Towards Personalized Medicine in Cystic Fibrosis: Patient-Specific Modeling of Mucociliary Clearance Using Physiologically-Based Flow Constraints

by

Monica E. Shapiro

B.S. Chemical Engineering, Lehigh University, 2017

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Monica E. Shapiro

It was defended on

May 24, 2023

and approved by

Carol A. Bertrand, PhD, Department of Pediatrics

Jason E. Shoemaker, PhD, Department of Chemical & Petroleum Engineering

Daniel G. Cole, PhD, Department of Mechanical Engineering & Materials Science

Dissertation Co-director: Robert S. Parker, PhD, Department of Chemical & Petroleum Engineering

Dissertation Co-director: Timothy E. Corcoran, PhD, Division of Pulmonary, Allergy, and Critical Care Medicine

Towards Personalized Medicine in Cystic Fibrosis: Patient-Specific Modeling of Mucociliary Clearance Using Physiologically-Based Flow Constraints

Monica E. Shapiro, PhD

University of Pittsburgh, 2023

Healthy humans have a thin layer of mucus lining the airways that protects the lungs from inhaled particulate or pathogens. Trapped particles are conveyed up the airway tree and out of the lungs by tiny, hair-like structures called cilia in a process called mucociliary clearance (MCC). Cystic fibrosis (CF) is a rare, lethal, genetic disease that dehydrates this mucus layer and disrupts MCC. This causes chronic infection and inflammation, leading to respiratory failure. Typical treatment for CF takes about 2 hours a day, creating a high treatment burden and leading to poor compliance. With the development of highly-effective modulators that target underlying defects of CF has come a push to reduce treatment burden. One time-consuming therapy that is widely used is aerosolized hypertonic saline (HS). HS aims to rehydrate airway mucus and improve MCC, however, the efficacy varies greatly between individuals. This creates a need for new screening tools to predict HS efficacy on a per-patient basis, which is the focus of this dissertation.

MCC in different sections of the airway tree varies, even within an individual. We thus developed a physiologically-based dynamic model of MCC that captured local variability within the lung. The granularity of the model enabled identification of focal defects, but had poor parameter identifiability. We reduced the number of free parameters to improve identifiability, while preserving the physiological constraints. The reduced model contained 5 free parameters and only increased the mean absolute error per grid by 8.7% from the original 114-parameter model. Finally, we fit this model to nuclear imaging data from CF participants on two separate days: one where they inhaled non-therapeutic isotonic saline and one where they inhaled HS. We developed a statistical model to estimate the parameter change after HS treatment for these participants based on clinical and *in vitro* measurements. The end result was a tool that can be used to estimate personalized MCC response of CF individuals to HS treatment.

Table of Contents

List of Tables

List of Figures

Preface

Five chapters later, and this was by far the hardest section of my thesis to write. How does one recognize all of the people who made this possible in just a few paragraphs? To the numerous grad students, faculty, and staff who will go unnamed, but who made Pitt not just a job, but a home – I have not forgotten you. Rather, I am grateful to have had so many friends and supporters to thank that I couldn't possibly list you all by name.

I first need to thank Bob, Tim, and Carol – who, of course, guided the technical side of the work, but more importantly, were a constant source of kindness and compassion when my inner critic threatened to overwhelm. Where others might have torn me down, you built me up – and for that, I am extremely grateful.

I was also supported by my incredible labmates, who embraced me from my first day to my last. From competing over our Game of Thrones fantasy league to designing lab beers, you taught me the importance of having fun, even in the midst of immense pressures. A special thank you to Michelle, who welcomed me whole-heartedly into her life the day we met. You have been a constant source of honesty, courage, and love, and though we are now separated by most of I-95, I know you will always be there for me and I for you. And I could not have done this without Megan, who taught me to find the positive in every situation. Your radical kindness and generosity continually inspire me to do better for myself and others.

Outside of work, my friends from Fleet Feet Run Club kept me grounded. They reminded my that, as important as every paper, abstract, and presentation seemed, there were more important things in life like family and friendship. And finally, speaking of family, I would not be where I am today without the love and support of my parents – the original Dr. Shapiros. From a young age, you instilled in me a love of learning that has never diminished and because of your love, I had the courage to fail. And fail. And fail again. Because I knew that you were there to catch me and prop me back up, so I could try (and probably fail) again.

