = 100 (Z_k / Z_F)$ (Figure 4-8a), permitting us to set the retention factor bounds for each analyte in the 'recipe file'. We used this to our advantage to isolate analytes with similar hue that were well separated.

4.9.2 Time Resolved Plate Numbers

Excluding IP , and disregarding the virtual focusing described above, theoretical plate numbers, $N = (Z_k^2 / \sigma_k^2)$, increased for all analytes over the duration of the development (Figure 4-8b) after the initial mixture had been separated (~ 50 s). In this particular case, migration distance is dominant over longitudinal broadening. Theoretical plate number curves for all analytes appear to plateau near the end of the development with the largest PN occurring for DY , at nearly 300.

4.9.3 Time Resolved Separation Numbers

Due to the virtual focussing effect discussed earlier, it becomes difficult to accurately determine the separation number (SN), predominantly early on in development at times < 50 s. As longitudinal broadening prevails over the virtual focussing, separation number can be defined by⁷⁷

$$SN = \frac{Z_F}{b_1 + b_0} - 1 \quad (4-4)$$

where Z_F is the front distance from the initial spot location, while b_1 and b_0 are the extrapolated full width half maximum at $hR_F = 100$ and $hR_F = 0$, respectively. Similarly to N , Figure 4-8c shows SN increases monotonically and surpasses 7 at 400 s.

4.9.4 Time Resolved Resolution

Another important metric is resolution (RS) between adjacent spots and can be obtained by¹⁸

$$RS = \frac{1}{2} \cdot \frac{Z_{k,2} - Z_{k,1}}{\sigma_{k,2} + \sigma_{k,1}} \quad (4-5)$$

A nearly complete separation between neighboring analytes is achieved for $RS \geq 1$, at which point there is only a 3% overlap. As shown in Figure 4-8d, both the $AR:IP$ and the $IP:OV$ curves approach $RS = 1$ but fail to cross this boundary. As noted by Spangenberg¹⁸, if interested in separating two specific analytes with similar elution strength, mobile and/or stationary phase should be optimized such that the average analyte hR_F value lies at $100/3$. The low (< 15) hR_F values of AR , IP , and OV fall too far from the optimized value of $100/3$ to

separate adequately. Unfortunately, limits of detection cannot be reported as we do not have precise control over spot volumes in this work.

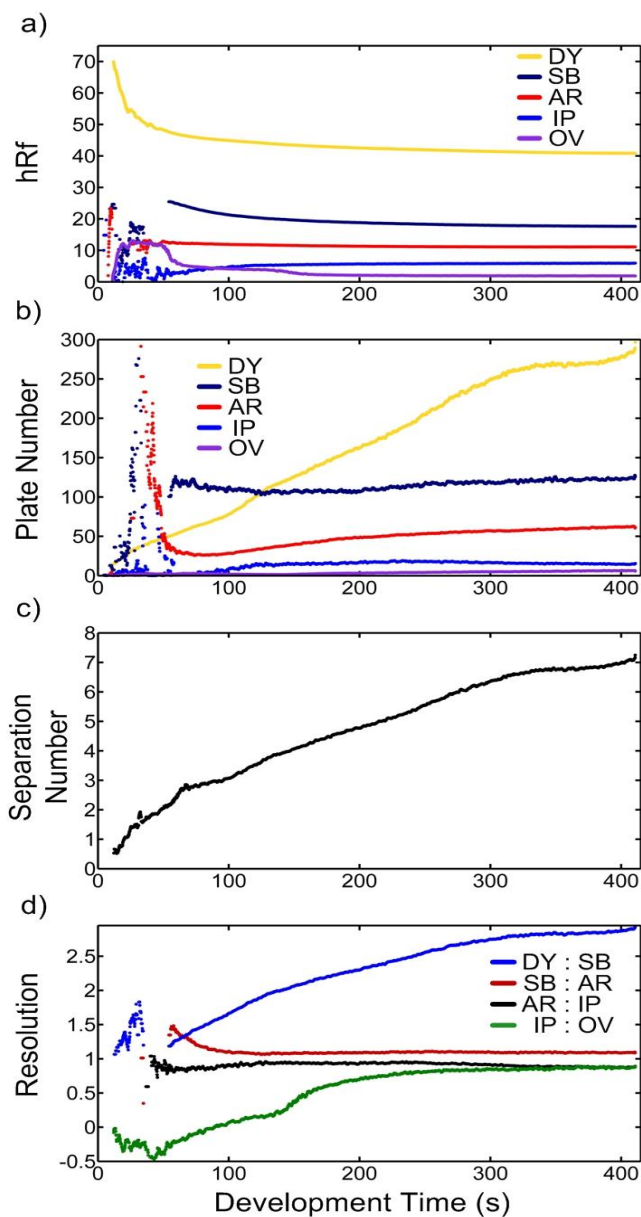


Figure 4-8: (a) Time-dependent migration distances and (b) peak standard deviations of each analyte are shown. A 2 second (20 frames) moving average was applied to all curves.

4.10 Prospective System Applications of TR-UTLC

TR-UTLC offers many potential advantages over alternative planar chromatography detection methods. The relatively simple and inexpensive system enables rapid automated analysis. Although this investigation focussed on colorful analytes that may be detected visually, incorporation of UV-fluorescent separation media or an appropriate imaging system (e.g. CCD, intensified CCD) could extend this method into the ultraviolet realm. Human error can play a detrimental role in obtaining desirable results for UTLC separations, and the inherently small size and short migration distances make UTLC separations more sensitive to human mishandling, especially evident in GLAD-UTLC media. By reducing human interactions, acquiring consistent separation analysis for identical plates is more likely using the TR-UTLC approach; an important characteristic for quantitative study. TR-UTLC automatically consolidates an extraordinary amount of data obtained in a single development and allows users to better assess plate performance using any one of the many attainable figures of merit. The computational time (using current MATLAB scripts) takes just under one second and generates approximately 25 kB of data per frame analyzed. Hence, a complete development, at a typical frame rate of 10 s^{-1} , will routinely produce over 100 MB of data. This system could potentially be used as a tool to optimize certain chromatographic parameters such as mobile phase composition, sample spot loading, and even to improve stationary phase fabrication methods.

4.11 Conclusion

In this chapter, we designed and implemented a TR-UTLC imaging system using a commercially available high-definition video camera. In our setup we measured a UTLC plate under a white light source. Our system was used to record high-definition videos of test dye mixture separations performed on monolithic silica gel UTLC plates. The RGB values from the extracted frames were transformed to HSV values which were used in subsequent analysis. We verified the camera's color fidelity using a reflective color standard, which showed excellent agreement under controlled lighting conditions. An algorithm was developed for identifying color peaks, giving us the capability to generate and analyze time-resolved chromatograms. Figures of merit such as analyte migration distance, peak width, theoretical plate number, resolution, and separation number were automatically calculated as functions of development time. Combining TR-UTLC with the office chromatography concept⁵³, in which office peripherals replace traditional planar chromatography equipment, could eventually lead to an effective, simple, and inexpensive approach for UTLC quantification and investigation. Gaining a deeper awareness into the intricacies of UTLC by time-resolved measurements ultimately gives the researcher more control and may eventually serve as a platform for optimizing other planar chromatographic separations, such as in TLC or HPTLC.

5 Transmission Mode Video UTLC

5.1 Introduction

Several detection schemes have been used successfully for TLC in the past including transmittance, absorption, fluorescence and reflectance^{41,49,67,69}. In this chapter I will discuss a new detection approach that utilizes the transmission detection mode in combination with the TR-UTLC method introduced in Chapter 4. With the TR-UTLC system we were able to demonstrate many benefits, such as offering further insight into the separation dynamics of TLC. However, it proved especially difficult to de-convolve spots that fail to sufficiently separate. This could potentially result in unacceptable detection reliability, particularly early on in developments.

This chapter is meant as a proof of concept to demonstrate an innovative approach that improves upon the TR-UTLC system early on in developments, where spots have failed to completely separate. The white LED light source used in the TR-UTLC system is limited by color mixing, due to overlapping spots. Rather than a simple white light source, a more complex system of colored LEDs controlled by an Arduino microcontroller is implemented. Each colored LED is preferentially responsive to particular visible spots, allowing more reliable detection in the case of poorly separated analytes. For simplicity, rather than using the HSV color model, described in depth in Chapter 3, the lightness (1 –

darkness) model ³⁷ utilized in Chapter 2 is used to produce chromatograms in this work.

5.2 Experimental Setup

The work done here employs a similar system setup as used in Chapter 3 (Figure 3-1). A high-definition video camera mounted inside a customized PVC enclosure records the development of UTLC plates.

5.2.1 Development Chamber Design

A new development chamber is utilized that encompasses a transparent back glass cover (Figure 5-1). Due to the lack of a conditioning trough only the reservoir is filled with mobile phase. This condition likely results in insufficient

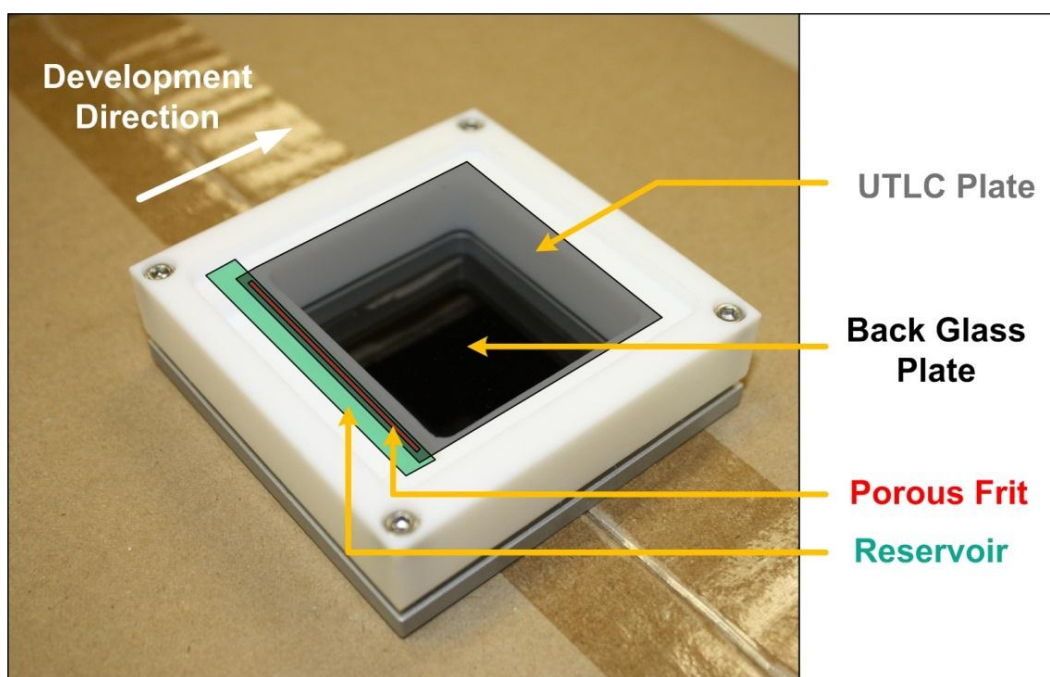


Figure 5-1: Development chamber used for transparent mode video ultrathin layer chromatography.

chamber saturation, thus affecting separation quality. It is important to note that the separation quality is insignificant in this case (a poor separation is actually preferred), as the goal of this work is to show the power of this new technique when detecting incompletely separated spots.

5.2.2 LED Source Control

Another system parameter that differed from that used in Chapter 3 was the LEDs used and their mounted locations. A total of 18 LEDs were employed; six each of red (Lite-On Standard LED -- through hole, P/N: 859-LTL-353CKR-H4, Mouser Electronics Inc.), green (Lite-On Standard LED -- through hole, P/N: 859-LTL353TGK, Mouser Electronics Inc.), and blue (VCC Standard LED -- through hole, P/N: 593-VAOL-5GSBY4, Mouser Electronics Inc.).

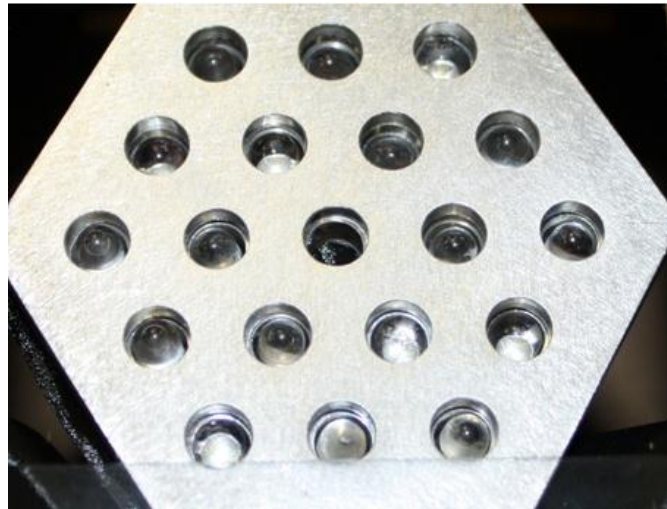


Figure 5-2: Customized aluminum LED holder consisting of 19 holes. The middle hole is left empty, while the remaining 18 holes are used to evenly distribute 3 different colored LEDs.

The LEDs were mounted in a custom ‘honeycomb’ holder (see Figure 5-2) constructed of aluminum and placed beneath the development chamber. A diffuser (Fisher Wide Mouth Bottle, HDPE, Fisher Scientific) was also placed between the LEDs and the development chamber to provide a more uniform light source. A descriptive diagram of the setup is shown in Figure5-3.

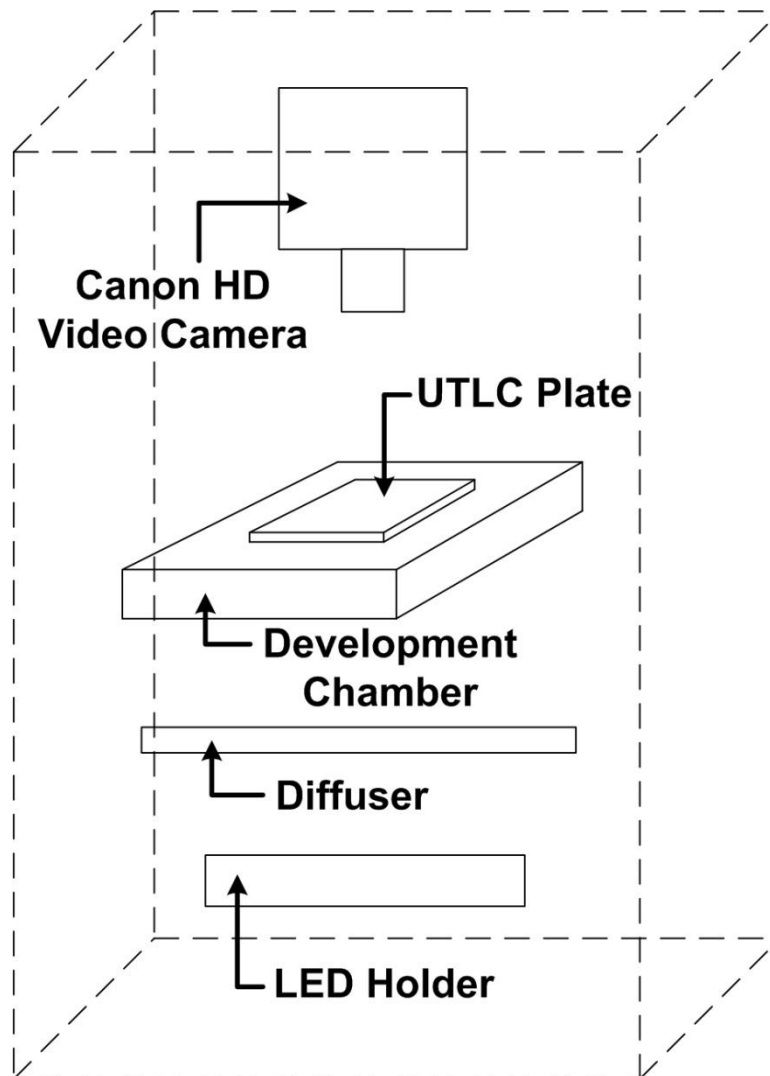


Figure5-3: Setup schematic of transparent mode TR-UTLC detection system.

An Arduino microcontroller (Arduino Uno Board Rev 3, P/N: 1050-1024-ND) in conjunction with corresponding LED drivers controlled the on/off times for each colored set of LEDs. Each colored group of LEDs were turned ‘on’ separately in sequence for 84 ms with an ‘off’ transition phase of 18 ms for a total period of 102 ms per color. This ensured that at least one out of every three frames (34 ms per frame) was fully saturated with a single color LED, and thus analysis on at least one of these frames could be completed. A timing schematic for the LEDs is shown in Figure 5-4.



Figure 5-4: Schematic of the timing diagram used in the transmission mode video UTLC system. Over each complete period, it is ensured that at least one camera frame fully captures each LED color.