1.0 Introduction

Cystic fibrosis (CF) is a rare, life-shortening genetic disorder that primarily impacts the lungs. There is typically a high treatment burden placed on people with CF and/or their caregivers, with the average CF individual spending just under 2 hours a day on treatments [1, 2]. The cost – both monetary and in terms of time – contributes to widespread non-adherence to medication schedules, which has been linked to worse long-term outcomes [2, 3]. In a 2022 survey, CF participants on average were willing to accept a 4.2 year decrease in life expectancy for large improvements in quality of life [4]. This has made reducing the treatment burden while maintaining the same standard of care a key research area in CF [5, 6].

One challenge faced when trying to reduce treatment burden is in determining which of the many medications CF individuals take are having the greatest or least impact. Additionally, this varies from person to person. It would therefore be useful to have tools for predicting efficacy of treatment on an individual basis. In this work, we will focus on developing a tool for one such treatment (hypertonic saline) and its impact on mucociliary clearance – a key process in maintaining lung health that is dysfunctional in CF.

1.1 Mucociliary Clearance (MCC) in the Airways

1.1.1 Airway Mucus

Fluid lining the airways is made up of several interacting layers, as shown in Figure 1. The epithelial tissue is made up of several cell types, including primarily ciliated cells, secretory cells, and basal cells [7]. Ciliated cells have hair-like structures that protrude from the cells at the apical surface. Secretory cells secrete the mucins necessary to form a gel-like mucus layer. Basal cells are stem cells that can differentiate into new epithelial cell types. Covering the epithelium is a fluid layer called the periciliary liquid layer (PCL), which is isotonic relative to plasma [8] and contains a number of mucins. Above the PCL is a gel layer of mucus. Current theory suggests that cilia interact with tethered mucins at the base of the mucus layer, which subsequently interact with the rest of the mucus layer (gel-on-brush) [7, 9]. Ion channels in the epithelial tissue help regulate the hydration of the PCL and mucus layers. The cilia beat in a coordinated pattern to sweep the mucus layer up the airways in a process called mucociliary clearance (MCC).