5.3 Circuit Design

A schematic of the circuit design used to power the LEDs is shown in Figure 5-5. An Arduino microcontroller, powered by a 5 V DC power supply, controls the entire circuit. A simple method of turning the whole system on/off was completed using a switch (SW1), which was wired to one of the analog

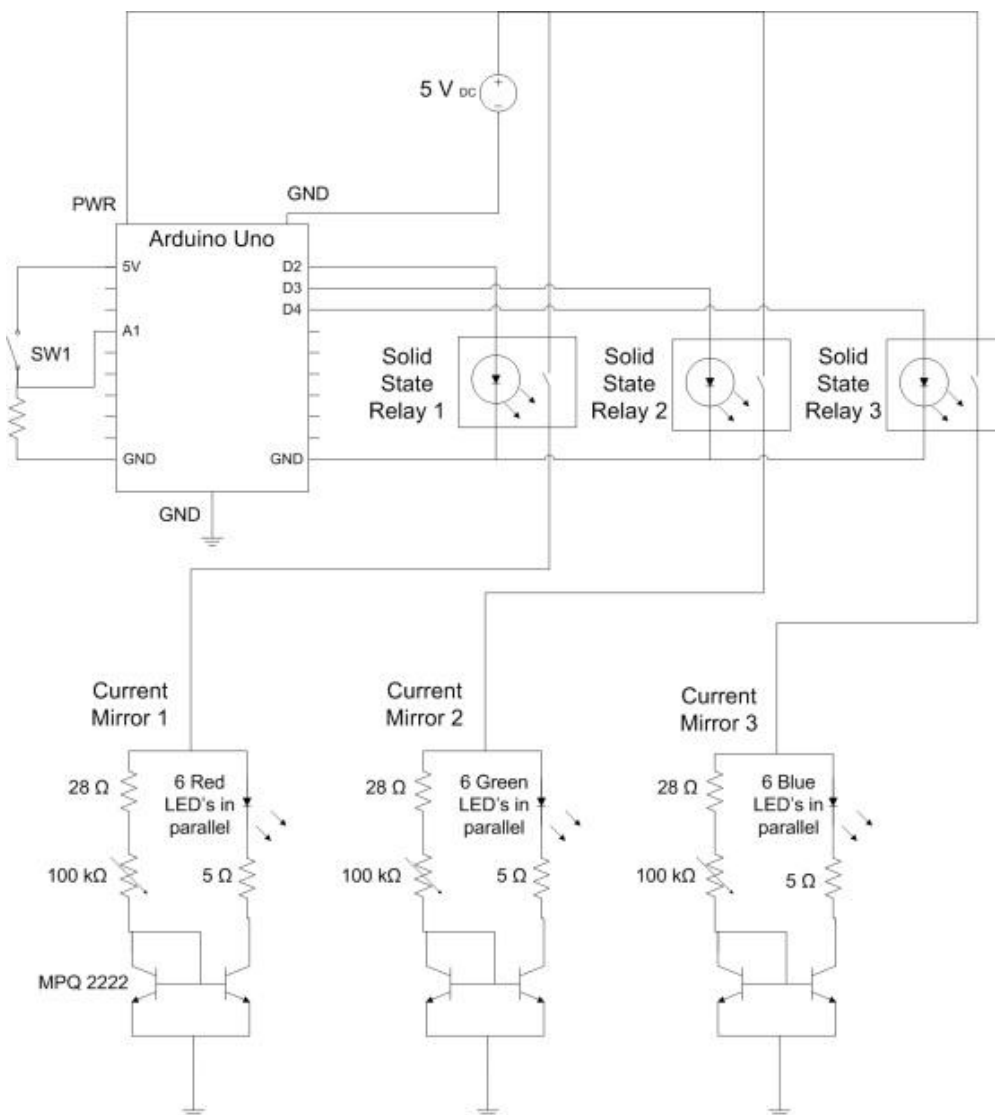


Figure 5-5: A schematic of LED drivers controlled by means of an Arduino microcontroller and solid state relays.

inputs (A1) of the microcontroller.

5.3.1 LED Drivers

A separate LED driver was employed for each colored LED group (every group contains 6 LEDs in parallel). All drivers consisted of a high speed solid state relay (SSR) (IC relay photovoltaic, 60 V, 1.5 A, 8-DIP, P/N: PVDZ172NPBF-ND) in series with a common emitter current mirror implementing npn bipolar junction transistors (MPQ2222 Quad Transistors, Central Semiconductor Corp.). The SSRs were controlled through the microcontroller's digital out board (D2 – D4). The 28 Ω and 5 Ω resistor values (both rated for 1 W) in each current mirror were chosen such that when the 100 k Ω potentiometers (trimmer 100K OHM 0.5W TH, P/N: 987-1102-ND) were shorted, an approximate current of 20 mA flowed, fully saturating each LED. The LEDs could also be linearly dimmed, such that a current as low as 0.5 mA was maintained (at which point the LEDs were only faintly visible), by increasing the potentiometer to its maximum resistance.

5.4 Measurement Principle

Two experiments were completed in this study. The first experiment involved the use of food dyes, which are examined more closely in this section. Both experiments implement the LEDs discussed above.

5.4.1 Food Dye Analysis

Several food dyes were analyzed with a spectrometer (Lambda 900, Perkin Elmer Inc., Massachusetts, USA) to determine their spectral response. As shown in Figure 5-6, and summarized in Table 5-1, all the dyes have unique absorption characteristics in the visible spectrum. The normalized peak absorption of each dye was approximately 50%, with the only exception being the blue at 60%. The orange dye had the largest wavelength range over which it is responsive and the green dye was the smallest with full width half maximums of 104 nm and 52 nm, respectively. The peak transmission wavelengths of all the dyes ranged from 481 nm for yellow up to 626 nm for green.

Table 5-1: Characteristics of four different colored food dyes

<i>Dye</i>	<i>Blue</i>	<i>Green</i>	<i>Orange</i>	<i>Yellow</i>
Full Width Half Maximum (nm)	95	52	104	87
Peak Wavelength (nm)	612	626	503	481
Normalized Peak Absorption (%)	60	50	50	30

5.4.2 LED Selection

Table 5-2 summarizes the properties of the three LEDs used in this study. The LEDs were chosen so as to be preferentially responsive as possible to each dye, given what was readily available. Figure 5-6 gives an excellent visible indication as to this preferential response to each dye for all LEDs. Only the blue

and green LEDs are absorbed by the yellow and orange dyes. The blue LED interacts slightly more with the yellow dye, whereas the green LED reacts more strongly with the orange dye. The red LED is the only one to be absorbed, albeit marginally, by the green dye. The blue dye is slightly absorbed by both the red and green LEDs. Unfortunately, due to the lack of a LED with peak wavelength in the 600 – 650 nm range, the green and blue dyes are not well absorbed by any of the selected LEDs. This suggests that for future work the LEDs may need to be chosen specifically for the materials being separated.

Table 5-2: Characteristics of three different LEDs used as light sources in the transparent mode video UTLC system

<i>LED</i>	<i>Blue</i>	<i>Green</i>	<i>Red</i>
Spectral Line Half-Width (nm)	45	35	20
Peak Emission Wavelength (nm)	470	525	660
Typical Forward Voltage (V)	3.5	3.5	1.8
Typical Forward Current (mA)	20	20	20
Luminous Intensity (mcd)	7000	6000	1000

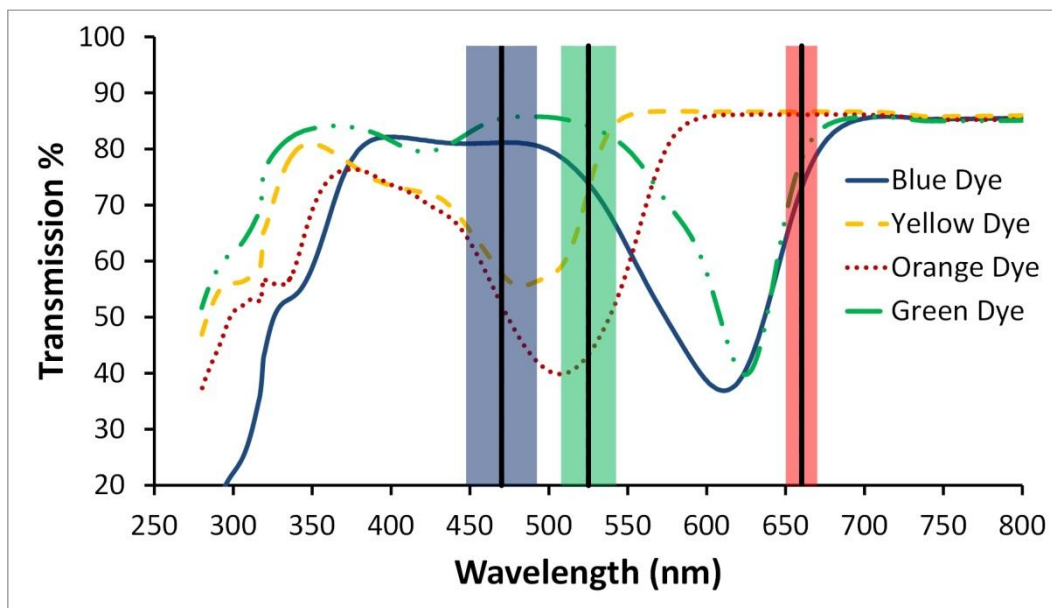


Figure 5-6: Transmission spectrum for four different colored food dyes. The peak wavelengths (blue – 470 nm, green – 525 nm, red – 660 nm) and spectral line half-widths of each LED are shown by the vertical lines and shaded color regions, respectively.

5.5 Results and Discussions: Experiment One

Two experiments were performed in order to validate the improvement this new detection method could offer.

5.5.1 Introduction

In the first experiment, a 5 μm vertical post GLAD UTLC plate was spotted with five different colored food dyes; blue (Indigo Carmine, 8 mg/mL in water), green (Green B, 46 mg/mL in water), orange (New Coccine, 5 mg/mL in water), yellow (Sunset Yellow, 7 mg/mL in water), and red (Chro fb, 100 mg/mL in water). After application of these spots, it was ensured that none of them overlapped. These plates were imaged using the TR-UTLC system, but not

developed. The goal of this experiment was to verify that detection of a fully separated spot could be successfully analyzed.

5.5.2 Chromatograms

The area occupied by all five spots was cropped and converted from RGB to HSV using MATLAB. Chromatograms of the cropped regions, shown in Figure 5-7, were generated by integrating the value (in the HSV color model) across the region's width. One of each fully saturated colored frames was subsequently analyzed. Rather than each peak lying above the baseline in the conventional manner, the chromatograms are flipped vertically. In this case, the value (lightness) is integrated across each track, corresponding to the transmission strength of each spot. However, detection of a spot with the current system setup relies solely on the absorption of the particular spot by the LED. Thus, the baseline for all the chromatograms lies above each spot's peak value.

5.5.3 Analyte Response to LEDs

As illustrated in Figure 5-7, each LED clearly shows a preferential transmission for dyes of similar wavelength and strong absorption for dyes of dissimilar wavelength, as expected from examining Figure 5-6. For example, the chromatogram obtained from the blue frame shows that the blue dye is only marginally absorbed, whereas both the orange and yellow dyes are strongly absorbed. An inspection of the red frame also conveys what is expected from analyzing Figure 5-6, with a larger absorption of the blue and green dyes than from that of the red and yellow dyes. This provided the necessary validation

required in order to undergo analysis of a post-developed UTLC plate using the transmission mode TR-UTLC system.

5.5.4 Spotting Deformities

Due to circular chromatography occurring during spotting (a process in which the analytes migrate in all directions due to no restrictions on the eluent flow direction), the chromatograms fail to resemble Gaussian curves. Rather, they tend to spike near the edge of each spot, where the majority of the dye has accumulated. Thus, the peak of each spike is likely a more accurate representation of the actual absorption of each dye. Taking this into account, an even larger discrepancy in the relative absorption of each spot is observed.

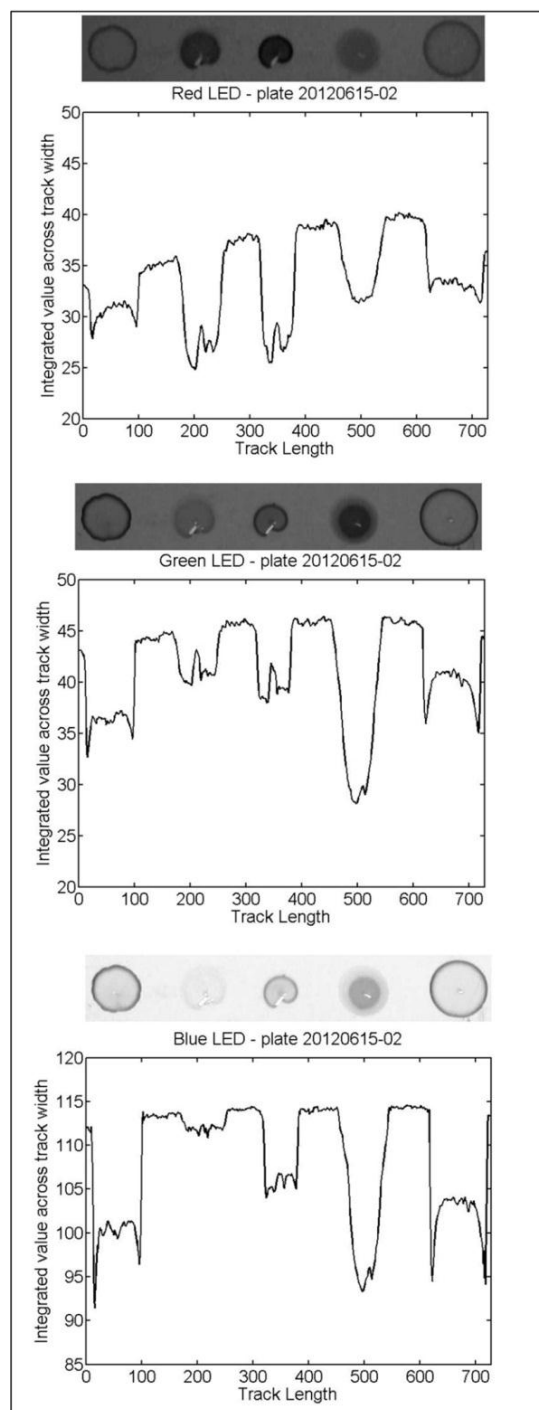


Figure 5-7: Lightness chromatograms of applied food dyes (from left to right: 5 mg/mL New Coccine, 8 mg/mL Indigo, 46 mg/mL Green B, 100 mg/mL Chro fb, 7 mg/mL Sunset Yellow) under various LED colors (from top to bottom: red, green, and blue).

5.6 Results and Discussions: Experiment Two

5.6.1 Introduction

In the second experiment, test dye mixture III was separated and examined on a Merck UTLC plate using toluene ($\geq 99.5\%$, Sigma-Aldrich Canada, Oakville, Canada) as the solvent. The goal of the second experiment was to show the improved ability of this updated TR-UTLC system to de-convolute closely spaced peaks. The six dyes comprising this mixture eluted in the following manner (from highest to lowest retention factor): dimethyl yellow, oracet red, sudan blue II, ariabel red, indophenol, and oracet violet. Similarly to what was observed in Chapter 4, the oracet red was too faint to be picked up visibly. In this case, the development was deliberately halted before all the spots could separate, so that the potential to analyze overlapping analytes (using this new detection approach) could be verified.

5.6.2 Background Correction

MATLAB was used as a tool to crop a track from the UTLC plate. A chromatogram of the plate's center track was established by integrating the value across the track's width. A rectangular region adjacent to each track, but not including any of the separated dye spots, was also cropped in order to perform a background correction. A background correction was necessary to reduce variations in lightness across the length of each track. This was achieved by subtracting the chromatogram of the background track from the chromatogram of the separation track.

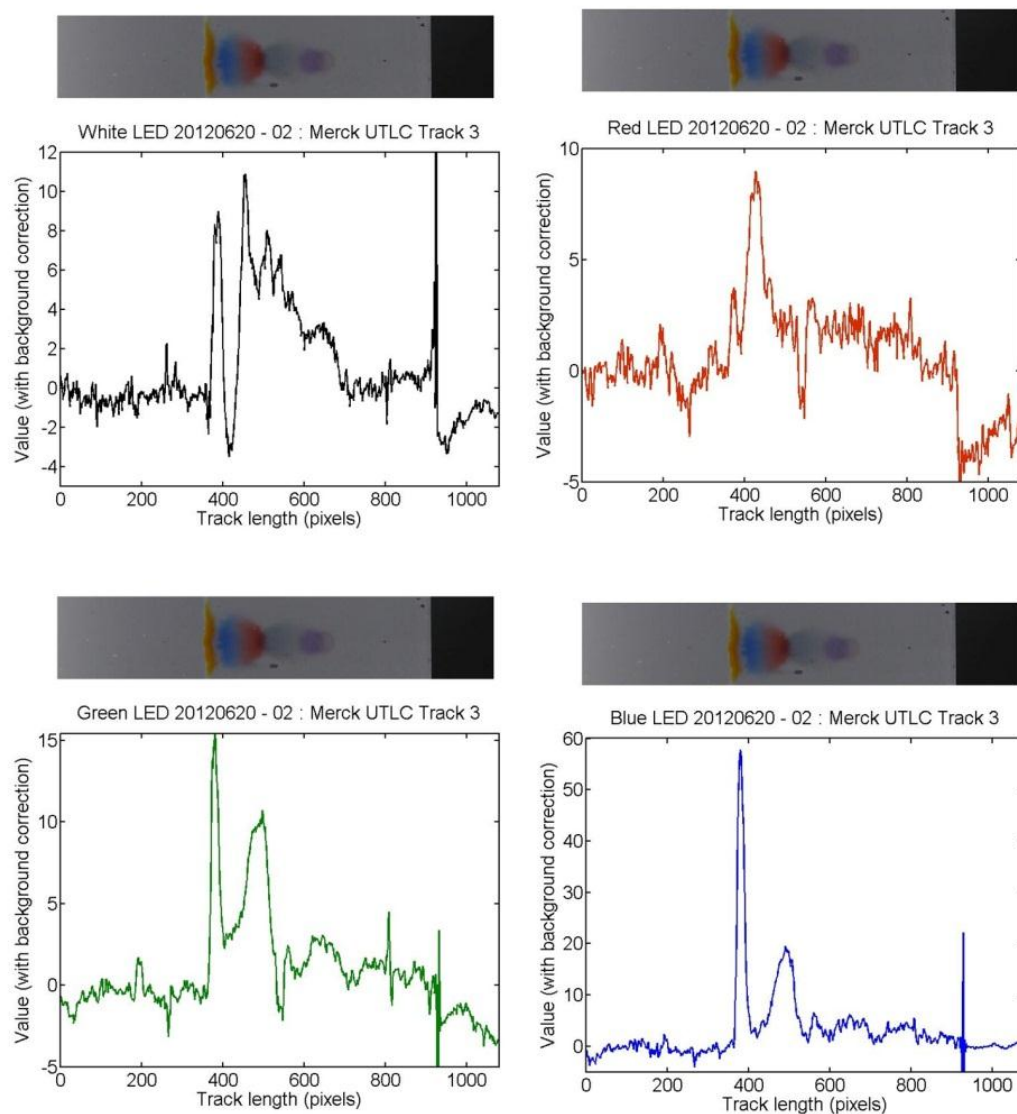


Figure 5-8: Lightness chromatograms of a single frame from a Merck UTLC plate separation of test dye mixture III under white (top left), red (top right), green (bottom left), and blue (bottom right) LED illumination.

5.6.3 Detection Capabilities

Besides the three colored LEDs, a white LED source (similar to that used in Chapter 4) was also analyzed separately to show the benefits offered by this

alternative system. Excluding the dimethyl yellow and oracet red, which quickly separated from the rest of the analytes due to their high retention factors, no well-defined peaks exist for any of the other analytes under the white LED illuminated frame (Figure 5-8a). Only the red frame (Figure 5-8b) was able to successfully isolate the sudan blue II peak. The suboptimal selection of the red LED for this particular dye set (which has low absorption for the majority of the dyes) gives rise to the large noise seen in this frame. The green illuminated frame (Figure 5-8c) shows a high selectivity for both the dimethyl yellow and the ariabel red, resulting in clear distinguishable peaks for both of these dye components. The wavelength range over which the green LEDs operated was sufficiently large as to be slightly absorbed by the other four dyes, however not enough to warrant definite peaks. The blue frame (Figure 5-8d) obtained the highest signal-to-noise ratio, and as expected was able to isolate peaks for both the dimethyl yellow and ariabel red dyes. Even with unoptimized LED selections, there appears to be clear improvements over white light detection. Moreover, there are cases in which dyes were only visible with the use of colored LEDs. It is expected that further system optimization will result in significant improvements in detection sensitivity.

5.7 Conclusion and Future Work

5.7.1 Source Synchronization

Due to time constraints, *in-situ* analysis of a separation was not performed. Syncing the colored LEDs “on” interval with the camera frames proved to be very challenging, but was achieved by adding an “off” interval between each LED. In

future work, it would be desirable to integrate the Arduino and the camera's hardware in order to have optimum sync control between the LEDs and the camera frame rate. For automation purposes, updated MATLAB scripts would also need to be written that amalgamated the information for all similar colored frames.

5.7.2 Enhanced Spectral Sensitivity

Although the transmission mode TR-UTLC system discussed in this chapter offers potential advantages over existing set-ups, many improvements are possible. Many of the dyes are not absorbed well, due to the currently limited wavelengths occupied by the LEDs. Ideally, the use of multiple high intensity LEDs covering the entire visible range, in combination with line filters (bandpass), could be used to obtain the highest conceivable detection sensitivity and potential LOD enhancement. This would ensure that a LED existed for each dye's peak absorption wavelength. However, the large number of LEDs required to perform this task would not only increase the system complexity but would also limit the time sensitivity. In other words, more colored frames would be necessary and thus, the complete cycling duration would increase by 34 ms per frame. Assuming the line filters reduced each LED to 10 nm spectral widths and that the LEDs could be timed perfectly with the camera, a minimum time sensitivity of approximately one second would be required to cover the entire visible range (400 – 700 nm).

5.7.3 Limits of Detection

Another weakness of the current approach is the spotting method. Rather than spotting by hand, a more precise spotting technique could be investigated that would support quantitative analysis. The availability of quantitative data would allow us to calculate LOD's, and thus determine which system approach (i.e. hue filters or multiple LED source) offers the most potential.

6 Conclusion

6.1 Summary of Thesis Results

Unlike TLC, where the stationary phase typically consists of suspended silica gel particles, UTLC plates contain a monolithic stationary phase. The TLC particle size can be controlled, thereby modifying the retention properties of the separation media. However, silica gel TLC plates cannot offer many of the advantages representative of GLAD UTLC. GLAD UTLC plates offer enhanced separation control through alterations of the stationary phase morphology, thickness, and material choice.

In particular, anisotropic GLAD UTLC containing a SBD nanostructured stationary phase (ie: anisotropic vertical blades) present the opportunity to provide more development tracks on a single plate. This is accomplished through induced channels in the separation media, thus reducing diffusion in the cross channel direction. Although GLAD UTLC plates extend the potential of planar chromatography, traditional development procedures are not as effective due to the more sensitive nature of the GLAD UTLC plates. Characterizing the anisotropic GLAD UTLC plates was made extremely difficult due to the special handling requirements such as removing the plate from the development chamber and air drying the developed plate.

In-situ measurements of the developing plates appeared to be the best method to obtain the most accurate separation data possible. We established the

time-resolved UTLC system to attain this objective. This system allowed us to bypass any destruction to the separation data by eliminating much of the human intervention traditionally required. A commercially available high-definition video camera mounted in a customized PVC enclosure recorded the entire development. Extracted video frames, at up to 30 frames per second, were then input into a custom MATLAB script and subsequently analyzed to produce time-resolved figures of merit.

While this new time-resolved UTLC system provided much more reliable data, there still remained challenges in obtaining accurate data during the initial stages in the development at which point analytes have not completely separated. A transmission mode video UTLC system, utilizing a transparent development chamber and LED source control, was utilized to help better de-convolve overlapping peaks. This proof-of-concept experiment appeared to provide more sensitive peak detection of low resolution peaks.

6.2 Ongoing Efforts

As expected in any scientific endeavor, much room for improvement is evident in the projects discussed in this thesis.

6.2.1 GLAD UTLC

Materials other than SiO_2 could potentially offer improved separation performance. GLAD UTLC plates implementing other metal oxide stationary phases may warrant unique separation capabilities. Post-processing of GLAD UTLC thin films, similar to the work carried out by Jim *et al*³⁸ may be used to enhance plate performance in the future.

6.2.2 Detection and Analysis

Combining more precise spotting techniques with the TR-UTLC system could enable separation quantification, which currently is an on-going challenge in UTLC. Detection of analytes which do not absorb in the visible using our current TR-UTLC system could be made possible by integrating laser fluorescence quenching⁷⁸. It would also be recommended for future users of the TR-UTLC system to decrease the run-time of the MATLAB scripts by using more efficient programming procedures, as the current script can take up to approximately one second to analyze each frame. Color models other than the HSV model used in this thesis may prove to be a better alternative and offer certain advantages over the hue filtered approach.

References

1. Sherma, J. Basic TLC Techniques, Materials, and Apparatus. *Handbook of Thin Layer Chromatography* (2003).
2. Berezkin, V. G. & Kormishkina, E. V. Study of a new version of classical thin-layer chromatography with a closed adsorbent layer. *JPC - Journal of Planar Chromatography - Modern TLC* **19**, 81–85 (2006).
3. Jim, S. R. *et al.* Engineered anisotropic microstructures for ultrathin-layer chromatography. *Analytical Chemistry* **82**, 5349–5356 (2010).
4. Bezuidenhout, L. W. & Brett, M. J. Ultrathin layer chromatography on nanostructured thin films. *Journal of Chromatography A* **1183**, 179–185 (2008).
5. Jim, S. R., Oko, a J., Taschuk, M. T. & Brett, M. J. Morphological modification of nanostructured ultrathin-layer chromatography stationary phases. *Journal of chromatography. A* **1218**, 7203–10 (2011).
6. Siouffi, A.-M. From Paper to Planar: 60 Years of Thin Layer Chromatography. *Separation & Purification Reviews* **34**, 155–180 (2005).
7. Ettre, L. S. Nomenclature for chromatography (IUPAC Recommendations 1993). *Pure and Applied Chemistry* **65**, 819–872 (1993).
8. Ettre, L. & Sakodynskii, K. M . S . Tswett and the Discovery of Chromatography h Early Work (1899-1903)*. *Chromatographia* **35**, 223–231 (1993).
9. Berezkin, V. G. & Dzido, T. H. The discovery of thin-layer chromatography by N.A. Izmailov and M.S. Shraiber. *JPC - Journal of Planar Chromatography - Modern TLC* **21**, 399–403 (2008).
10. Poole, C. F. Thin-layer chromatography: challenges and opportunities. *Journal of chromatography. A* **1000**, 963–84 (2003).
11. Indrayanto, G. *et al.* *High-Performance Thin-Layer Chromatography (HPTLC)*. *High-Performance Thin-Layer Chromatography (HPTLC)* (Springer Berlin Heidelberg: Berlin, Heidelberg, 2011).doi:10.1007/978-3-642-14025-9
12. Hahn-Deinstrop, E. *Applied Thin-Layer Chromatography*. *Applied Thin-Layer Chromatography* (Wiley-VCH, Weinheim: 2006).

13. CAMAG Basic Equipment for Modern Thin-Layer Chromatography. at <<http://www.mercury-ltd.co.il/admin/userfiles/CAMAG-basic-equipment-08.pdf>>
14. Hauck, H. E., Bund, O., Fischer, W. & Schulz, M. Ultra-thin layer chromatography (UTLC) — A new dimension in thin-layer chromatography. *JPC - Journal of Planar Chromatography - Modern TLC* **14**, 234–236 (2001).
15. Hauck, H. E. & Schulz, M. Ultrathin-layer chromatography. *Journal of chromatographic science* **40**, 550–2 (2002).
16. Hauck, H. E. & Schulz, M. Ultra thin-layer chromatography. *Chromatographia* **57**, S313–S315 (2003).
17. Poole, S. K. & Poole, C. F. High performance stationary phases for planar chromatography. *Journal of chromatography. A* **1218**, 2648–60 (2011).
18. Spangenberg, B., Poole, C. F. & Weins, C. *Quantitative Thin-Layer Chromatography. Media* (Springer Berlin Heidelberg: Berlin, Heidelberg, 2011).doi:10.1007/978-3-642-10729-0
19. Kowalska, T. Theory and Mechanism of Thin-Layer Chromatography. *Handbook of Thin Layer Chromatography* (2003).
20. van Deemter, J. J., Zuiderweg, F. J. & Klinkenberg, A. Longitudinal diffusion and resistance to mass transfer as causes of nonideality in chromatography. *Chemical Engineering Science* **5**, 271–289 (1956).
21. Hawkeye, M. M. & Brett, M. J. Glancing angle deposition: Fabrication, properties, and applications of micro- and nanostructured thin films. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* **25**, 1317–1335 (2007).
22. Robbie, K., Friedrich, L., Dew, S., Smy, T. & Brett, M. Fabrication of thin films with highly porous microstructures. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* **13**, 1032 (1995).
23. Steele, J. J. & Brett, M. J. Nanostructure engineering in porous columnar thin films: recent advances. *Journal of Materials Science: Materials in Electronics* **18**, 367–379 (2006).
24. Van Dijken, J. G., Wu, N., Fleischauer, M. D., Buriak, J. M. & Brett, M. J. Morphology control and nanoscale patterning of small molecule organic thin films. *SPIE* **8435**, 84350R–84350R–8 (2012).

25. Thomas, M. *et al.* C60 fullerene nanocolumns--polythiophene heterojunctions for inverted organic photovoltaic cells. *ACS applied materials & interfaces* **3**, 1887–94 (2011).
26. Rider, D. A. *et al.* Indium tin oxide nanopillar electrodes in polymer/fullerene solar cells. *Nanotechnology* **22**, 085706 (2011).
27. Krabbe, J. D., Leontyev, V., Taschuk, M. T., Kovalenko, A. & Brett, M. J. Square spiral photonic crystal with visible bandgap. *Journal of Applied Physics* **111**, 064314 (2012).
28. Hawkeye, M. M. & Brett, M. J. Optimized Colorimetric Photonic-Crystal Humidity Sensor Fabricated Using Glancing Angle Deposition. *Advanced Functional Materials* **21**, 3652–3658 (2011).
29. Summers, M. A., Tabunshchyk, K., Kovalenko, A. & Brett, M. J. Fabrication of 2D–3D photonic crystal heterostructures by glancing angle deposition. *Photonics and Nanostructures - Fundamentals and Applications* **7**, 76–84 (2009).
30. Smetaniuk, D. P., Taschuk, M. T. & Brett, M. J. Photocatalytic Titanium Dioxide Nanostructures for Self-Regenerating Relative Humidity Sensors. *IEEE Sensors Journal* **11**, 1713–1719 (2011).
31. Krause, K. M., van Popta, A., Steele, J. J., Sit, J. C. & Brett, M. J. Microstructured humidity sensors fabricated by glancing angle deposition: characterization and performance evaluation. *Proceedings of SPIE* **6800**, 68000Y–68000Y–11 (2007).
32. Hawkeye, M. M. & Brett, M. J. Narrow bandpass optical filters fabricated with one-dimensionally periodic inhomogeneous thin films. *Journal of Applied Physics* **100**, 044322 (2006).
33. Vick, D., Smy, T. & Brett, M. J. Growth behavior of evaporated porous thin films. *Journal of Materials Research* **17**, 2904–2911 (2011).
34. Hrudehy, P. C. P., Szeto, B. & Brett, M. J. Strong circular Bragg phenomena in self-ordered porous helical nanorod arrays of Alq₃. *Applied Physics Letters* **88**, 251106 (2006).
35. Robbie, K. *et al.* Ultrahigh vacuum glancing angle deposition system for thin films with controlled three-dimensional nanoscale structure. *Review of Scientific Instruments* **75**, 1089 (2004).

36. Bezuidenhout, L. W. & Brett, M. J. Ultrathin layer chromatography on nanostructured thin films. *Journal of Chromatography A* **1183**, 179–185 (2008).
37. Jim, S. R. *et al.* Engineered anisotropic microstructures for ultrathin-layer chromatography. *Analytical chemistry* **82**, 5349–56 (2010).
38. Jim, S. R., Oko, a J., Taschuk, M. T. & Brett, M. J. Morphological modification of nanostructured ultrathin-layer chromatography stationary phases. *Journal of chromatography. A* **1218**, 7203–10 (2011).
39. Clark, J. E. & Olesik, S. V. Technique for ultrathin layer chromatography using an electrospun, nanofibrous stationary phase. *Analytical Chemistry* **81**, 4121–4129 (2009).
40. Clark, J. E. & Olesik, S. V. Electrospun glassy carbon ultra-thin layer chromatography devices. *Journal of Chromatography A* **1217**, 4655–4662 (2010).
41. Broszat, M., Ernst, H. & Spangenberg, B. a Simple Method for Quantifying Triazine Herbicides Using Thin-Layer Chromatography and a Ccd Camera. *Journal of Liquid Chromatography & Related Technologies* **33**, 948–956 (2010).
42. Hess, A. V. I. Digitally Enhanced Thin-Layer Chromatography: An Inexpensive, New Technique for Qualitative and Quantitative Analysis. *Journal of Chemical Education* **84**, 842 (2007).
43. Jim, S. R. *et al.* Engineered anisotropic microstructures for ultrathin-layer chromatography. *Analytical Chemistry* **82**, 5349–5356 (2010).
44. Mustoe, S. P. & McCrossen, S. D. A comparison between slit densitometry and video densitometry for quantitation in thin layer chromatography. *Chromatographia* **53**, S474–S477 (2001).
45. Oko, A. J., Jim, S. R., Taschuk, M. T. & Brett, M. J. Analyte migration in anisotropic nanostructured ultrathin-layer chromatography media. *Journal of chromatography. A* **1218**, 2661–7 (2011).
46. Zhang, L. Background reconstruction for camera-based thin-layer chromatography using the concave. *Optics Express* **14**, 10386–10392 (2006).
47. Zhang, L. & Lin, X. Quantitative evaluation of thin-layer chromatography with image background estimation based on charge-coupled device imaging. *Journal of chromatography. A* **1109**, 273–8 (2006).

48. Berezkin, V. G. & Chausov, A. V. Quasi-continuous videodensitometric recording of chromatograms in circular TLC. *JPC - Journal of Planar Chromatography - Modern TLC* **24**, 188–195 (2011).
49. Lancaster, M., Goodall, D. M., Bergström, E. T., McCrossen, S. & Myers, P. Real-time image acquisition for absorbance detection and quantification in thin-layer chromatography. *Analytical chemistry* **78**, 905–11 (2006).
50. Taschuk, M. T., Hawkeye, M., Brett, M. J. & Martin, P. M. *Handbook of Deposition Technologies for Films and Coatings: Science, Applications and Technology*. 621 (2010).
51. Hawkeye, M. M. & Brett, M. J. Glancing angle deposition: Fabrication, properties, and applications of micro- and nanostructured thin films. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* **25**, 1317 (2007).
52. Robbie, K. Advanced techniques for glancing angle deposition. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures* **16**, 1115 (1998).
53. Morlock, G. E., Oellig, C., Bezuidenhout, L. W., Brett, M. J. & Schwack, W. Miniaturized planar chromatography using office peripherals. *Analytical chemistry* **82**, 2940–6 (2010).
54. Tait, R. N., Smy, T. & Brett, M. J. Modelling and characterization of columnar growth in evaporated films. *Thin Solid Films* **226**, 196–201 (1993).
55. Krause, K. M. *et al.* Surface area characterization of obliquely deposited metal oxide nanostructured thin films. *Langmuir: the ACS journal of surfaces and colloids* **26**, 4368–76 (2010).
56. Hodgkinson, I. & Wu, Q. hong Serial Bideposition of Anisotropic Thin Films with Enhanced Linear Birefringence. *Applied Optics* **38**, 3621 (1999).
57. Rasband, W. S. ImageJ. U. S. National Institutes of Health at <<http://rsb.info.nih.gov/ij>>
58. Python Reference Manual. (2012).at <<http://docs.python.org/ref/ref.html>>
59. Fan, J.-G., Dyer, D., Zhang, G. & Zhao, Y.-P. Nanocarpet Effect: Pattern Formation during the Wetting of Vertically Aligned Nanorod Arrays. *Nano Letters* **4**, 2133–2138 (2004).

60. Fan, J.-G. & Zhao, Y.-P. Spreading of a water droplet on a vertically aligned Si nanorod array surface. *Applied Physics Letters* **90**, 013102 (2007).
61. Fan, J.-G., Fu, J.-X., Collins, A. & Zhao, Y.-P. The effect of the shape of nanorod arrays on the nanocarpet effect. *Nanotechnology* **19**, 045713 (2008).
62. Vovk, I., Popović, G., Simonovska, B., Albrecht, A. & Agbaba, D. Ultra-thin-layer chromatography mass spectrometry and thin-layer chromatography mass spectrometry of single peptides of angiotensin-converting enzyme inhibitors. *Journal of chromatography. A* **1218**, 3089–94 (2011).
63. Sherma, J. Planar chromatography. *Analytical chemistry* **82**, 4895–910 (2010).
64. Morlock, G. E. & Schwack, W. The contribution of planar chromatography to food analysis. *JPC - Journal of Planar Chromatography - Modern TLC* **20**, 399–406 (2007).
65. Zhang, B., Bergström, E. T., Goodall, D. M. & Myers, P. Capillary action liquid chromatography. *Journal of separation science* **32**, 1831–7 (2009).
66. Lancaster, M., Goodall, D. M., Bergström, E. T., McCrossen, S. & Myers, P. Quantitative ultraviolet measurements on wetted thin-layer chromatography plates using a charge-coupled device camera. *Journal of chromatography. A* **1182**, 219–25 (2008).
67. Petrović, M., Kastelan-Macan, M., Ivanković, D. & Matečić, S. Video-densitometric quantitation of fluorescence quenching on totally irradiated thin-layer chromatographic plates. *Journal of AOAC International* **83**, 1457–62 (2000).
68. Vovk, I. & Prošek, M. Reproducibility of densitometric and image analysing quantitative evaluation of thin-layer chromatograms I. *Journal of Chromatography A* **779**, 329–336 (1997).
69. Vovk, I. & Prošek, M. Quantitative evaluation of chromatograms from totally illuminated thin-layer chromatographic plates. *Journal of Chromatography A* **768**, 329–333 (1997).
70. Petrović, M., Kastelan-Macan, M., Andrić, S. & Bokić, L. Application of a colour analyzer in quantitative thin-layer chromatography. *Journal of Chromatography A* **771**, 251–257 (1997).