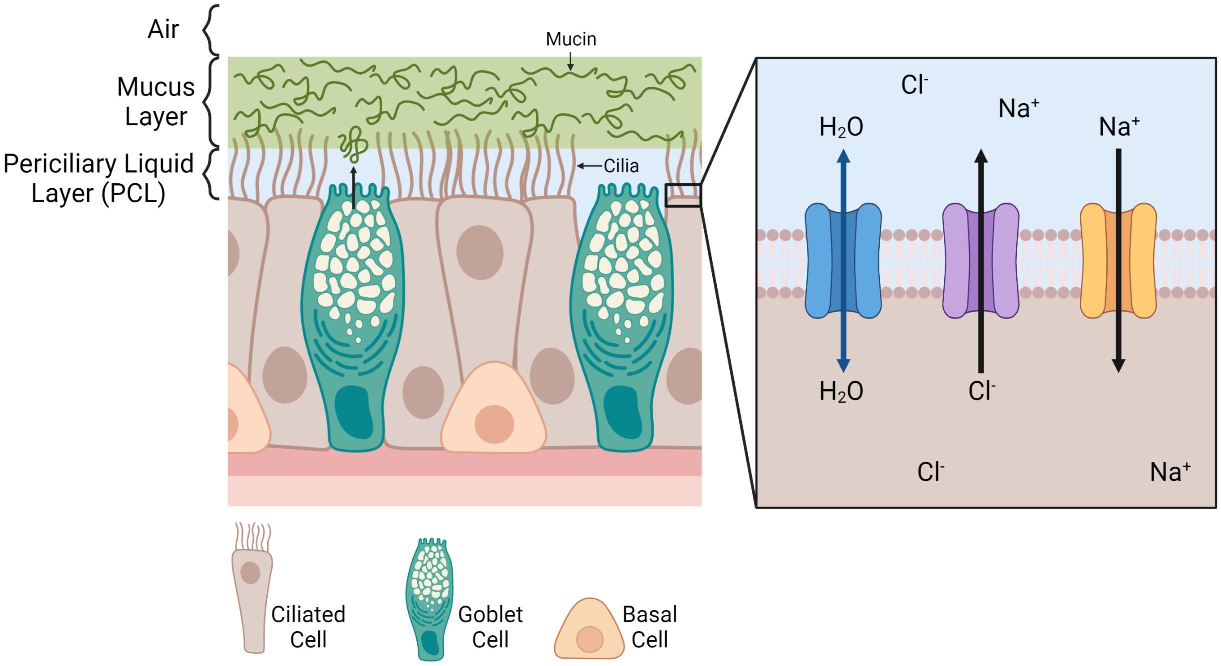


Figure 1: Simplified depiction of the different layers lining the airway epithelium: Epithelial tissue containing goblet cells, which secrete mucins, ciliated cells, which beat to push mucus up the airways, and basal cells, which can differentiate into other cell types as needed, the periciliary liquid layer (PCL), which provides a barrier between cells and mucus and facilitates beating of cilia, the apical membrane containing ion channels that regulate hydration of PCL, the mucus layer, which traps inhaled particles in a viscous gel

Disruption to one or many parts of these interacting layers, as occurs in many airway diseases [8, 10–13], can result in tissue damage. Over secretion of mucins could result in a denser, more viscous mucus layer, which would exert more compressive force on the cilia.

This in turn could result in less effective MCC. Similarly, dehydration of the PCL and mucus layers can lead to over compression or even complete collapse of cilia caused by osmotic forces, also resulting in reduced MCC capacity. Stagnant mucus means that pathogens or other inhaled debris trapped in the mucus layer remain in the airways for longer. The subsequent prolonged inflammatory response can cause damage to the tissue.

1.1.2 Airway Structure

The airways have a branched structure, as depicted in Figure 2. Air enters the lungs through the trachea, which splits into two main bronchi - one leading into each lung. These split into further bronchi (3 on the right, 2 on the left), leading to each of the lobes of the lung. From there, the airways continue to branch into narrower airways that are larger in number. Work from Weibel and Gomez [14] estimates the airways branch an average of 23 times from the trachea before terminating at the alveoli. The alveoli are comprised of high surface area "sacs", which allow for rapid gas exchange with the capillaries surrounding them. Here, oxygen is absorbed into the blood stream and carbon dioxide diffuses into the airways. Air is expired following the reverse path, where air flows from small airways into fewer larger airways back up the airway tree.

Mucus is secreted along the length of the conducting airway tree (excluding the alveoli). Mucus is pushed by cilia up the airway tree from many smaller airways into fewer larger airways, all the way up the tree. The alveoli do not have any ciliated cells, and thus do not participate in MCC. The density of ciliated cells is higher in the larger airways [15], which allows MCC to occur much more quickly and thus keep up with the rates flowing in from smaller airways. In muco-obstructive diseases, such as cystic fibrosis and COPD, chronic inflammation from decreased MCC in the small airways contribute to the deformation of the airway walls in a process called bronchiectasis. This, in turn, leads to further accumulation of mucus in the small airways. In the next section, we will take a closer look at the cause of this in the case of cystic fibrosis.

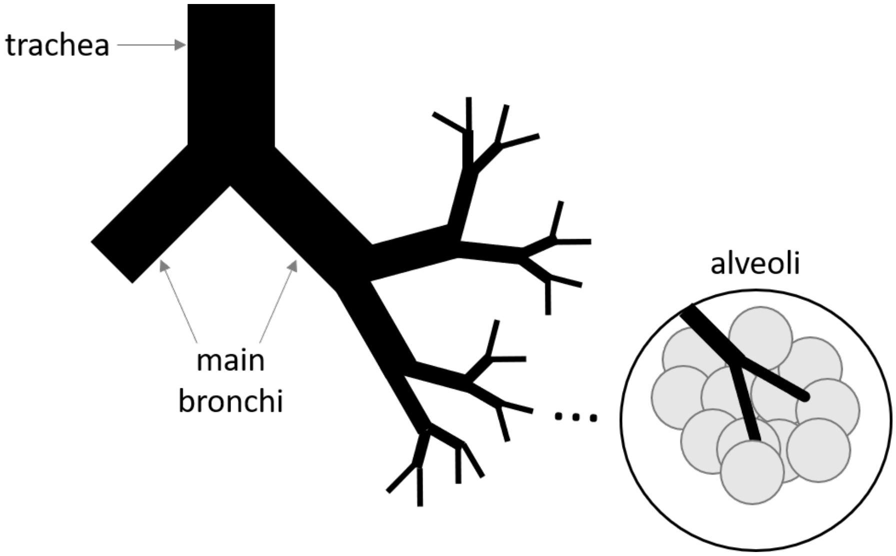


Figure 2: Depiction of the airway structure. The trachea branches into two main bronchi, leading to each lung. From there, airways continue to branch, getting smaller in size and larger in quantity until they terminate into alveoli. Airways become less ciliated as they get smaller. There are no cilia in the alveoli.