71. Van der Auwera, G., David, P. T. & Reisslein, M. Traffic and Quality Characterization of Single-Layer Video Streams Encoded with the H.264/MPEG-4 Advanced Video Coding Standard and Scalable Video Coding Extension. *IEEE Transactions on Broadcasting* **54**, 698–718 (2008).
72. Komsta, L. Suppressing the charged coupled device noise in univariate thin-layer videoscans: a comparison of several algorithms. *Journal of chromatography. A* **1216**, 2548–53 (2009).
73. Komsta, L. Dealing with charged-coupled device noise in thin-layer videodensitometry. Optimization of several image-denoising techniques. *Acta Chromatographica* **21**, 355–367 (2009).
74. Di Marco, V. B. & Bombi, G. G. Mathematical functions for the representation of chromatographic peaks. *Journal of chromatography. A* **931**, 1–30 (2001).
75. Rutter, L. Some applications of a modified technique in paper chromatography. *The Analyst* **75**, 37 (1950).
76. Reich, E. & Schibli, A. *High-Performance Thin-Layer Chromatography for the Analysis of Medicinal Plants*. (Thieme Medical: 2007).
77. Halpaap, H. & Krebs, K.-F. Thin-layer chromatographic and high-performance thin-layer chromatographic ready-for-use preparations with concentrating zones. *Journal of Chromatography* **142**, 823–853 (1977).
78. Morlock, G. E. Detection, Identification, and Documentation. *Handbook of Thin Layer Chromatography* (2003).
79. Oko, A. J., Jim, S. R., Taschuk, M. T. & Brett, M. J. Time resolved chromatograms in ultra-thin layer chromatography. *Journal of chromatography. A* **1249**, 226–32 (2012).

Appendix A Analysis Code

This appendix includes scripts written for the TR-UTLC system discussed in Chapter 4 (completed using MATLAB) and also includes code for the LED timing discussed in Chapter 5 (completed using Arduino Microcontroller).

A.1 MATLAB Scripts

I would like to acknowledge Steven Jim for the many insights he provided that led to multiple code improvements and optimizations. The final code is divided into the following three objectives:

1. Define UTLC analysis parameters
2. Colour-filtered densitogram extraction
3. Figure of merit calculations

A.1.1 Define UTLC Analysis Parameters

This section involves 2 main functions and 2 sub-functions:

MAIN 1: `checkHueRangev2.m` : Used to define colour filters to isolate each coloured analyte spot.

MAIN 2: `createRecipeUTLCv4.m` : Used to specify and error check the required parameters for analysis.

SUB 1: `getChi4.m` : This script is designed to calculate the migration constant 'chi' (in units px^2/frame). It considers a region cropped from select frames in a given directory ('filepath'). The user identifies the migration front by eye. These measurements are used to fit the required 'chi' and 'zStart' scalar values.

SUB 2: `migrationfit.m` : Used to fit the front position to the formula $Z = \sqrt{\text{chi} \cdot \text{time}}$.

A.1.1.1 checkHueRangev2.m

```
function [spotLoc,path] = checkHueRangev2(BGpixelpos,Lfixed,spotLoc,path)

%% Load image file
% Initialize path to image if required
if exist('path', 'var') == 0 % Recently used path to standard *.csv
    path = "";
end

% Select reference image (*.jpg)
[imagename,path] = uigetfile('*.jpg', ...
    'Select reference image used to identify hue ranges (*.jpg)', ...
    path);
if imagename == 0
    error('Invalid reference image *.jpg file, Silly!');
end
refFrame0 = imread([path imagename]); % Load *.jpg file

%% Perform colour correction
% Invoke BGColourCorrect.m: [C, fit] = BGColourCorrect(A,pos,Lfixed,seefit)
% Don't show generated plots or export fitting parameters (fit).
disp(' ')
if strcmp('none',Lfixed)
    refFrame = refFrame0;
    disp('***NO COLOUR CORRECTION PERFORMED***')
else
    disp('***PERFORM COLOUR CORRECTION***')
    [C] = BGColourCorrect(refFrame0,BGpixelpos,Lfixed,'n');
    refFrame = uint8( double(refFrame0) - C );
end

%% Preview initial and colour-corrected reference frames
figure(101);
subplot(1,2,1); imshow(refFrame0);
    title('Selected reference frame with overlaid background zone');
    hold on; plot(BGpixelpos(:,1),BGpixelpos(:,2),'r.');
```

```
hold off;
subplot(1,2,2); imshow(refFrame);
    title(['Colour corrected reference frame: ' imagename '']);

% Check colour corrected frames
disp(' ')
goodcolour = input('Is colour correction good? Type "n" to stop: ');
if goodcolour == 'n'
    error('Why are you so picky, Silly!?!')
end
```

```

clear goodcolour;

%% How many tracks and spots should be considered?
disp(' ')
disp('***HOW MANY TRACKS AND SPOTS SHOULD BE
CONSIDERED?***')
if iscell(spotLoc)
    nTracks = size(spotLoc,1);
    nSpots = size(spotLoc,2);
    disp('Input "spotLoc" cell array is valid...')
    disp(['  nTracks = ' num2str(nTracks) ', nSpots = ' num2str(nSpots)])
else
    disp('Input "spotLoc" cell array does not exist or is not valid...')
    nTracks = input(['Number of tracks from which spots will ' ...
        'be isolated: ']);
    nSpots = input('Number of analyte spots from each track to analyze: ');
    spotLoc = cell(nTracks,nSpots);
end

% Initialize cell array. Rows correspond to track number (left to right);
% columns correspond to spot number (descending hRF); layers correspond to
% cropped image RGB array (3D), HSV array (3D), and array of H,S
% (hue,saturation) ordered pairs (2D).
spot = cell(nTracks,nSpots,3);

%% Crop spots and identify hue ranges
disp(' ')
disp('***ISOLATE AND ANALYZE SPOTS***')
disp('For each track (left-right), crop analyte spots in descending hRF')
figure(102);
imshow(refFrame);

for m = 1:nTracks
    disp(['Consider track ' num2str(m) ' of ' num2str(nTracks) '...'])
    for n = 1:nSpots
        disp(['  Crop analyte spot ' num2str(n) ' of ' num2str(nSpots)])
        % Use predefined spot locations or define new ones
        if isempty(spotLoc{m,n}) == 0
            spot{m,n,1} = imcrop(refFrame, spotLoc{m,n});
        else
            [spot{m,n,1} spotLoc{m,n}] = imcrop(refFrame);
        end
        % Dimensions of crop
        cropHeight = size(spot{m,n,1},1);
        cropWidth = size(spot{m,n,1},2);
        spot{m,n,2} = rgb2hsv(spot{m,n,1}); % Convert (RGB to HSV)
    end
end

```

```

    % Extract relevant H,S values and reshape into 2D array
    spot{m,n,3} = spot{m,n,2}(:,1:2);
    spot{m,n,3} = reshape(spot{m,n,3}, cropHeight*cropWidth, 2, 1);
    % Ensure H [0,360), S [0,1], V [0,1]
    spot{m,n,3}(:,1) = round( 360*spot{m,n,3}(:,1) );
    spot{m,n,3}(spot{m,n,3} == 360) = 0; % Adjust for H = 360
    clear cropHeight cropWidth;
end
end

%% Calculate saturation sums for each analyte spot on each track
disp(' ')
disp('***CALCULATE HUE-FILTERED SATURATION SUMS FOR EACH SPOT***')

% Cell array to hold calculated saturation sums. Each cell array column
% represents a different analyte spot. First cell array row contains 2D
% arrays (360 x nTracks) whose columns represent the saturation sums of a
% given analyte spot on different tracks. Second row contains similar
% arrays with sums scaled by the total in a given column (prevents
% domination by most intense spots)
satSum = cell(2,nSpots);

for n = 1:nSpots
    disp(['Consider spot ' num2str(n) ' of ' num2str(nSpots) '...'])
    for m = 1:nTracks
        disp([' Processing spot ' num2str(n) ' from track ' ...
            num2str(m) ' of ' num2str(nTracks)])
        H = spot{m,n,3}(:,1);
        S = spot{m,n,3}(:,2);
        % Hues h over [0,359]: Consider saturation
        for h = 1:360
            satSum{1,n}(h,m) = sum( S( H == (h-1) ) );
        end
        clear H S;
    end
end

% Scale values so that sum of scaled saturation sums for all hues = 1
disp(' Also calculating scaled saturation sums')
satSum{2,n} = satSum{1,n} ./ repmat(sum(satSum{1,n},1),360,1);

% Plot bar graphs for given analyte spot
disp(' Generating bar plots of saturation sums')
figure(n);
clear bar_nSpot;
ax(1) = subplot(2,2,1);

```

```

    bar([0:359]',satSum{ 1,n});
    title('Saturation sums vs. hue angle [SEPARATE TRACKS]');
    xlim([0 360]);
ax(2) = subplot(2,2,2);
    bar([0:359]',satSum{ 2,n});
    title('Scaled saturation sums vs. hue angle [SEPARATE TRACKS]');
    xlim([0 360]);
ax(3) = subplot(2,2,3);
    bar([0:359]',sum(satSum{ 1,n},2));
    title('Saturation sums vs. hue angle [SUM OVER ALL TRACKS]');
    xlim([0 360]);
ax(4) = subplot(2,2,4);
    bar([0:359]',sum(satSum{ 2,n},2));
    title(['Scaled saturation sums vs. hue angle ' ...
          '[SUM OVER ALL TRACKS]']);
    xlim([0 360]);
    linkaxes(ax,'x');
end
disp(' ')
disp('***COMPLETED CHECKING HUE RANGE***')
end

```

A.1.1.2 createRecipeUTLCv3

```

% Select old recipefile
disp(' ')
disp('PART 0: FILE DIRECTORIES')
oldRecipe0 = input('Load old recipe file (y/n): ','s');
if strcmp(oldRecipe0,'y')
    [oldRecipe0,recipePath0] = uigetfile('*.mat', ...
        'Select old recipe file');
    recipePath0 = [recipePath0 '\'];
    load([recipePath0 oldRecipe0]);
    oldRecipe = oldRecipe0; recipePath = recipePath0;
    clear oldRecipe0 recipePath0;
    disp(['Loaded recipe file "' oldRecipe "''])
else
    oldRecipe = ''; recipePath = '';
    disp('Creating entirely new recipe file.')
end

% Specify output recipefile name
disp('Define location and name of new recipe file.')
recipePath = uigetdir(recipePath,['Select directory in which to '...

```

```

    'place new recipe file.'];
recipePath = [recipePath '\'];
newRecipe = input('Enter output recipe file name (name.mat): ','s');

% Initialize output recipe file (verify that save is permitted)
save([recipePath newRecipe]);

% Set directory of UTLC video frames to analyze:
disp(' ')
disp('Choose directory of frame *.jpg extracted from video')
if ~exist('filepath','var')
    filepath = ''; % Initialize if it doesn't already exist
end
filepath = uigetdir(filepath,... % Assume location to be same as recipe
    'Choose directory of frame *.jpg extracted from video');
filepath = [filepath '\'];

% Initialize relevant filename, image arrays, etc.
filename = dir([filepath '*.jpg']);
nFrames = length(filename);
firstFrame = imread([filepath filename(1).name]);
lastFrame = imread([filepath filename(nFrames).name]);
middleFrame = imread([filepath filename(round(nFrames/2)).name]);
disp(['Selected directory contains ' num2str(nFrames) ' frames.'])
disp(['The first frame is: ' filename(1).name ''])
disp(['The last frame is: ' filename(nFrames).name ''])

%% Input information about video and calculate resolution
disp(' ')
disp('PART 1: VIDEO FRAME SET INFORMATION')
clf; imshow(lastFrame); axis on;
confirm = 'n';
while ~strcmp(confirm,'y')
    disp(' ')
    if ~exist('frameRate','var')
        % Input frame rate
        frameRate = input('Enter frame extraction rate (fps): ');
    end
    if ~exist('frameRes','var')
        % Calculate frame image resolution
        disp('Specify 4 corners of rectangular reference region used to')
        disp('calculate frame image resolution [px/mm]. CW from top left.')
        refrect = zeros(4,2);
        [refrect(:,1),refrect(:,2)] = ginput(4); % 4 corners of rect.
        disp('Input real dimensions [mm] of rectangle as row vector')
        refrectsize = input(['width height: ']);
    end
end

```



```

    refract2 = zeros(4,2); % refract with rows rearranged
    refract2(1:3,:) = refract(2:4,:); refract2(4,:) = refract(1,:);
    refract2 = refract - refract2;
    refractlengths = [norm(refract2(1,:)), norm(refract2(2,:)), ...
        norm(refract2(3,:)), norm(refract2(4,:))];
    % Calculate resolution by dividing circumferences: px/mm.
    frameRes = sum(refractlengths(1:4)) / ( 2*sum(refractsize) );
    refractlengths(5) = norm(refract(1,:) - refract(3,:));
    refractlengths(6) = norm(refract(2,:) - refract(4,:));
    clear refract2;
end
disp(['Current frame extraction rate: ' num2str(frameRate) ' fps'])
disp(['Reference rectangle of real width x height = ' ...
    num2str(refractsize(1)) ' x ' num2str(refractsize(2)) ' [mm]'])
disp('has measured pixel dimensions of top, right, bottom, left')
disp('edges and TL-BR, TR-BL diagonals of:')
disp(refractlengths)
disp(['Current frame image resolution: ' num2str(frameRes) ' [px/mm]'])
confirm = input('Confirm these values (y/n): ','s');
if ~strcmp(confirm,'y') % Clear and redefine
    clear frameRate frameRes refractsize refractlengths;
end
end
save([recipePath newRecipe]); % Save changes

%% Input number of possible tracks and crop
disp(' ')
disp('PART 2: SPECIFY NUMBER AND LOCATION OF TRACKS')
clf; imshow(lastFrame); axis on;
confirm = 'n';
while ~strcmp(confirm,'y')
    disp(' ')
    if ~exist('nTracks','var') || ~exist('defineTrack','var')
        nTracks = input('Enter number of tracks that can be analyzed: ');
        defineTrack = zeros(nTracks,4);
        disp('Define track locations (height of crop will be forced to)')
        disp('the full frame height')
        for trackNum=1:nTracks
            disp([' Crop location of track ' num2str(trackNum) ...
                ' of ' num2str(nTracks)])
            clf;
            [dummy,defineTrack(trackNum,:)] = imcrop(lastFrame);
            clear dummy
        end
        % Ensure defineTrack selects full height of frame
        defineTrack = round(defineTrack);
    end
end

```

```

        defineTrack(:,2) = 0; defineTrack(:,4) = size(lastFrame,1);
    end
    % Correct imported defineTrack array if required
    defineTrackTEST = round(defineTrack);
    defineTrackTEST(:,2) = 0; defineTrack(:,4) = size(lastFrame,1);
    defineTrackTEST = sortrows(defineTrackTEST,1);
    if max(max( defineTrack ~= defineTrackTEST ))
        defineTrack = defineTrackTEST;
        disp("'defineTrack" array updated so that all entries are')
        disp('integers, all tracks span full height of frame, and ')
        disp('all tracks are sorted left to right.')
    end
    clear defineTrackTEST;
    % Preview track locations
    disp('Current track locations (row contains x,y,width,height in px):')
    disp(defineTrack);
    clf; imshow(lastFrame); axis on;
    for i = 1:nTracks % Indicate location of each track
        % TOP left corner coordinates and dimensions (TOP because of
        % inverted image y-axis
        rectangle('Position', defineTrack(i,:));
    end
    confirm = input('Confirm these track locations (y/n): ','s');
    if ~strcmp(confirm,'y') % Clear redefine
        clear nTracks defineTrack;
    end
end
save([recipePath newRecipe]); % Save changes

%% Get initial spot location for each track from first frame
disp(' ')
disp('PART 3: INITIAL SPOT LOCATIONS')
confirm = 'n';
while ~strcmp(confirm,'y')
    disp(' ')
    % Preview track locations
    clf; imshow(firstFrame); axis on; hold on;
    for i = 1:nTracks % Indicate location of each track
        % TOP left corner coordinates and dimensions (TOP because of
        % inverted image y-axis
        rectangle('Position', defineTrack(i,:));
    end
    % Input spot locations if required
    if ~exist('spotLocationX','var') || ~exist('spotLocationY','var')
        disp('Locate each initial spot position (left to right)')
        [spotLocationX,spotLocationY] = ginput(nTracks);
    end
end

```

```

end
% Correct imported location vectors if required
spotLocationTEST1 = [spotLocationX,spotLocationY];
spotLocationTEST2 = sortrows(spotLocationTEST1,1);
if max(max( spotLocationTEST1 ~= spotLocationTEST2 ))
    spotLocationX = spotLocationTEST2(:,1);
    spotLocationY = spotLocationTEST2(:,2);
    disp('"spotLocationX" and "spotLocationY" vectors updated so ')
    disp('that all spot locations are sorted left to right.')
end
clear spotLocationTEST1 spotLocationTEST2;
% Preview spot positions and track locations
disp('Initial spot x,y positions (left to right) defined as [px]:')
disp([spotLocationX spotLocationY])
for i = 1:nTracks % Add locations of each spot to figure
    plot(spotLocationX(i),spotLocationY(i),'or');
end
hold off;
confirm = input('Confirm these values (y/n): ','s');
if ~strcmp(confirm,'y') % Clear redefine
    clear spotLocationX spotLocationY;
end
end
save([recipePath newRecipe]); % Save changes

%% Input Labels and Hue Ranges for each analyte
disp(' ')
disp('PART 4: ANALYTE SPOT LABELS AND HUE RANGES')
confirm = 'n';
while ~strcmp(confirm,'y')
    disp(' ')
    % Define hueRange array
    if ~exist('hueRange','var') || (min(hueRange(:)) < 0) || ...
        (max(hueRange(:)) > 360)
        nSpots = input('Enter number of analytes to analyze: ');
        hueRange = zeros(nSpots,2); % Initialize hueRange array
        disp('Enter analyte hue ranges in decreasing hRF as row vectors')
        for i = 1:nSpots
            hueRange(i,:) = input([' Enter hue range for ' ...
                'analyte ' num2str(i) ': ']);
        end
    end
end
hueRange(hueRange == 360) = 0; % Check for 360 == 0 discontinuity
if ~exist('hueRangeLabel','var') || (size(hueRangeLabel,1) ~= nSpots)
    % Display and label selected hueRange
    disp('Label analyte hue ranges')

```

```

hueRangeLabel = cell(nSpots,1);
for i = 1:nSpots
    templabel = input([' Name analyte ' num2str(i) ...
        ': H = [' num2str(hueRange(i,1)) ':' ...
        num2str(hueRange(i,2)) ']: ', 's');
    hueRangeLabel{i} = [templabel ': H = [' ...
        num2str(hueRange(i,1)) ':' num2str(hueRange(i,2)) ']];
end
end
% Display current hueRange and hueRangeLabel and confirm
disp('Current analyte hue ranges (labels) in descending hRF are:')
disp(hueRangeLabel)
confirm = input('Confirm these values (y/n): ', 's');
if ~strcmp(confirm, 'y') % Clear hueRange and/or hueRangeLabel
    redohueRange = input(' Redefine "hueRange" (y/n)? ', 's');
    redohueRangeLabel = ...
        input(' Redefine "hueRangeLabel" (y/n)? ', 's');
    if ~strcmp(redohueRange, 'n')
        clear hueRange;
    end
    if ~strcmp(redohueRangeLabel, 'n')
        clear hueRangeLabel;
    end
    if strcmp(redohueRange, 'n') && strcmp(redohueRangeLabel, 'n')
        disp(' No values changed. Current values confirmed.')
        confirm = 'y';
    end
end
clear templabel redohueRange redohueRangeLabel;
end
save([recipePath newRecipe]); % Save changes

%% Information about spot loading
disp(' ')
disp('PART 5: SPOT LOADING INFORMATION')

% Undiluted dye concentration
disp(' ')
disp('PART 5A: Undiluted dye concentration')
disp([num2str(nSpots) ' analyte hue filters are defined.'])
goPart = input('Proceed with this part (y/n)? ', 's');
confirm = 'n';
while ~strcmp(confirm, 'y') && strcmp(goPart, 'y')
    disp(' ')
    if ~exist('undilutedConc', 'var')
        disp('Specify the concentraton of each analyte for which a hue')
    end
end

```

```

disp('filter was defined. Define as a row vector of form')
undilutedConc = input('[conc1 conc2 conc3...] in units [g/L]: ');
undilutedConc = undilutedConc'; % Convert to column vector
else
disp('Current undiluted analyte conc. [g/L] is (descending hRF):')
disp(undilutedConc)
end
if length(undilutedConc) == nSpots
confirm = input('Confirm these values (y/n): ','s');
end
if ~strcmp(confirm,'y') % Clear hueRange and redefine
clear undilutedConc;
end
end
clear goPart;
save([recipePath newRecipe]); % Save changes

% Applied spot mass loading
disp(' ')
disp('PART 5B: Applied spot mass loading')
disp([num2str(nTracks) ' separation tracks/ initial spots were defined.'])
goPart = input('Proceed with this part (y/n)?: ','s');
confirm = 'n';
while ~strcmp(confirm,'y') && strcmp(goPart,'y')
disp(' ')
if ( ~exist('relConc','var') || ~exist('spotVolume','var') || ...
~exist('spotMass','var') )
relConc = input(['Specify relative sample concentration ' ...
'([0,1]; ex. 0.2 for 20%): ']);
% Applied spot volumes (row vector: 1 x nTracks)
disp('Specify applied spot volumes [nL] for each track')
spotVolume = input('Enter row vector: ');
else
disp(['The relative sample concentration is: ' num2str(relConc)])
disp('The applied spot volumes are (left to right):')
disp(spotVolume)
end
spotMass = relConc .* (undilutedConc * spotVolume);
disp('Calculated mass [ng] of each analyte (rows) in spot applied ')
disp('to each track (columns):')
disp(spotMass)
confirm = input('Confirm these values (y/n): ','s');
if ~strcmp(confirm,'y') % Clear hueRange and redefine
clear relConc spotVolume spotMass;
end
end
end

```

```

clear goPart;
save([recipePath newRecipe]); % Save changes

% Applied spot dimension measurements
disp(' ')
disp('PART 5C: Applied spot dimension measurements (with fitted ellipses)')
appliedSpotCSVpath = filepath; % Initialize location of *.csv files
confirm = 'n';
while ~strcmp(confirm,'y')
    disp(' ')
    loadSpotData = ''; % Initialize variable
    % Test for existing data
    if ( ~exist('spotdim_px','var') || ~exist('spotdim_mm','var') || ...
        ~exist('spotdim_res','var') )
        % Prompt for data entry method
        disp('No applied spot dimension data exists. Load data in *.csv')
        loadSpotData = ...
            input('(y), enter manually (n), or skip this part (s): ','s');
        spotdim_px = zeros(3,nTracks); % Initialize array for spot data
    end

    if strcmp(loadSpotData,'s') % Skip Part 5C
        deletePart5C = ...
            input('Clear any existing variables in PART 5C (y/n): ','s');
        if strcmp(deletePart5C,'y')
            clear appliedSpotCSV* spotdim* deletePart5C;
        end
        break; % Exit while loop
    end

    if strcmp(loadSpotData,'n') % Manually enter data.
        % Source image resolution (ex. 4800 dpi)
        disp('Enter linear resolution of image from which spot area data')
        spotdim_res = input('was obtained [dpi = dots per inch]: ');

        disp('Enter applied spot area data. Spots indicated by rows;')
        disp('Columns for: area, majoraxis, minoraxis')
        disp('input data for each spot (left-right) as 1 x 3 row vector')
        for i = 1:nTracks
            spotdim_px(:,i) = (input([' Applied spot ' ...
                num2str(i) ': ' ])); % Transpose row into column vector
        end
    end

    if strcmp(loadSpotData,'y') % Import data from *.csv
        % Source image resolution (ex. 4800 dpi)

```

```

disp('Enter linear resolution of image from which spot area data')
spotdim_res = input('was obtained [dpi = dots per inch]: ');

disp(['Load measured applied spot areas and fitted ellipse ' ...
    'parameters *.csv'])
[appliedSpotCSVname,appliedSpotCSVpath] = uigetfile('*.csv', ...
    'Select applied spot area information *.csv file', ...
    appliedSpotCSVpath);
appliedSpotCSVpath = [appliedSpotCSVpath '\'];
appliedSpotCSV = csvread([appliedSpotCSVpath ...
    appliedSpotCSVname],1,0); % Ignore the header row (Row 0)
% Each row represents one spot/track; 15 columns for: 1 spot
% number, 2 area (px^2); 3 mean, 4 min, 5 max grey values [0,255];
% centroid 6 x, 7 y position; fitted ellipse 8 major, 9 minor axis,
% 10 angle of major axis wrt +x direction, 11 circularity, 12
% %area, 13 aspect ratio, 14 roundness, 15 solidity.

% Include x-pos (csv column 6) to ensure left-right track sorting
spotdim_pxTEMP = appliedSpotCSV(:,[6 2 8 9]);
spotdim_pxTEMP = sortrows(spotdim_pxTEMP,1);
spotdim_pxTEMP(:,1) = []; % Delete x-pos col to give 3-columns
spotdim_px = spotdim_pxTEMP'; % Transpose into desired 3 x 8 form
clear spotdim_pxTEMP;
end

if size(spotdim_px,2) == nTracks % Verify enough spot area data
    % Current data
    disp('Spot data was extracted from an image of resolution:')
    disp([num2str(spotdim_res) ' [dpi] = ' ...
        num2str(spotdim_res/25.4) ' [px/mm]'])
    disp(' ')
    disp('Current loaded spot area data is (transposed --> rows are')
    disp('area [px^2]; fitted major axis [px], minor axis [px]):')
    disp(spotdim_px') % Transpose for display purposes
    % Calculated and confirm data in mm, mm^2
    spotdim_mm(1,:) = (25.4/spotdim_res)^2 * spotdim_px(1,:);
    spotdim_mm(2:3,:) = (25.4/spotdim_res) * spotdim_px(2:3,:);
    disp('area [mm^2]; fitted major axis [mm], minor axis [mm]):')
    disp(spotdim_mm') % Transpose for display purposes
    confirm = input('Confirm these values (y/n): ','s');
else
    disp(['"spotdim_px" array has more/less entries than the ' ...
        num2str(nTracks) ' tracks defined (transposed array):'])
    disp(spotdim_px') % Transpose for display purposes
end

```

```

if ~strcmp(confirm,'y') % Clear variables and redefined
    clear spotdim_px spotdim_mm spotdim_res appliedSpotCSV;
    clear appliedSpotCSVname; % Retain appliedSpotCSVpath if defined
end
clear loadSpotData;
end
save([recipePath newRecipe]); % Save changes

%% Get mobile front flow constant (chi)
disp(' ')
disp('PART 6: MOBILE PHASE FRONT FLOW CONSTANT
CALCULATION')
confirm = 'n'; close all;
while ~strcmp(confirm,'y')
    disp(' ')
    measurechi = ''; % Initialize
    if ~exist('chi','var') || ~ exist('zStart','var')
        % Specify a global chi value or individual values for each track
        disp('No valid "chi" and/or "zStart" values present')
        disp('Measure a separate "chi" value for each track (s),')
        disp('Measure a single global "chi" value for all tracks (g),')
        measurechi = input(['or define manually as a 1 x nTracks row ' ...
            'vector (d): '],'s');
        chi = zeros(1,nTracks); zStart = zeros(1,nTracks); % Initialize
    end
    if strcmp(measurechi,'s') % Measure separate chi values
        disp(' ')
        for i = 1:nTracks
            disp(['Measure "chi"/"zStart" value on track ' ...
                num2str(i) ' of ' num2str(nTracks)])
            % Define ROI as a region centred on the given track but
            % 5 times wider. (If invalid, it will be redfined by getchi4.m)
            chitrkloc = defineTrack(i,:);
            chitrkloc(1) = chitrkloc(1) - 2*chitrkloc(3);
            chitrkloc(3) = 5*chitrkloc(3);
            goodchifit = 0; % Initialize
            while ~goodchifit
                [chi(i),zStart(i)] = getChi4(filepath,filename,...
                    chitrkloc,10);
                % Error-check fitted values and itterate if necessary
                goodchifit = isreal(chi(i)) && (chi(i) > 0) && ...
                    isreal(zStart(i)) && (zStart(i) > 0);
            end
            if ~goodchifit
                disp([' Bad measured "chi"/"zStart" ' ...
                    'values returned:'])
                disp([' Chi = ' num2str(chi(i)) ...

```



```

        ' [px^2/frame], zStart = ' num2str(zStart(i)) ...
        ' [px].')]
    acceptbadchi = input(['  Accept (a), ' ...
        'invalidate as "NaN" (nan), or remeasure '...
        '(r)? : ', 's']);
    if strcmp(acceptbadchi, 'a')
        goodchifit = 1;
    elseif strcmp(acceptbadchi, 'nan')
        chi(i) = NaN; zStart(i) = NaN;
        goodchifit = 1;
    end
end
end
clear chitrkloc goodchifit acceptbadchi;
end
end
if strcmp(measurechi, 'g') % Measure a single chi and make global
    disp('Measure a "chi"/"zStart" value and apply to all tracks.')
    % Keeping argument 4 empty forces getChi4 to manually define ROI.
    [chi(1), zStart(1)] = getChi4(filepath, filename, '', 10);
    chi(:) = chi(1); zStart(:) = zStart(1);
end
if strcmp(measurechi, 'd') % Manually enter values
    disp(['Define "chi" and "zStart" as 1 x ' num2str(nTracks) ...
        ' row vectors'])
    chi = input('Enter "chi" vector: ');
    zStart = input('Enter "zStart" vector: ');
end
% Display current values
disp('Current "chi" and "zStart" (transposed row vectors):')
disp('"'chi" vector (left) is [px^2/frames] (one entry per track):')
disp('"'zStart" vector (right) is [px] (one entry per track):')
disp([chi' zStart'])
% Ensure values are valid and confirm
if (length(chi) == nTracks) && (length(zStart) == nTracks)
    confirm = input('Confirm these values (y/n): ', 's');
else
    disp(['Data is insufficient for ' num2str(nTracks) ' tracks.'])
    disp(' ')
end
if ~strcmp(confirm, 'y') % Clear and redefine
    clear chi zStart;
end
clear measurechi;
end

```

```

% Calculate absolute front position (wrt top of frame)
% nFrames x nTracks array of front positions
zf0 = sqrt( (1:nFrames)' * chi ) + repmat(zStart,nFrames,1);
% Calculate migration front distance wrt applied spot location
zf = zf0 - repmat(spotLocationY',nFrames,1);
disp('Calculated migration front positions for all tracks in all frames.')

save([recipePath newRecipe]); % Save changes

%% Initial guesses and fitting limits for peaks (formerly getHRf.m)
disp(' ')
disp('PART 7: INITIAL GUESSES/WINDOWS OF ANALYTE RF VALUES')
confirm = 'n'; close all;
while ~strcmp(confirm,'y')
    disp(' ')
    measureRF = ''; % Initialize
    if ~exist('rfGuess','var')
        % Specify a global chi value or individual values for each track
        disp('No valid "rfGuess" values present.')
        disp('Measure separate "rfGuess" values for each track (s),')
        disp('Measure global "rfGuess" values for all tracks (g),')
        measureRF = input(['or define manually as a 1 x nSpots ' ...
            'row vector (d): ','s']);
        % Initialize values
        rfGuess = zeros(nSpots,nTracks); % Guesses on RF values
        disp(' ')
    end
    if strcmp(measureRF,'s') % Measure separate RF values
        disp('Click on the (approximate) positions of each spot on each')
        disp('track. Identify spots on tracks from left to right in')
        disp('descending RF value. If unable to identify, click well')
        disp('above initial applied spot location to skip.')
        for i = 1:nTracks
            % Show last frame with overlaid track and applied spot
            clf; imshow(lastFrame); axis on; hold on;
            rectangle('Position', defineTrack(i,:));
            plot(spotLocationX(i),spotLocationY(i),'or'); hold off;
            disp(['Measure "rfGuess" values on track ' ...
                num2str(i) ' of ' num2str(nTracks)])
            [dummy,rfGuessTEMP] = ginput(nSpots);
            clear dummy
            % Negate spots with locations that cannot be identified
            % (selection is < 80% of the spotLocationY value.
            rfGuessTEMP(rfGuessTEMP < 0.8*spotLocationY(i)) = NaN;
            rfGuess(:,i) = (rfGuessTEMP - spotLocationY(i)) / ...
                zf(nFrames,i); % Estimate RF values for last frame
        end
    end
    confirm = input('Continue? (y/n): ');
end

```

```

end
% Assume that an analyte spot that could not be identified has a
% similar RF value to other spots of the same analyte
for j = 1:nSpots
    rfGuessTEMP = rfGuess(j,:);
    if (max(isnan(rfGuessTEMP)) == 1) && ...
        (min(isnan(rfGuessTEMP)) == 0)
        rfGuessTEMP(isnan(rfGuessTEMP)) = ...
            mean(rfGuessTEMP(~isnan(rfGuessTEMP)));
        disp(['Adjusted "unknown" values for spot ' num2str(j) ...
            ' of ' num2str(nSpots)])
    elseif (max(isnan(rfGuessTEMP)) == 1) && ...
        (min(isnan(rfGuessTEMP)) == 1)
        disp(['No valid "rfGuess" value for spot' num2str(j)])
        rfGuessTEMP(1,:) = input(['on any tracks! Manually ' ...
            'define for all tracks [0,1]: ']);
    end
    rfGuess(j,:) = rfGuessTEMP;
end
end
if strcmp(measureRF,'g') % Measure separate RF values
    clf; imshow(lastFrame); axis on; hold on;
    disp('On which track should RF guess be based? Pick a track')
    picktrk = input(['number from 1-' num2str(nTracks) ': ']);
    % Preview track and applied spot location
    rectangle('Position', defineTrack(picktrk,:));
    plot(spotLocationX(picktrk),spotLocationY(picktrk),'or'); hold off;
    % Select spot locations
    disp('Click on the (approximate) positions of each spot on ')
    disp('this track. Identify spots in descending RF value.')
    [dummy,rfGuessTEMP] = ginput(nSpots);
    clear dummy
    rfGuessTEMP = (rfGuessTEMP - spotLocationY(picktrk)) / ...
        zf(nFrames,picktrk); % Estimate RF values for last frame
    rfGuess = repmat(rfGuessTEMP,1,nTracks);
end
if strcmp(measureRF,'d')
    disp(['Specify "rfGuess" values for ' num2str(nSpots) ...
        ' analyte in descending'])
    rfGuessTEMP = input(['RF value as a row vector with elements ' ...
        '[0,1]: ']);
    rfGuess = repmat(rfGuessTEMP,1,nTracks);
end

% Display current values
disp('Current "rfGuess" values (one analyte per row, ')

```

```

disp('one track per column) are:')
disp(rfGuess)

% Use rfGuess to set limits of possible spot RF values (rfLimits)
% Repeat criteria used by A.J. Oko in getHRf.m to set limits
rfLimits = zeros([size(rfGuess) 2]);
rfLimitsLOW = rfGuess; rfLimitsUP = rfGuess;
rfLimitsLOW(rfGuess < 0.05) = rfGuess(rfGuess < 0.05) - 0.03;
rfLimitsLOW(rfGuess >= 0.05) = rfGuess(rfGuess >= 0.05) / 1.3;
rfLimitsUP(rfGuess < 0.05) = rfGuess(rfGuess < 0.05) + 0.03;
rfLimitsUP(rfGuess >= 0.05) = rfGuess(rfGuess >= 0.05) * 1.3;
rfLimits(:,1) = rfLimitsLOW; rfLimits(:,2) = rfLimitsUP;
disp("'rfLimits" calculated from above based on A.J. Oko criteria')

% Ensure values are valid and confirm
disp(' ')
good_rfGuess = min(~isnan(rfGuess(:))) && ...
    (min(rfGuess(:)) >= -0.1) && (max(rfGuess(:)) <= 1.1) && ...
    (size(rfGuess,1) == nSpots) && (size(rfGuess,2) == nTracks);
good_rfLimits = (size(rfLimits,1) == nSpots) && ...
    (size(rfLimits,2) == nTracks) && (size(rfLimits,3) == 2) && ...
    min(~isnan(rfLimits(:)));
if good_rfGuess && good_rfLimits
    confirm = input('Confirm these values (y/n): ','s');
else
    disp("'rfGuess" and "rfLimits" values are invalid. Redefine.')
end
clear measureRF picktrk rfGuessTEMP rfLimitsLOW rfLimitsUP;
clear good_rfGuess good_rfLimits;
if ~strcmp(confirm,'y') % Clear and redefine
    clear rfGuess rfLimits;
end
end
save([recipePath newRecipe]); % Save changes

%% Define blank regions on frames
disp(' ')
disp('PART 8: DEFINE BLANK REGIONS FOR SYSTEM NOISE
CALCULATION')
confirm = 'n'; close all;
while ~strcmp(confirm,'y')
    disp(' ')
    % Blank tracks between separation tracks
    disp(['Defining "blank" regions between ' num2str(nTracks) ...
        ' separation'])
    disp('tracks to use in system noise calculation ("defineBlank")')
end