1.2 Cystic Fibrosis Background

Cystic fibrosis (CF) is a lethal, autosomal recessive disease that affects over 70,000 people worldwide [16]. It is caused by over 2,000 known mutations [17] affecting the cystic fibrosis transmembrane conductance regulator (CFTR) gene, which codes for an anion channel at the epithelial surface of the same name [16]. Though it impacts all epithelial tissue and affects multiple organs, the primary cause of death is due to respiratory failure [18]. Predicted survival rates in the U.S. have increased dramatically over the past decade with a median expected survival age of 53.1 years for those born between 2017 and 2021. This is due in part to the approval of highly effective CFTR modulators for 90% of the population over the age of 11 [18]. Though this is a vast improvement over the predicted age of 38 for those born between 2007 and 2011, it is still a long way off the median life expectancy in the U.S. of ∼79 for those born over the same time frame [18, 19].

1.2.1 Airway Electrophysiology

CFTR is an ATP-binding cassette transporter that functions as an epithelial anion channel [20, 21]. Mutations to this gene directly affect transport of anions, such as chloride and bicarbonate, across the membrane causing osmotic imbalances that lead to hyperabsorption of airway surface liquid and increased acidity of that liquid [21, 22]. They also cause a cascading dysregulation of other ion channels and water transport. Evidence suggests that sodium permeability through the epithelial sodium channel (ENaC) is significantly increased in CF [23, 24] and the permeability of chloride through constitutively active chloride channels (i.e. not CFTR) is significantly decreased [23, 25, 26]. Additionally, there is evidence of increased transcellular water permeability, possibly through increased aquaporin 3 activation [23, 27]. Due to the complexity of interactions, there is still much that is poorly understood in terms of how CF impacts the function of other ion channels and transporters. However, it is clear that the combination of downstream dysregulation results in hyperabsorption of fluid from the airway surface.

1.2.2 Mucus Thickening and Accumulation

CF is a muco-obstructive disease, meaning that chronic airway inflammation causes an accumulation of mucus. This can be attributed to multiple factors. Defective CFTR, as well as dysregulation of other epithelial ion channels, causes hyperabsorption of liquid from the mucus and PCL, compressing or collapsing the cilia. Futhermore, evidence suggests that those with CF produce more of the mucin, MUC5AC, relative to MUC5B than in healthy controls [28]. MUC5AC is thought to be a "stickier" mucin and thus increases the viscosity of the mucus layer making it harder to clear. Slow or stagnant mucus can lead to a prolonged inflammatory response when encountering a pathogen. This immune response can result in a build-up of DNA and actin from degraded neutrophils [29]. In turn, this increases the viscosity of the mucus layer, further deacreasing MCC. The inflammation can also cause tissue damage and thickening of the airway walls, thus reducing overall pulmonary function.

1.2.3 Current Treatment Paradigm & Challenges

Treatment of CF is complex and often personalized to an individual. Due to the rarity of the disease, clinical trials are necessarily small and thus do not have the same statistical power to detect changes in response to new therapies that large-scale clinical trials typically do. Further, there is a lot of heterogeneity in the population. Even for those with the same mutations, disease progression and individual response to treatment varies greatly. Drug efficacy in these small-scale clinical studies can be confounded by subgroups of responders or non-responders. This presents a challenge in determining clinical best practices on a broad scale.

Typical treatment involves a multifaceted approach to the underlying defects as well as the host of resulting symptoms. CFTR modulator drugs work by directly affecting the amount and function of CFTR on the epithelial surfaces. Mucolytic drugs and airway clearance techniques help to reduce the accumulation of mucus in the airways. Antibiotics are often prescribed to treat infection, and pancreatic enzymes help treat gastrointestinal dysfunction that can lead to malnutrition.