```

```

% Set the top left corner position and width of blank track
disp('Calculating position and width of blank regions.')
defineBlank0 = zeros((nTracks-1),4); % 2D array to be tiled into 3D
defineBlank0(:,1) = defineTrack(1:(nTracks-1),1) + ...
    defineTrack(1:(nTracks-1),3) + 1;
defineBlank0(:,2) = ceil( max(spotLocationY(1:(nTracks-1)), ...
    spotLocationY(2:nTracks)) );
defineBlank0(:,3) = defineTrack(2:nTracks,1) - defineBlank0(:,1) - 1;
defineBlank = repmat(defineBlank0,[1 1 nFrames]);

% Set bottom edge of crops manually or float with migration front
% (within manual/automatic bounds)
disp(' ')
disp('Manually/automatically set height of blank regions')
confirm2 = 'n';
while ~strcmp(confirm2,'y')
    clf; imshow(middleFrame); axis on; hold on; % Show middle frame
    disp(' ')
    % Check existing variables if present
    good_LowEdge = exist('defineBlank_LowEdge','var') && ...
        isnumeric(defineBlank_LowEdge) && ...
        (defineBlank_LowEdge > max(defineBlank0(:,2))) && ...
        (defineBlank_LowEdge <= size(middleFrame,1));
    good_LowEdgeType = exist('defineBlank_LowEdgeType','var') && ...
        ( strcmp(defineBlank_LowEdgeType,'fix') || ...
        strcmp(defineBlank_LowEdgeType,'front') );
    % Define new values if not present or invalid
    if ~good_LowEdge || ~good_LowEdgeType
        % How should bottom edge of blank regions be defined
        disp(' How should the lower edge of the blank regions')
        disp(' be defined? Fix to a given position ("fix") or')
        disp(' move with migration front ("front") up to a set')
        defineBlank_LowEdgeType = ...
            input(' bound (type without quotes): ','s');
        disp(' Define this manually (m) or with graphical')
        defineBlank_LowEdge = input(' interface (g): ','s');
        % What value should be the edge/bound on the blank regions
        if strcmp(defineBlank_LowEdge,'g')
            disp(' Use cursors to define blank region lower edge')
            disp(' position or bound (ignore x-cursor).')
            [dummy,defineBlank_LowEdge] = ginput(1);
            clear dummy
            defineBlank_LowEdge = floor(defineBlank_LowEdge);
            disp([' Pixel position of ' ...
                num2str(defineBlank_LowEdge) ' defined.'])
        end
    end
end

```

```

else
    disp('  Manually enter a pixel position/bound between')
    disp('  starting position and bottom of frame. ie.')
    defineBlank_LowEdge = input(['  between ' ...
        num2str(max(defineBlank0(:,2))) ' and ' ...
        num2str(size(middleFrame,1)) ' px.: ']);
end
% Error correct if necessary
if defineBlank_LowEdge <= max(defineBlank0(:,2))
    defineBlank_LowEdge = max(defineBlank0(:,2)) + 1;
elseif defineBlank_LowEdge > size(middleFrame,1)
    defineBlank_LowEdge = size(middleFrame,1);
end
disp(' ')
end
% Display current blank edge calculation method and position
if strcmp(defineBlank_LowEdgeType,'fix')
    disp('  Bottom edge of blank regions will be FIXED to')
    disp(['  a position ' num2str(defineBlank_LowEdge) ...
        ' px from top of frame.'])
else
    disp('  Bottom edge of blank regions will MOVE WITH ')
    disp('  MIGRATION FRONT to a maximum position of ')
    disp(['  ' num2str(defineBlank_LowEdge) ...
        ' px from top of frame.'])
end
% Check current variables
good_LowEdge = exist('defineBlank_LowEdge','var') && ...
    isnumeric(defineBlank_LowEdge) && ...
    (defineBlank_LowEdge > max(defineBlank0(:,2))) && ...
    (defineBlank_LowEdge <= size(middleFrame,1));
good_LowEdgeType = exist('defineBlank_LowEdgeType','var') && ...
    ( strcmp(defineBlank_LowEdgeType,'fix') || ...
    strcmp(defineBlank_LowEdgeType,'front') );
disp(' ')
if good_LowEdge && good_LowEdgeType
    disp('  Preview defined position/bound as green line.')
    plot([1:size(middleFrame,2)],... % Add green line
        [defineBlank_LowEdge;defineBlank_LowEdge],'-g');
    hold off;
    confirm2 = input('  Confirm these values (y/n): ','s');
else
    disp('  Redefine: "defineBlank_LowEdge" and/or')
    disp('  "defineBlank_LowEdgeType" variables are invalid.')
end
if ~strcmp(confirm2,'y')

```

```

        clear defineBlank_LowEdge defineBlank_LowEdgeType;
    end
    clear good_LowEdge good_LowEdgeType;
end
clear confirm2;

% Calculate defineBlankHeight required to satisfy the above
defineBlankHeight = defineBlank_LowEdge .* ...
    ones(size(zf0,1),(nTracks-1)); % Fixed position wrt top of frame
if ~strcmp(defineBlank_LowEdgeType,'fix') % Moving lower edge
    % Set lower edge of blank using front position wrt top of frame
    % Consider front of track to left then to right of blank region
    defineBlankHeight = min( defineBlankHeight, zf0(:,1:(nTracks-1)) );
    defineBlankHeight = min( defineBlankHeight, zf0(:,2:nTracks) );
end
defineBlankHeight = defineBlankHeight - ... % wrt top of blank region
    repmat(defineBlank0(:,2),nFrames,1);
defineBlank(:,4,:) = floor( permute(defineBlankHeight,[2 3 1]) );

% Delete blank tracks with 0 or negative calculated track widths
defineBlankTEST = defineBlank;
defineBlankTEST(find(defineBlankTEST(:,3,1) <= 0),:,:) = [];
if size(defineBlank,1) ~= size(defineBlankTEST,1)
    disp([' Deleting ' num2str(size(defineBlank,1) - ...
        size(defineBlankTEST,1)) ' invalid blank tracks since these'])
    disp(' have negative or 0 track width.')
    defineBlank = defineBlankTEST;
end
clear defineBlankTEST;
nBlankTracks = size(defineBlank,1); % Number of blank tracks
disp(' ')
disp(['"defineBlank" array defined. Rows for each of ' ...
    num2str(nBlankTracks) ' blank regions, '])
disp('columns for (x,y,width,height), layers for each frame.')

% Display an overview of blank regions
clf; imshow(middleFrame); axis on; hold on; % Show middle frame
disp(' ')
disp('Preview MIDDLE frame with overlaid blank region definitions:')
if strcmp(defineBlank_LowEdgeType,'fix')
    disp(' Green : Bottom edge of all blank regions in all frames.')
else
    disp(' Green : Bottom edge of blank regions move with the')
    disp(' migration fronts to a maximum position represented')
    disp(' by this line.')
end
end

```

```

plot([1:size(middleFrame,2)],... % Add green line
     [defineBlank_LowEdge;defineBlank_LowEdge],'-g');
disp('  Red   : Blank regions for MIDDLE frame.')
disp('  Blue  : Blank regions for LAST frame.')
for i = 1:nBlankTracks
    rectangle('Position', defineBlank(i,:,round(nFrames/2)),...
              'EdgeColor','r'); % Middle frame
    rectangle('Position', defineBlank(i,:,nFrames),...
              'EdgeColor','b'); % Last frame
end
hold off;

% Confirm calculated values and continue
disp(' ')
confirm = input('Confirm these values (y/n): ','s');
if ~strcmp(confirm,'y')
    clear defineBlank nBlankTracks;
end
clear defineBlank0 defineBlankHeight;
end

%
% confirm = 'n';
% while ~strcmp(confirm,'y')
%     disp(' ')
%     if ~exist('VARIABLE','var')
%     else
%         disp('VARIABLE:')
%         disp('VARIABLE')
%     end
%     confirm = input('Confirm these values (y/n): ','s');
%     if ~strcmp(confirm,'y') % Clear hueRange and redefine
%         clear 'VARIABLE';
%     end
% end

%% Conclusion
clear confirm* i j trackNum tempFrame dummyImage;
save([recipePath newRecipe]); % Save changes
disp('***SUCCESS SCRIPT FINALLY DONE!!!***')

end

```


A.1.1.3 getChi4.m

```
function [chi,zStart] = getChi4(filepath,filename,chitrkloc,nSteps)

%% Define frame numbers
close all;
disp(' ')
if ~exist('nSteps','var') || (nSteps > length(filename))
    nSteps = 10; % A default value for nSteps.
    disp('Assume a default value of nSteps = 10')
end
startFrameNum = 1;
stepSize = round(length(filename) / (nSteps+2));
finishFrameNum = startFrameNum + (stepSize*(nSteps - 1));
A = imread([filepath filename(finishFrameNum).name]);
fmHeight = size(A,1); fmWidth = size(A,2);

%% Define region of interest (if required)
good_chitrkloc = exist('chitrkloc','var') && ...
    (size(chitrkloc,2) == 4) && (min(chitrkloc(:)) >= 0) && ...
    (max(chitrkloc(1)) <= fmWidth) && ... % Ensure x,width<fmWidth
    (max(chitrkloc(3)) <= fmWidth) && ...
    (max(chitrkloc(2)) <= fmHeight) && ... % Ensure y,Height<fmHeight
    (max(chitrkloc(4)) <= fmHeight);
if ~good_chitrkloc
    disp(' ')
    disp('No valid region of interest ("chitrkloc") row vector defined.')
    disp('Crop area to be used to find front position. Recommend cropping')
    disp('centre two tracks and regions to left/right of these tracks.')
    disp('This zone will be shown with plotted signals.')
    clf;
    [dummy,chitrkloc] = imcrop(A);
end
% Correct chitrkloc if required (ensure it covers full frame height)
chitrkloc = round(chitrkloc);
chitrkloc(2) = 0; chitrkloc(4) = fmHeight;

% Measure front position in selected images
Zf_measured = zeros(nSteps,1); % Initialize Zf_measured vector
disp('Measure migration front positions. If unable to identify, select')
disp('a point behind the applied spot position to skip.')
disp(' ')
for stepNum=1:nSteps
    % Import and crop image to chitrkloc
    tempImage = imread([filepath ...
        filename(startFrameNum + stepSize*(stepNum-1)).name]);
```

```

A = imcrop(tempImage, chitrkloc);

% Display rotated track image and title
fig = figure(1);
set(fig,'Position', [100 100 1100 800]);
clf; sp = zeros(1,2); % Initialize subplot handles
sp(1) = subplot(2,1,1);
% Use image() instead of imshow() since it permits non-square pixels.
% This allows the image to stretch to fill the figure and match the
% axis of the subplot(2,1,2) below
image(imrotate(A,90)); axis on;
set(sp(1),'Position',[0.1 0.6 0.85 0.35]);
title(['Region ' num2str(stepNum) ' of ' num2str(nSteps) ...
' cropped from: ' filename(startFrameNum + ...
stepSize*(stepNum-1)).name ' (rotated)']);
xlim([0 size(A,1)]);

% Calculate and plot lightness [0,255]
L = 0.5 * (double(max(A,[],3)) + double(min(A,[],3)) );
sp(2) = subplot(2,1,2); hold on;
set(sp(2),'Position',[0.1 0.1 0.85 0.5]);
plot(sum(A(:,1),2),'-r'); % Add RED signal
plot(sum(A(:,2),2),'-g'); % Add GREEN signal
plot(sum(A(:,3),2),'-b'); % Add BLUE signal
plot(sum(L,2),'-k'); % Add LIGHTNESS signal
xlabel('Position [px]'); ylabel('Summed R,G,B,L [0,255] values');
legend('RED','GREEN','BLUE','LIGHTNESS', 'Location', 'NorthEast');
xlim([0 size(A,1)]); hold off;
linkaxes(sp,'x');

% Click on the front position to retrieve x-position. Note that ginput
% returns x,y positions within the subplot in which the user clicks.
% Outside of the subplots, ginput returns positions that would be
% measured on the current axes (ie. those of subplot 2 above).
% However, since the axes are linked, the user may click anywhere
% within the figure to retrieve the correct x-position.
[Zf_measured(stepNum),dummy] = ginput(1);
disp(['Measured front position = ' num2str(Zf_measured(stepNum)) ...
' px'])
% Discard value if it is lower than in previous frame
if (Zf_measured(stepNum) < Zf_measured(1))
    disp(' Could not identify front position. (It is less than')
    disp(' the first measurement.) Set to "NaN" and exclude.')
    Zf_measured(stepNum) = NaN;
end
end
end

```

```

%% Fit the chi and zStart values
Zf_measured(Zf_measured == 0) = nan;
options = optimset('Display','off','LargeScale','off');
time = (startFrameNum:stepSize:finishFrameNum)'; % in units of "frames"

% Edit out erroneous data (NaN and <0)
gooddata = (time >= 0) & (Zf_measured >= 0);
Zf_measured = Zf_measured(gooddata);
time = time(gooddata);

% Define guesses for fitting (chi_guess is the average of measured values)
chi_guess = mean(Zf_measured.^2 ./ time);
zStart_guess = 0;
parameters = [chi_guess zStart_guess];
% Fit measured data to extract chi and zStart scalar values
output = fminunc(@migrationfit3,parameters,options,Zf_measured,time);
chi = output(1);
zStart = output(2);
clearvars -except chi zStart;

end

```

A.1.1.4 migrationfit3.m

```

function [squared_diff] = migrationfit3(parameters,Zf_measured,time)

% Zf_measured: Vector containing raw measured front positions
% time:        Vector containing frame numbers at which Zf was measured

% Key parameters to be fit
chi = parameters(1);
Zo = parameters(2);

% Edit out erroneous data (NaN and <0)
gooddata = (time >= 0) & (Zf_measured >= 0);
time = time(gooddata);
Zf_measured = Zf_measured(gooddata);
clear gooddata;

% Vector of REAL differences between the fit and measured data
variation = real( sqrt(chi.*time) + Zo - Zf_measured );

% A positive quantity to be minimized by fminunc

```

```
squared_diff = sum(variation.^2);
end
```

A.1.2 Colour-Filtered Densitogram Extraction

This section involves 1 main function and 2 sub-functions:

MAIN 1: mainUTLCv8.m : This script is used to analyze a directory of frame *.jpg files extracted from a UTLC development video. It is used to extract colour-filtered chromatograms from tracks defined on each frame in the directory. This script produces one output file. This contains all data produced in this script and that included in the recipe file.

SUB 1: filteredDensitogram.m : This script is designed to analyze an input UTLC track image ('img', developed top to bottom) or a blank track region given a set of colour filters ('hueRange'). It then outputs an array ('D') whose columns contain colour-filtered densitograms. It also outputs row vectors containing the calculated mean ('avg') and standard deviation ('stdev') of the colour-filtered saturation values.

SUB 2: gaussFitUTLC.m : This function is used to fit a gaussian of the form : $y = \text{amp} * \exp(-(x-\mu)^2 / (2*\sigma^2))$ to an input UTLC video colour-filtered densitogram signal. It assumes that the input signal is gaussian in nature and contains only one peak that falls within a region defined by rfLimits.