Unfortunately, this often translates to a high treatment burden on CF individuals and/or their caregivers [1, 2], which can lead to decreased quality of life and compliance with prescribed therapies [2, 3]. In particular, inhaled therapies require substantial time to prepare, administer, and clean equipment. About 90% of CF adults take at least one inhaled therapy, with about 30% of individuals taking three or more [18]. With highly-effective CFTR modulators available for a large portion of CF individuals, it is likely that quality of life will play a more important role in future CF treatment than it has previously. Thus, there is a great need for better tools to understand which therapies will provide the most benefit on an patient-specific basis.

1.2.3.1 Modulators

CF mutations have historically been broken down into 5 classes as described in Table 1. Some of these classes still have some residual function, while others are more severe. Modulator therapies aim to restore some or all of this function for specific mutations or classes.

Table 1: Classes of CFTR Mutations

Class	Mutation Type	Description
1	Protein production	Mutation prevents any functional CFTR from being transcribed; Often severe
2	Protein processing	CFTR is transcribed, but mutations cause misfolding of the protein; Includes most common CF mutation, F508del
3	Gating	Prevents CFTR channel from opening upon activation
4	Conduction	CFTR is formed, but mishaped channel slows chloride transport; Often retain residual function
5	Insufficient protein	Not enough protein is produced, or it is degraded too quickly

Currently, there are four CFTR modulators approved in the U.S. for CF individuals with certain mutations. Ivacaftor is a potentiator, meaning it holds the channel of CFTR open, allowing for increased transport of chloride into the PCL. This can be used alone for those with at least one gating mutation, or in combination with one or more corrector – drugs that help CFTR fold correctly – for certain mutations, including F508del.

For those who are eligible, modulator therapies have become a staple in the treatment of CF. Prior to the approval of the triple combination therapy of elexacaftor, tezacaftor, and ivacaftor in 2019, fewer than 11,800 individuals with CF (of the over 30,000 in the CF Foundation Patient Registry) were prescribed a CFTR modulator. Since then, modulator use has increased to about 22,300 individuals, about 86.3% of whom are on the triple combination therapy [18]. Early indications suggest this could greatly extend life expectancy and improve long term lung health, however, it is too soon to gauge the full impact it will have. Additionally, while most people with CF have two alleles that can be affected by CFTR modulators, about 10% of CF individuals have mutations that are ineligible for any CFTR modulator [18]. Those would be individuals with a stop mutation (Class 1 – no CFTR transcribed) and a second allele that does not produce any functional CFTR (Class 1 or 2).

1.2.3.2 Mucus clearance therapies

Another priority area of treatment is in mucus clearance. Inhaled therapies focus on decreasing the viscosity of mucus in order to make it easier for cilia to clear. Most CF individuals (∼88%) are prescribed dornase alfa [18], which is a mucolytic agent that breaks down DNA polymers within the mucus to decrease the viscosity. Osmotic therapies, such as hypertonic saline (HS) and mannitol work by creating an osmotic driving force and drawing water to the airway surface. This rehydrates both the PCL and mucus layers to improve MCC. HS is widely used in the U.S. with about 70% of CF individuals taking it [18]. One challenge with both dornase alfa and HS is that it must be delivered as an aerosol via a nebulizer. This takes a while to prepare, deliver, and clean properly, which can decrease compliance [1].

Nearly all CF individuals are also recommended to perform airway clearance techniques (ACT) [18]. Individuals typically do this shortly following inhalation of mucolytics, such as dornase alfa and HS to create a synergistic effect. Most commonly, patients use a high-frequency chest wall oscillation vest [18] that mechanically inflates and deflates the chest to help separate mucus from the airway walls and into larger airways. About half of adults also use aerobic exercise as a secondary form of airway clearance. In adults over 30, about 20% of individuals use aerobic exercise as their primary form of ACT [18].

1.2.3.3 Microbiology

Infection is a constant source of concern with CF. Acute infections often require intravenous antibiotics to get pulmonary exacerbations under control. Even afterwards, however, many infections are not completely cleared and result in chronic infection. The most common microorganisms found are *Staphylococcus aureas* and *Pseudomonas aeruginosa*. For those who develop chronic *P. aeruginosa* infections, inhaled antibiotics, such as tobramycin and aztreonam, or oral azithromycin are commonly prescribed. Some individuals, especially those with frequent exposure to antibiotics develop antibiotic resistant strains.

1.3 Measuring MCC *In Vivo*

Quantifying MCC at the organ-scale *in vivo* is non-trivial. It is typically measured via planar scintigraphy, in which subjects inhale radiolabeled aerosol and then sequential 2D images of the activity in the lungs are captured with a gamma camera. A large, gamma emitting, non-absorbable particle is delivered via a liquid aerosol. Because it cannot be absorbed, any 2D motion of the particles that deposit in the lungs (some aerosol is swallowed) must be associated with MCC. Typically, only the right lung is used for analysis to avoid interference from particles that have been swallowed and are in the stomach. The change in activity in the whole right lung can then be used as a biomarker for MCC.

There are several challenges that naturally arise with this type of measurement. The first is that there is limited ability to control where the aerosol deposits. Although different breathing patterns can be used to target aerosol deposition to particular airway classes, there is still a lot of variability in the patterns observed between different individuals. This is key because the larger, more proximal airways have been shown to clear mucus more quickly than distal airways [15]. Previous work has shown an association between deposition pattern and measured whole lung MCC [30], which hinders the utility of the measurement for quantifying nominal vs abnormal behavior.

Attempts have been made to account for deposition-based differences between individuals. It is common to break the whole lung area from these nuclear images into two separate regions of interest (ROIs): a central ROI, which is defined as a rectangle half the height and width of a bounding box around the whole lung area that is placed at the medial edge, and a peripheral ROI, which consists of the remaining area in the whole lung [10, 31–33]. An example of these ROIs is shown in Figure 3. This can then be used to determine if there are significant differences in deposition between groups. Alternatively, it can be used to adjust MCC measurements based on the proportion of activity deposited in each ROI relative to the population average [34].

While these provide better comparison between individuals, they only provide a coarse measurement. There is a lot of heterogeneity within each ROI, which leaves a lot of room for deposition to confound measurements. Additionally, many of those with CF have mucus plugging, which can result in focal defects. Since these are lumped in with the remainder of the ROI in which they are located, their impact could easily be missed.

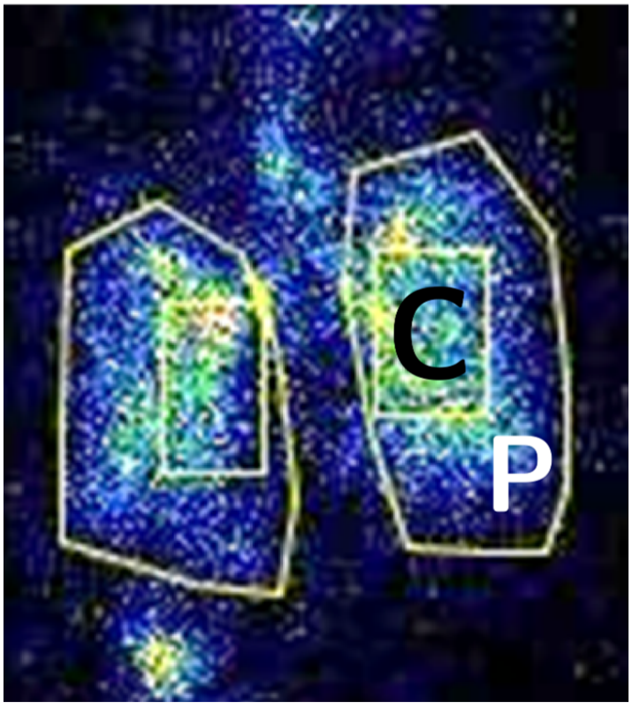


Figure 3: Example of standard regions of interest (ROIs). A whole lung outline is drawn. The central ROI (indicated C) is defined as a rectangle of half the height and width of a box bounding the whole lung outline, centered at the medial edge. The peripheral ROI (indicated P) is defined as the remainder of the whole lung area. Radioactive counts are measured within each ROI to measure MCC.

Another challenge with the use of planar imaging is that it is a 2D projection of a 3D system, meaning that there is an inevitable loss of information. Although there are other nuclear imaging methods that can reconstruct an image of the lung in 3D from a series of 2D images at different angles (i.e. single photon emission computed tomography), the time needed to reconstruct a single 3D image (approximately 15 min) is much longer than the time scale of MCC from the proximal airways and thus, is not suited for this purpose. Without this 3rd dimension, however, it is impossible to know where along the airway tract particles deposit. For example, activity that overlays the main bronchus in 2D may actually be in the alveoli. This means that the standard central ROI can include rapidly clearing large airways, slow clearing small airways, and alveoli, where there is not MCC. Depending on where the emitting particles actually are along the airway tract, this can confound typical MCC measurements.