A.1.2.1 mainUTLCv8.m

```
%% Set file directories
clear all; close all;
disp(' ')
disp('PART 0: SET INPUT RECIPE AND OUTPUT DATA
DIRECTORIES/FILES')
disp(' ')
% Select and load recipe file
disp('Select input recipe file.')
[ipRecipe0,ipRecipePath0] = uigetfile('*.mat', ...
    'Select input recipe file');
ipRecipePath0 = [ipRecipePath0 '\'];
load([ipRecipePath0 ipRecipe0]);
ipRecipe = ipRecipe0; ipRecipePath = ipRecipePath0;
clear ipRecipe0 ipRecipePath0;
disp(['Loaded input recipe file "' ipRecipe '".'])

% Specify output file name
```

```

disp(' ')
disp('Define location and name of output file. This contains data')
disp('from input recipe file and calculations from this script.')
opFilePath = uigetdir(ipRecipePath,['Select directory in which to '...
    'place output file.']);
opFilePath = [opFilePath '\'];
opFile = input('Enter output data file name (name.mat): ','s');

% Initialize output recipe file (verify that save is permitted)
save([opFilePath opFile]);

% Set directory of UTLC video frames to analyze:
disp(' ')
disp('CONFIRM directory of frame *.jpg extracted from video')
% Assume location to be same as defined when creating recipe
filepath = uigetdir(filepath,...
    'CONFIRM directory of frame *.jpg extracted from video');
filepath = [filepath '\'];

% Initialize relevant filename, image arrays, etc.
filename = dir([filepath '*.jpg']);
nFrames = length(filename);
disp(['Selected directory contains ' num2str(nFrames) ' frames.'])
disp(['The first frame is: ' filename(1).name ''])
disp(['The last frame is: ' filename(nFrames).name ''])

%% Display possible and set desired types of analyses
disp(' ')
disp('PART 1: DEFINE SUBSET OF AVAILABLE DATA TO ANALYZE')
disp(' ')
nRows = size(lastFrame,1);
disp(['Input file ' ipRecipePath])
disp(['contains data for ' num2str(nTracks) ' separation and ' ...
    num2str(nBlankTracks) ' blank tracks.'])
disp(['Defined hue ranges for ' num2str(nSpots) ' analytes in ' ...
    'descending hRF are:'])
disp(hueRangeLabel)
% Specify subset of data if desired
confirm = 'n';
while ~strcmp(confirm,'y')
    % Preview predefined separation and blank tracks
    disp('Preview separation and blank track regions on last frame.')
    clf; imshow(lastFrame); axis on; hold on;
    title('Predefined separation (black) and blank (red) tracks')
    for i = 1:nTracks % Overlay separation tracks and applied spots
        rectangle('Position', defineTrack(i,:), 'EdgeColor', 'k');
    end
end

```

```

    plot(spotLocationX(i),spotLocationY(i),'ok');
end
for j = 1:nBlankTracks % Overlay blank tracks
    rectangle('Position', defineBlank(j,:,nFrames),'EdgeColor','r');
end
hold off;

% Define subset of separation tracks and hue ranges
disp('Select a subset of tracks and hue ranges if desired.')
disp('Enter row vectors with elements sorted in ascending order.')
sVect = sort(input('  Analyte spots on each track to analyze: '));
tVect = sort(input('  Separation tracks to analyze: '));
tbVect = sort(input('  Blank tracks to analyze: '));

% Select subset of relevant variables
nTracks1 = length(tVect);
defineTrack1 = defineTrack(tVect,:);
spotLocationX1 = spotLocationX(tVect);
spotLocationY1 = spotLocationY(tVect);
zf1 = zf(:,tVect);
nSpots1 = length(sVect);
hueRange1 = hueRange(sVect,:);
hueRangeLabel1 = hueRangeLabel(sVect);
spotMass1 = spotMass(sVect,tVect);
spotVolume1 = spotVolume(tVect);
spotdim_mm1 = spotdim_mm(tVect);
spotdim_px1 = spotdim_px(tVect);
rfLimits1 = rfLimits(sVect,tVect,:);
nBlankTracks1 = length(tbVect);
defineBlank1 = defineBlank(tbVect,:,:);

% Preview selected separation and blank tracks
disp(' ')
disp(['Selected hue ranges for ' num2str(nSpots1) ' analytes in ' ...
    'descending hRF are:'])
disp(hueRangeLabel1)
disp('Preview separation and blank track regions on last frame.')
clf; imshow(lastFrame); axis on; hold on;
title('Selected separation (black) and blank (red) tracks')
for i = 1:nTracks1 % Overlay separation tracks and applied spots
    rectangle('Position', defineTrack1(i,:), 'EdgeColor','k');
    plot(spotLocationX1(i),spotLocationY1(i),'ok');
end
for j = 1:nBlankTracks1 % Overlay blank tracks
    rectangle('Position', defineBlank1(j,:,nFrames),'EdgeColor','r');
end

```

```

hold off;

% Confirm values
disp(' ')
confirm = input('Confirm these values (y/n): ','s');
if ~strcmp(confirm,'y') % Clear selections and redefine
    clear sVect tVect tbVect;
    disp(' ')
end
end
close all;

%% Initialize variables
x = (1:nRows)'; % Vector for position along tracks

% Raw and fitted densitogram data (temporary variables)
Densitogram0 = zeros(nRows,nSpots1);
fitDensitogram0 = zeros(nRows,nSpots1);

% Temporary row vectors used to store fitted parameters for spots on a
% given track in a given frame; also other fitting/calculated values
sigma = zeros(1,nSpots1); % fitted peak stdev
mu = zeros(1,nSpots1); % fitted peak centre
amp = zeros(1,nSpots1); % fitted peak amplitude
zs = zeros(1,nSpots1); % fitted peak centre wrt applied spot
tolerance = 0.5; % fraction of peak height to use in sigma guess
% Arrays used to store relevant fitted peak data
stdSpot = zeros(nFrames,nSpots1,nTracks1); % Width of peak (one sigma)
zSpot = zeros(nFrames,nSpots1,nTracks1); % Peak centre wrt applied spot
ampSpot = zeros(nFrames,nSpots1,nTracks1); % Amplitude of fitted peak
% Arrays used to assess "goodness" of fit
sumResiduals = zeros(nFrames,nSpots1,nTracks1);
sumSqResiduals = zeros(nFrames,nSpots1,nTracks1);
stdResiduals = zeros(nFrames,nSpots1,nTracks1);

% Arrays to contain mean and standard deviation of hue-filtered saturation
% values on each blank track and for all pixels extracted from blank tracks
avgBlankTrk = zeros(nFrames,nSpots1,nBlankTracks1);
stdevBlankTrk = zeros(nFrames,nSpots1,nBlankTracks1);
avgBlank = zeros(nFrames,nSpots1);
stdevBlank = zeros(nFrames,nSpots1);

%% Confirm start
disp(' ')
disp('PART 2: PROCESS FRAMES EXTRACTED FROM UTLC
DEVELOPMENT VIDEO')

```

```

disp(' ')

% Initialize Densitogram and fitDensitogram if required
disp('Save "Densitogram" and "fitDensitogram" arrays (y/n)?')
disp('Warning: These data arrays may require very large amounts of')
saveDensitogram = input('memory!'): 's');
if strcmp(saveDensitogram,'y')
    % Raw and fitted densitogram data
    Densitogram = zeros(nRows,nSpots1,nFrames,nTracks1);
    fitDensitogram = zeros(nRows,nSpots1,nFrames,nTracks1);
    disp('    Initialized "Densitogram" and "fitDensitogram" arrays.')
end

disp(' ')
process = input('Start processing data or exit (y/n)?: 's');
if ~strcmp(process,'y')
    error('Script stopped by Silly user... :-(');
end
clear process;

%% Begin analysis
tic; tstart = tic;          % Start timer

for f = 1:nFrames
    disp(['Now analyzing frame ' num2str(f) ' of ' num2str(nFrames)])
    rgbFrame = imread([filepath filename(f).name]);

    %% Extract colour-filtered densitograms and fit with gaussian

    for t = 1:nTracks1
        imgT = imcrop(rgbFrame,defineTrack1(t,:));
        % [D,avg,stdev] = filteredDensitogram(img,hueRange)
        [Densitogram0(:,t),dummy,dummy] =
        filteredDensitogram(imgT,hueRange1);
        clear dummy
        % Fit densitogram to gaussian function
        for s = 1:nSpots1
            % [sigma,mu,amp]=gaussFitUTLC(xRows,y,h,init_spot,zf,rfLimits)
            [sigma(s),mu(s),amp(s)] = gaussFitUTLC(...
            x, Densitogram0(:,s), tolerance, spotLocationY1(t), ...
            zf1(f,t), [rfLimits1(s,t,1) rfLimits1(s,t,2)]);
            % Calculate fitDensitogram0 signal
            zs(s) = mu(s) - spotLocationY1(t);
            fitDensitogram0(:,s) = amp(s) * exp( -(x-mu(s)).^2 / ...
            (2*sigma(s)^2) ); % Equation for gaussian distribution
            % Isolate the region of the fit +/- 2 sigma away from the mean,

```



```

% calculate residuals, sum residuals and squared residuals
leftLimit = round(mu(s) - 2*sigma(s));
rightLimit = round(mu(s) + 2*sigma(s));
% If isolated region does not exceed edges of plate, calculate:
% sum of the residuals, sum of the squared residuals, and stdev
% of residuals to assess "goodness" of fit.
if (leftLimit >= 1) && (rightLimit <= nRows)
    residuals = ...
        fitDensitogram0(leftLimit:rightLimit,s) - ...
        Densitogram0(leftLimit:rightLimit,s);
    sumResiduals(f,s,t) = sum(abs(residuals));
    sumSqResiduals(f,s,t) = sum(residuals.^2);
    stdResiduals(f,s,t) = std(residuals);
else
    % Isolated ROI exceeds at least one of the plate edges
    sumResiduals(f,s,t) = NaN;
    sumSqResiduals(f,s,t) = NaN;
    stdResiduals(f,s,t) = NaN;
end
end
% Store the calculated values
stdSpot(f,:,t) = sigma;
zSpot(f,:,t) = zs;
ampSpot(f,:,t) = amp;

% Save Densitogram0 and fitDensitogram0 signals to arrays iff
% desired
if strcmp(saveDensitogram,'y')
    Densitogram(:,f,t) = Densitogram0;
    fitDensitogram(:,f,t) = fitDensitogram0;
end
end

%% Consideration of blank pixel noise (stdev) and average

% 1 px wide image to hold all pixels extracted from the blank tracks
blankpx = uint8( zeros(...
    dot((defineBlank1(:,3,f)+1),(defineBlank1(:,4,f)+1)), 1, 3) );

% Analyze each blank track and add all pixels to blankpx
ind1 = 1; % Start insertion index
for tb = 1:nBlankTracks1
    imgB = imcrop(rgbFrame,defineBlank1(tb,:,f));
    % [D,avg,stdev] = filteredDensitogram(img,hueRange)
    [dummy,avgBlankTrk(f,:,tb),stdevBlankTrk(f,:,tb)] = ...
        filteredDensitogram(imgB,hueRange1);

```

```

clear dummy
% Append cropped blank pixels to blankpx array
imgB = reshape(imgB,[],1,3);
ind2 = ind1 + size(imgB,1) - 1; % Calculate end index
blankpx(ind1:ind2,1,:) = imgB; % Insert all of imgB pixels
ind1 = ind2 + 1; % Increment start index
end

% Analyze entire set of blank track pixels together
[dummy,avgBlank(f,:),stdevBlank(f,:)] = ...
    filteredDensitogram(blankpx,hueRange1);

clear imgT imgB ind1 ind2 blankpx dummy;
% Estimate time remaining
testimate = toc(tstart) * (nFrames-f) / f;
disp([' Elapsed time is ' num2str(fix(toc)) ' seconds.'])
disp([' Estimated time remaining: ' num2str(fix(testimate)) ...
    ' seconds.'])
end
clear i j f t s tb zs sigma amp mu residuals tstart testimate confirm;
clear leftLimit rightLimit Densitogram0 fitDensitogram0;

% Save calculated data
disp(' ')
disp(['Saving output file as: "' opFile '"'])
save([opFilePath opFile]); % Save changes
disp('***SUCCESS SCRIPT FINALLY DONE!!!***')

end

```

A.1.2.2 filteredDensitogram.m

```

function [D,avg,stdev] = filteredDensitogram(img,hueRange)

%% Error checking
good_img = (size(img,3) == 3) && isa(img(:),'uint8');
if ~good_img
    error('Input RGB image array "img" is invalid, Silly!')
end
good_hueRange = (size(hueRange,2) == 2) && (max(hueRange(:)) <= 360) &&
...
    (min(hueRange(:)) >= 0);
if ~good_hueRange
    error('Input "hueRange" array is invalid, Silly!')

```

```

end
hueRange(hueRange == 360) = 0; % Check for 360 == 0 discontinuity

%% Analyze track image
% RGB -> HSV conversion
img = rgb2hsv(img);
img(:,:,1) = 360 .* img(:,:,1); % Scale hue to range [0,360)

% Initialize output array and vectors
nFilters = size(hueRange,1);
D = zeros(size(img,1),nFilters);
avg = zeros(1,nFilters); stdev = zeros(1,nFilters);

% Define and apply hue filters
filteredSat = zeros(size(img,1),size(img,2));
for i = 1:size(hueRange,1)
    if hueRange(i,1) <= hueRange(i,2) % Ex. [30 200]
        filteredSat = img(:,:,2) .* ( (img(:,:,1) >= hueRange(i,1)) & ...
            (img(:,:,1) <= hueRange(i,2)) );
    elseif hueRange(i,1) > hueRange(i,2) % Ex. [200 30]
        filteredSat = img(:,:,2) .* ( (img(:,:,1) >= hueRange(i,1)) | ...
            (img(:,:,1) <= hueRange(i,2)) );
    end
    D(:,i) = sum(filteredSat,2); % hue-filtered densitogram
    avg(i) = mean(filteredSat(:));
    stdev(i) = std(filteredSat(:));
    clear filteredSat;
end
clearvars -except D avg stdev;

end

```

A.1.2.3 gaussFitUTLC.m

```

function [sigma,mu,amp] = gaussFitUTLC(x,y,h,init_spot,zf,rfLimits)

% Inputs:
% x : position vector [1 2 3 ...]'
% y : densitogram signal to be fit
% h : a fitting parameter used to make an initial guess at the peak's
%     standard deviation. It is the threshold which is the fraction
%     from the maximum y height from which the initial guess for standard
%     deviation is derived. h should be a number within (0,1).
% init_spot : position of the initial applied sample spot

```

```

% zf : migration front position
% rfLimits : limits used with zf to set bounds on the input signal. It
% ensures that fitting occurs only in a region known to contain the
% isolated peak.

%% Definitions
L = length(y); % Length of input vector
init_spot = round(init_spot);

% Find min and max values for each analyte (lower/upper limit of peak
% fitting region calculate wrt init_spot). If necessary, adjust lower
% limit to x = 1 and/or upper limit to x = L
minZS = round(zf*rfLimits(1));
if (init_spot + minZS) <= 1
    minZS = 1 - init_spot;
    % A negative value so that lower bound: (init_spot + minZS) = 1
end
maxZS = round(zf*rfLimits(2));
if (init_spot + maxZS) >= L
    maxZS = L - init_spot;
    % A positive value so that upper bound: (init_spot + minZS) = L
end

%% Initial guesses (based on smoothed signal)

% Create smoothed signal used only in calculation of initial guesses
y1 = smooth(y,5); % 5-point moving average smooth

% Guess location of peak centre as point where smoothed signal is highest
[ymax,loc] = max(y1( (init_spot + minZS) : (init_spot + maxZS) ));
loc = loc + init_spot + minZS - 1;

% Identify two points a,b to left,right of peak at which the signal is
% h*ymax. (Points wrt to start of signal)
a = find(y1(1:loc) < (h*ymax),1,'last');
b = loc + find(y1(loc:L) < (h*ymax),1,'first') - 1;

if isempty(a) || isempty(b) || isempty(loc) % No valid guesses
    mu = NaN;
    sigma = NaN;
    amp = NaN;
    return
end

% Guess standard deviation using calculated (b-a) distance at relative
% height h (Note that log(h) < 0 since 0<h<1)

```

```

stdv = sqrt(-((b - a)^2)/(8*log(h)));
% If guess at standard deviation is too wide (edges of signal are less than
% 3 * stdevs away).
if (round(loc - 3*stdv) < 1) || (round(loc + 3*stdv) > L)
    mu = NaN;
    sigma = NaN;
    amp = NaN;
    return
end

%% Perform gaussian fit (based on original signal)

% Exclude data more than 3 standard deviations (guessed) left and right of
% the peak centre.
y(1:round(loc - 3*stdv)) = NaN;    % left of peak
y(round(loc + 3*stdv):L) = NaN;    % right of peak

% Define fitting options and initial guesses
options = optimset('Display','off','LargeScale','off');
guess = [loc,stdv,ymax];

% Apply Gaussian Fit
params = fminunc(@fitGaussian1D,guess,options,y,x);
mu = params(1);    % fitted peak centre wrt start of signal
sigma = params(2); % fitted peak standard deviation
amp = params(3);   % fitted peak amplitude

%% Check validity of fit

% amp, min/max sigma values by A.J. Oko; allowed peak locations by S.R. Jim
badfit = (amp > (1.5*ymax)) || (mu <= 1) || (mu >= L) || ...
    (sigma <= 1.5) || (sigma >= 100);
if badfit
    mu = NaN;
    sigma = NaN;
    amp = NaN;
end

end

```

A.1.3 Figure of Merit Calculations

This section involves 1 main function:

MAIN 1: fomUTLC.m : Given information produced by the recipe file (output by createRecipeUTLCv*.m) and calculated/fitted data file (output by

mainUTLC.v*.m), this script calculates several figures of merit (FOM) that are used to evaluate performance from a UTLC development video.

A.1.3.1 fomUTLC.m

% Input: Specify the *.mat file that contains information produced by the
% aforementioned scripts.