1.4 Existing *In Silico* Models of MCC

In silico models can provide insight into the physiology underlying MCC changes observed *in vivo*. The structure of the model is based on physiology, so the fitted parameters can get at underlying processes, rather than just what can be measured directly. Previously, our group developed a population level dynamic model of MCC and paracellular airway surface liquid absorption (ABS) at the lung-scale[35]. The model structure is depicted in Figure 4, where each arrow corresponds to a first-order kinetic process and parameters were fit to planar imaging similar to what was described in the previous section. A key difference in this study, however, was the addition of a second gamma-emitting particle that was small enough to be absorbed paracellularly. The second particle could then be used as a biomarker for ABS. These probes were inhaled together and were measured simultaneously using different radioactive energy windows. The larger, non-absorbable particle that was used to measure MCC was Technetium-99m sulfur colloid (Tc-SC). Indium-111 diethylene triamine pentaacetic acid (In-DTPA), was used as the smaller second particle and could be cleared both by MCC and absorbed paracellularly. ABS was estimated by calculating the difference in retention between these two probes.

Parameters from the model were fit in two steps: first, the fraction of functional ciliated area in the central ROI ($FFCA$) and an MCC rate coefficient from the functional large airway (k_{LF}) were fit to Tc-SC retention data, setting ABS rate coefficients (k_{LB} and k_{DB}) to 0. The MCC parameters were fixed and then the ABS rate coefficients were fit to the

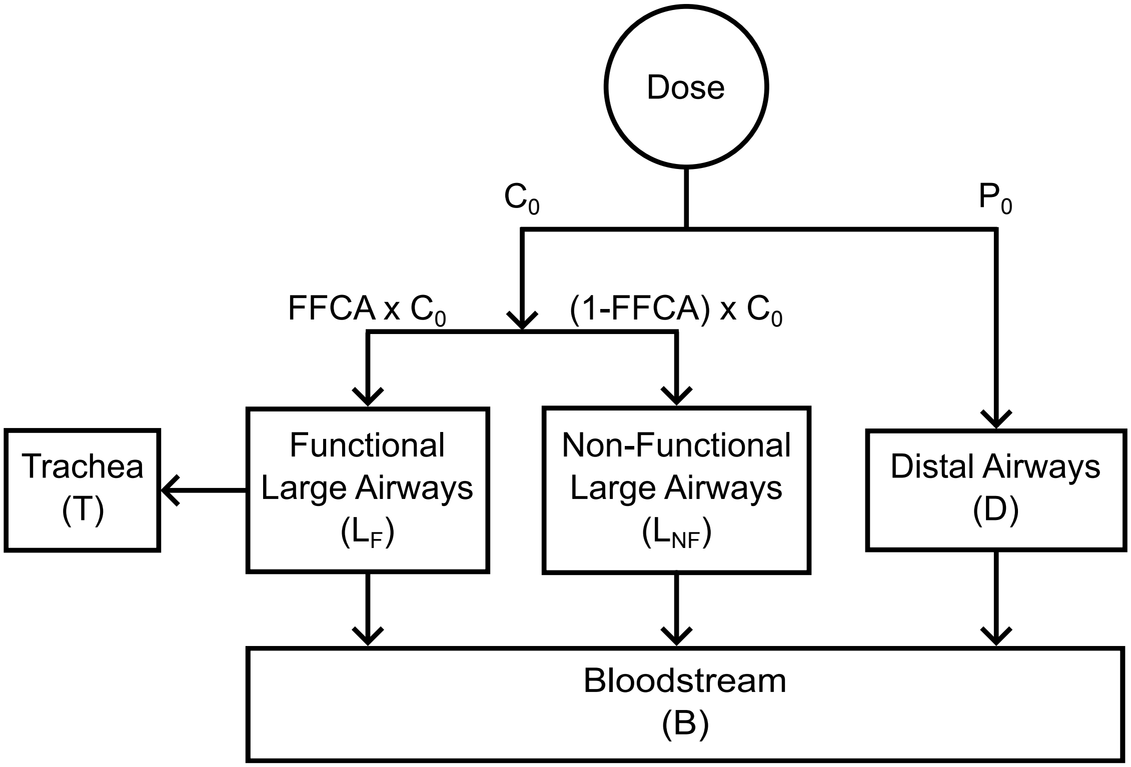


Figure 4: Compartmental model developed in [35]. Initial conditions are determined from deposition in each ROI (C_0, and P_0 for the central and peripheral ROIs, respectively) and a fitted parameter, fraction of functional ciliated area ($FFCA$). Technetium-99m sulfur colloid (Tc-SC) and indium-111 diethylene triamine pentaacetic acid (In-DTPA) both clear from the functional large airways (L_F) to the trachea following first-order kinetics. In-DTPA is also absorbed into the bloodstream from each airway compartment following first-order kinetics.