% Outputs: Relevant figures of merit saved in a new *.mat file.

```
%% Set file directories
clear all; close all;
disp(' ')
disp('PART 0: SET INPUT DATA AND OUTPUT FOM DATA FILE
DIRECTORIES/FILES')
disp(' ')
% Select and load recipe file
disp('Select input data file (output by mainUTLCv*.m)')
[ipData0,ipDataPath0] = uigetfile('*.mat', ...
    'Select input data file (output by mainUTLCv*.m)');
ipDataPath0 = [ipDataPath0 '\'];
load([ipDataPath0 ipData0]);
ipData = ipData0; ipDataPath = ipDataPath0;
clear ipData0 ipDataPath0;
disp(['Loaded input data file "' ipData "''])

% Specify output file name
disp(' ')
disp('Define location and name of output figures of merit data file. This')
disp('contains data from input file and FOMs calculate in this script.')
opDataPath = uigetdir(ipDataPath,['Select directory in which to '...
    'place output FOM data file.']);
opDataPath = [opDataPath '\'];
opData = input('Enter output FOM data file name (name.mat): ','s');

% Delete Densitogram and fitDensitogram signals if present
if exist('Densitogram','var')
    clear Densitogram;
    disp("'Densitogram" data array cleared from memory.')
end
if exist('fitDensitogram','var')
    clear fitDensitogram;
    disp("'fitDensitogram" data array cleared from memory.')
end
```

```

% Initialize output recipe file (verify that save is permitted)
disp(['Saving data to ' opData ])
save([opDataPath opData]);

%% Basic calculations
disp(' ')
disp('PART 1: BASIC CALCULATIONS: hRF, FWHM, PN, PH')

% hRF calculation
zf1_3D = repmat(permute(zf1,[1 3 2]),[1 nSpots1 1]);
spotHRF = 100 * zSpot ./ zf1_3D; % 100 * retention factor (RF)

% Peak width (FWHM)
spotFWHM = stdSpot*sqrt(8*log(2));

% Theoretical plate number and plate height
spotPN = (zSpot./stdSpot).^2;
spotPH = (stdSpot.^2) ./ zSpot; % = migration distance / plate number

clear zf1_3D;
disp(' ')
disp('Calculated hRF, FWHM, PN, PH')
disp(['Saving data to ' opData ])
save([opDataPath opData]); % Save changes

%% Resolution
disp(' ')
disp('PART 2: RESOLUTION BETWEEN ANALYTE SPOTS')
% Arrange arrays to calculate resSpot(analyte1,analyte2,frame,track)
% Example: resSpot(2,3,100,5) is the resolution between analytes 2 and 4
% (=3+1) in frame 100 on track 5.
% Introduce a 4th dimension (analyte2) to arrays with original dimensions
% (frame x analyte x track).

% Square base of "array1"s has ROWS that correspond to analyte1 number
% (analytes 1:(nSpots1-1)) and identical COLUMNS.
zSpot1 = permute(zSpot,[2 4 1 3]); % dim1 corresponds to analyte1 number
zSpot1 = repmat(zSpot1(1:(nSpots1-1),1,:,:),[1 (nSpots1-1) 1 1]);
stdSpot1 = permute(stdSpot,[2 4 1 3]); % similar to above
stdSpot1 = repmat(stdSpot1(1:(nSpots1-1),1,:,:),[1 (nSpots1-1) 1 1]);

% Square base of "array2"s has COLUMNS that correspond to analyte2 number
% (analytes 2:nSpots1) and identical ROWS.
zSpot2 = permute(zSpot,[4 2 1 3]); % dim2 corresponds to analyte2 number
zSpot2 = repmat(zSpot2(1,2:nSpots1,1,:,:),[(nSpots1-1) 1 1 1]);

```

```

stdSpot2 = permute(stdSpot,[4 2 1 3]); % similar to above
stdSpot2 = repmat(stdSpot2(1,2:nSpots1, :, :), [(nSpots1-1) 1 1 1]);

% Calculate resolution
resScale = 0.5; % Scaling factor depends on res criteria and spot width
resSpot = resScale * (zSpot1 - zSpot2) ./ (stdSpot1 + stdSpot2);

% Set all nonsensical and redundant resolution values to NaN
% Ex: resSpot(2,1,f,t) = NaN since this considers spot 2 vs spot 2
% Ex: resSpot(3,1,f,t) = NaN since this considers spot 3 vs spot 2 and
% resSpot(2,2,frame,track) already considers spot 2 vs spot 3.
% Use a lower triangular matrix of NaN values
resRedundant = repmat(tril(NaN*ones(nSpots1-1),-1),[1 1 nFrames nTracks1]);
resSpot = resSpot + resRedundant;
clear zSpot1 zSpot2 stdSpot1 stdSpot2 resScale resRedundant;
disp(['Calculated resolution between ' num2str(nSpots1) ' spots.'])
disp(['Saving data to ' opData])
save([opDataPath opData]); % Save changes

%% Signal, noise, and SNR
disp(' ')
disp('PART 4: SIGNAL, NOISE, AND SIGNAL-TO-NOISE')
disp(' ')

% Calculate the signal as the area under the fitted gaussian curves (input
% the length of the integration region in the development direction in
% number of stdevs).
disp('Define a symmetric region over which each fitted gaussian peak')
disp('should be integrated. Express the length of this region in')
disp('number of standard deviations ("sigmas"; ex. Type "4" for a')
disp('region that is +/- 2 sigma from the gaussian centre). Or type')
intLength = input("'FWHM" (with quotes) to use the sqrt(8*ln2): ');
confirm = 'n';
while ~strcmp(confirm,'y')
    if isa(intLength,'char') && strcmp(intLength,'FWHM')
        intLength = sqrt(8*log(2));
    end
    if isnumeric(intLength) && (length(intLength) == 1)
        intLength = abs(intLength);
        disp([' Selected integration region width is ' ...
            num2str(intLength) ' sigmas in length.'])
        confirm = input(' Confirm this value (y/n): ','s');
    end
    if ~strcmp(confirm,'y') % Clear hueRange and redefine
        clear intLength;
    end
end

```



```

end
clear confirm;
disp(' ')
% Consider integrated area to be the "signal" associated with each spot
% sigSpot(frame,spot,track)
sigSpot = sqrt(2*pi) * erf(intLength/(2*sqrt(2))) .* ampSpot .* stdSpot;
disp('Calculated signal as integral of fitted peak area for each spot')
disp('on each track in each frame.')

% Signal-to-noise ratio (SNR) calculation. Scale the noise by the absolute
% width of the track (px) (defineTrack(track,3,frame)+1) and of the
% integration zone length (px) (intLength*stdSpot).
% Single pixel noise is stdevBlank(frame,spot)
stdevBlank3D = repmat(stdevBlank,[1 1 nTracks1]);
% trkWidth scaling factor (+1 since a specified width of N pixels gives a
% crop N+1 pixels wide)
trkWidth3D = repmat( permute(defineTrack1(:,3),[2 3 1]), ...
    [nFrames nSpots1 1]) + 1;
% Array of specified integration region lengths (in px)
intLengthpx3D = intLength * stdSpot;

% Scale noise by area of spot for given mass -->
% an nFrames x nSpots1 x nTracks1 array
stdevBlankScaled = stdevBlank3D .* sqrt(trkWidth3D.*intLengthpx3D);
disp('Calculated noise scaled for 2D area of each spot on each track')
disp('in each frame.')

% Calculate SNR-- an nFrames x nSpots1 x nTracks1 array
spotSNR = sigSpot ./ stdevBlankScaled;
disp('Calculated SNR for each spot on each track in each frame.')

%% LOD
disp(' ')
disp('PART 5: LIMIT OF DETECTION (LOD)')
disp(' ')
% Skip analysis?
calcLOD = input('Calculate LOD (y/n)? :','s');
disp(' ')

% Display available data if it exists
if exist('spotMass1','var') && strcmp(calcLOD,'y')
    disp(['Spot signal, scaled noise, and area calculated for ' ...
        num2str(nSpots1) ' spots.'])
    disp('These spots have defined hue ranges of: ')
    disp(hueRangeLabel1)
    disp('For which spots should the LOD be calculated? It is best to ')

```

```

disp('select only those that have reliable gaussian fits.')
disp(' ')
disp(['"spotMass1" data exists for ' num2str(nSpots1) ' spots.'])
disp('This data corresponds to the subset of tracks ORIGINALLY')
disp('defined in recipe file that was selected in mainUTLCv*.m:')
disp(tVect) % Subset of tracks selected in main
disp('These tracks were renumbered by mainUTLCv*.m as NEW tracks:')
disp(1:nTracks1)
disp('Calculated mass [ng] of each analyte (rows) in spot applied ')
disp('to each of these NEW tracks (columns):')
disp(spotMass1)
disp('From which NEW tracks should the LOD be calculated?')
elseif exist('spotMass1','var') && ~strcmp(calcLOD,'y')
    disp('Sufficient data in "spotMass1" vector but analysis skipped.')
    calcLOD = 'n';
else
    disp('No "spotMass1" data exists. LOD cannot be calculated.')
    disp('Skip this analysis.')
    calcLOD = 'n';
end

disp(' ')
if strcmp(calcLOD,'y') % Proceed with LOD calculation setup
    % Select subset of data for LOD calculations
    confirm = 'n';
    while ~strcmp(confirm,'y')
        disp('Select subset of spots and NEW tracks for LOD calculation.')
        LODsVect = ...
            input(' Enter a row vector to select available spots: ');
        LODtVect = ...
            input(' Enter a row vector to select from NEW tracks: ');
        if isnumeric(LODsVect) && isnumeric(LODtVect)
            disp(' Calculated subset of "spotMass1" data:')
            LODspotMass1 = spotMass1(LODsVect,LODtVect); % Subset of masses
            disp(LODspotMass1)
            confirm = input(' Confirm this selection (y/n): ','s');
        else
            disp(' Invalid selections.')
        end
    end
    disp(' ')
end
clear confirm;

LODnSpots1 = size(LODspotMass1,1); % Number of spots to consider
% Subset of signals, noise, and SNR
LODsigSpot = sigSpot(:,LODsVect,LODtVect);

```

```

LODstddevBlankScaled = stdevBlankScaled(:,LODsVect,LODtVect);
LODspotSNR = spotSNR(:,LODsVect,LODtVect);

disp('On what criteria (# of sigmas) should LOD be based?')
lodcriteria = input('(ex. enter "3" for 3 * sigma criteria: ');

% LOD method 1: Use SNR
disp(' ')
disp('Method 1: Calculate "spotLOD0" based on SNR measurements')
spotLOD0 = zeros(nFrames,LODnSpots1); % calculated spot LODs
snrSpotFIT = zeros(nFrames,LODnSpots1,2); % linear regress. parameters
for f = 1:nFrames
    for s = 1:LODnSpots1
        % Extract relevant data and make into column vectors
        spotMass1_1 = LODspotMass1(s,:);
        spotSNR_1 = permute(LODspotSNR(f,s,:),[3 1 2]);
        % Ensure that there are at least two real number ordered pairs
        % (spotMass1_1(i),spotSNR_1(i)). Create a test vector that
        % uses the fact that Real_number .* NaN = NaN
        LOD0dataTest = spotMass1_1 .* spotSNR_1;
        if length(LOD0dataTest(~isnan(LOD0dataTest))) >= 2
            % Remove ordered pairs with NaNs as one or both ordinates
            spotMass1_1 = spotMass1_1(~isnan(LOD0dataTest));
            spotSNR_1 = spotSNR_1(~isnan(LOD0dataTest));
            % Linear regress. (ind. = 1), permute to insert into array
            snrSpotFIT(f,s,:) = permute( polyfit(spotMass1_1, ...
                spotSNR_1, 1), [1 3 2]);
            spotLOD0(f,s) = (lodcriteria - snrSpotFIT(f,s,2)) / ...
                snrSpotFIT(f,s,1);
        else % Cannot perform fit or calculate LOD
            snrSpotFIT(f,s,:) = NaN;
            spotLOD0(f,s) = NaN;
        end
        clear spotMass1_1 spotSNR_1 LOD0dataTest;
    end
end
clear f s;
disp('   Calculated LOD0 for each spot in each frame.')

% LOD method2: Use log10s of signals
disp(' ')
disp('Method 2: Calculate "spotLOD" based on absolute measurements')
disp('Use measurement of signal and (unscaled/scaled) signal noise')
spotMass1LOG = log10(LODspotMass1); % independent variable
sigSpotLOG = log10(LODsigSpot); % dependent variable

```

```

% Calculate noise floor for each frame/spot as mean of noise across all
% tracks (after scaling). + log10(lodcriteria) from setting LOD
% threshold to (lodcriteria)*stdevBlankScaled; log10(x*y) = log10(x) +
% log10(y) Cannot use mean(...) because this incorporates NaNs
stdevBlankScaledLOG = log10(LODstdevBlankScaled); % noise
noisefloorLOG1 = stdevBlankScaledLOG;
noisefloorLOG1(isnan(noisefloorLOG1)) = 0; % Duplicate with NaN-->0
% Count number of valid values to incorporate into average
noisefloorLOG2 = ~isnan(stdevBlankScaledLOG);
% Calculate average noise and scale by lodcriteria. (Note that if none
% of the tracks in the frame have valid peak fits for a given analyte,
% the calculated noisefloorLOG value is 0/0 == NaN
noisefloorLOG = sum(noisefloorLOG1,3) ./ sum(noisefloorLOG2,3) + ...
    log10(lodcriteria);
% now a nFrames x LODnSpots1 x 1 array

spotLOD = zeros(nFrames,LODnSpots1); % calculated spot LODs
sigSpotLOGFIT = zeros(nFrames,LODnSpots1,2); % linear regress. param.
for f = 1:nFrames
    for s = 1:LODnSpots1
        % Extract relevant data and make into column vectors
        spotMass1LOG1 = spotMass1LOG(s,:);
        sigSpotLOG1 = permute(sigSpotLOG(f,s,:),[3 1 2]);
        % Ensure that there are at least two real number ordered pairs
        % (spotMass1LOG1(i),sigSpotLOG1(i)). Create a test vector that
        % uses the fact that Real_number .* NaN = NaN
        LODdataTest = spotMass1LOG1 .* sigSpotLOG1;
        if length(LODdataTest(~isnan(LODdataTest))) >= 2
            % Remove ordered pairs with NaNs as one or both ordinates
            spotMass1LOG1 = spotMass1LOG1(~isnan(LODdataTest));
            sigSpotLOG1 = sigSpotLOG1(~isnan(LODdataTest));
            % Linear regress. (ind. = 1), permute to insert into array
            sigSpotLOGFIT(f,s,:) = permute( polyfit(spotMass1LOG1, ...
                sigSpotLOG1, 1), [1 3 2]);
            spotLOD(f,s) = (noisefloorLOG(f,s) - ...
                sigSpotLOGFIT(f,s,2)) / sigSpotLOGFIT(f,s,1);
            spotLOD(f,s) = 10 ^ spotLOD(f,s);
        else % Cannot perform fit or calculate LOD
            sigSpotLOGFIT(f,s,:) = NaN;
            spotLOD(f,s) = NaN;
        end
        clear spotMass1LOG1 sigSpotLOG1 LODdataTest;
    end
end
clear f s;
disp(' Calculated LOD for each spot in each frame.')

```

```

end
clear stdevBlank3D trkWidth3D intLengthpx3D noisefloorLOG1
noisefloorLOG2;
disp(['Saving data to ' opData ])
save([opDataPath opData]); % Save changes

%% Separation number
disp(' ')
disp('PART 6: SEPARATION NUMBER AND "REAL PLATE NUMBER"')
disp(' ')
disp(['Spot position and width data is available for ' ...
    num2str(nSpots1) ' spots.'])
disp('These spots have defined hue ranges of: ')
disp(hueRangeLabel1)
disp('On which spots (>=2) should the separation number calculation be')
disp('be based? Select only those that have reliable gaussian fits.')
confirm = 'n';
SNcalculate = ""; % Used to determine if SN should be calculated
while ~strcmp(confirm,'y')
    disp(' Enter a row vector to select spots or type "skip" (with ')
    SNsVect = input(' quotes) to skip separation number calculation: ');
    if isnumeric(SNsVect) && (length(SNsVect) >= 2)
        SNcalculate = 'y';
        confirm = input(' Confirm this selection (y/n): ','s');
    elseif strcmp(SNsVect,'skip')
        SNcalculate = 'SN not calculated';
        clear SNsVect;
        confirm = input(' Confirm this selection (y/n): ','s');
    else
        disp(' Invalid input or insufficient number of spots.')
    end
    disp(' ')
end
clear confirm;

% Calculate separation number using subset of data (if desired above)
if strcmp(SNcalculate,'y')
    % Select subset of available data NOTE: REMOVE VALUES NOT
    REQUIRED HERE LATER!!!
    % SNhueRange = hueRange1(SNsVect,:);
    % SNhueRangeLabel = hueRangeLabel1(SNsVect,:);
    % SNzSpot = zSpot(:,SNsVect,:);
    SNspotRF = spotHRF(:,SNsVect,:) / 100; % Scale back to range [0 1]
    % SNstdSpot = stdSpot(:,SNsVect,:);
    SNspotFWHM = spotFWHM(:,SNsVect,:);

```

```

% SNspotPN = spotPN(:,SNsVect,:);
% SNspotPH = spotPH(:,SNsVect,:);
% Perform fit of spot width vs. hRF data
SNfit = zeros(nFrames,nTracks1,2); % Initialize
for f = 1:nFrames
    for t = 1:nTracks1
        SNspotRF1 = SNspotRF(f,:,t)'; % Extract data to be fit
        SNspotFWHM1 = SNspotFWHM(f,:,t)';
        % Ensure that there are at least two real number ordered pairs
        % (SNspotRF1(i),SNspotFWHM1(i)). Create a test vector that
        % uses the fact that Real_number .* NaN = NaN
        SNdataTest = SNspotRF1 .* SNspotFWHM1;
        if length(SNdataTest(~isnan(SNdataTest))) >= 2
            % Remove pairs that have NaNs as one or both ordinates
            SNspotRF1 = SNspotRF1(~isnan(SNdataTest));
            SNspotFWHM1 = SNspotFWHM1(~isnan(SNdataTest));
            % Linear fit (ind = 1); permute to insert into array
            SNfit(f,t,:) = permute( polyfit(SNspotRF1,SNspotFWHM1,...
                1), [1 3 2]);
        else % Cannot perform fit
            SNfit(f,t,:) = NaN;
        end
        clear SNspotRF1 SNspotFWHM1 SNdataTest;
    end
end
% Calculate Separation Number (SN = zF / (b1 + b0) - 1) and
% N_real = 5.54 * (zF / (b1 - b0))^2; H_real = zF / N_real; where:
% b0 : FWHM extrap. to RF = 0 --> y-intercept: b0 = SNfit(f,s,2)
% b1 : FWHM extrap. to RF = 1 --> b1 = SNfit(f,t,1) * 1 + SNfit(f,t,2)
% b1 + b0 = SNfit(f,t,1) + 2 * SNfit(f,s,2)
% b1 - b0 = SNfit(f,t,1)
SN = zf1 ./ (SNfit(:,:,1) + 2 * SNfit(:,:,2)) - 1;
RealPN = sqrt(8*log(2)) * ( zf1 ./ SNfit(:,:,1) ) .^ 2;
RealPH = zf1 ./ RealPN;
disp('Calculated separation number "SN", real plate number')
disp('RealPN', and real plate height "RealPH" using method in')
disp('Thin-layer chromatography: a modern practical approach by ')
disp('P.E. Wall, 2005; Royal Society of Chemistry')
else
    disp('Calculation skipped. "SN","RealPN", "RealPH" not created.')
end
clear f t;
disp(['Saving data to ' opData ])
save([opDataPath opData]); % Save changes

% save([opDataPath opData]); % Save changes

```

```
disp('***SUCCESS SCRIPT FINALLY DONE!!!***')
end
```

A.2 Arduino Microcontroller Code

This code controls the timing of the LED's used to illuminate the development chamber. Each LED is turned 'on' for 84 ms and then 'off' for 18 ms before the next LED is powered.

```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {

  // define initial parameters
  int switch1 = 7;
  int redled = 4;
  int blueled = 2;
  int greenled = 3;

  pinMode(switch1, INPUT);
  pinMode(redled, OUTPUT);
  pinMode(blueled, OUTPUT);
  pinMode(greenled, OUTPUT);

  int switch1_value = digitalRead(switch1);
  if (switch1_value == HIGH){
    digitalWrite(redled, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(84); // wait for 84 ms
    digitalWrite(redled, LOW); // turn the LED off by making the voltage LOW
    delay(18); // wait for 18 ms

    digitalWrite(blueled, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(84); // wait for 84 ms
    digitalWrite(blueled, LOW); // turn the LED off by making the voltage LOW
    delay(18); // wait for 18 ms

    digitalWrite(greenled, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(84); // wait for 84 ms
    digitalWrite(greenled, LOW); // turn the LED off by making the voltage LOW
```

```
    delay(18);    // wait for 18 ms
  }
  else {
    digitalWrite(redled, LOW);
    digitalWrite(blueled, LOW);
    digitalWrite(greenled, LOW);
  }
}
```