In-DTPA retention data. For parameter identifiability reasons, MCC from the peripheral ROI into the central ROI was assumed to be negligible and set to 0. While this model provided a good description of the dynamics observed at the population level, some of the model assumptions broke down when applied to individuals.

Specifically, two assumptions were shown to be inaccurate for a number of individuals: 1) negligible clearance from the peripheral ROI into the central ROI, and 2) well-mixed activity within the ROIs. As shown in Figure 5, for some subjects, there is substantial MCC from the peripheral ROI over the study window. Remedying the problem is not as simple as fitting an MCC rate coefficient from the peripheral to central ROI because the activity is not evenly distributed, as shown in Figure 3. This means that activity that is more distal must be transported across the entire peripheral ROI before it can be cleared into the central ROI. Thus a compartmental model that treats the peripheral ROI as one compartment will never be able to accurately capture the dynamics observed. Therefore, we sought a more detailed model that would better capture individualized response and peripheral-to-central MCC while maintaining parameter identifiability.

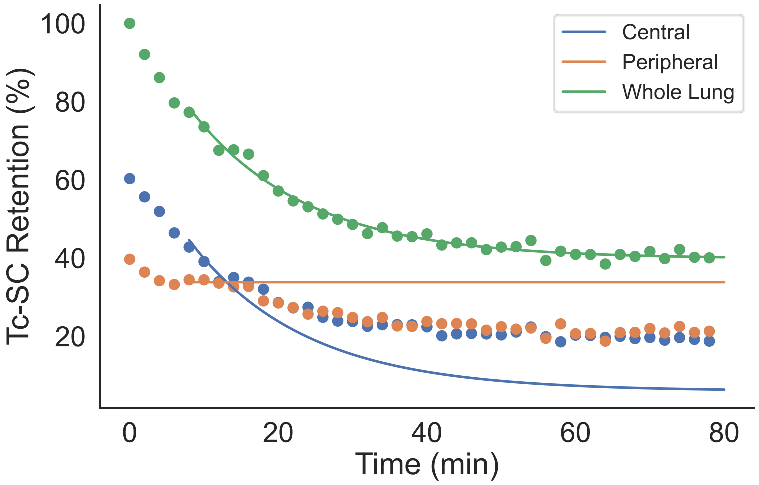


Figure 5: Comparison of experimental versus simulated technetium-99m sulfur colloid (Tc-SC) retention in whole lung, central, and peripheral ROIs for an example healthy control using the model from [35]. Circles represent experimental values and solid lines represent simulations of best-fit parameters.

1.5 Dissertation Overview

Through the course of this dissertation, we will develop an *in silico* model of MCC that captures localized differences and use it to characterize patient-specific response to inhaled HS. In **Chapter 2**, we develop a physiologically-based model of MCC that describes the flow of radiolabeled particles in nuclear imaging studies. We solve the multi-objective optimization problem this poses, allowing us to capture localized differences in MCC across subjects. This results in a high dimensional parameter space. Due to variability in the deposition of the radiolabeled particles, some of these parameters lack sufficient data to have confidence in their values. In **Chapter 3**, we take steps to rationally reduce the degrees of freedom in our model by simplifying flow constraints and lumping appropriate parameters. This enables us to use a stronger signal to inform each of our free parameters. With this model in hand, we apply it in **Chapter 4** to nuclear imaging studies of CF participants inhaling two different aerosols: i) isotonic saline (IS), which has been shown to be non-therapeutic, and ii) hypertonic saline (HS), which has been shown to have a therapeutic effect on MCC of varying degrees. Using this local description of MCC, we can capture patient-specific response to acute HS treatment. Taking non-invasive measurements from the same participants, we then build a statistical model to estimate response without the need for nuclear imaging. Finally, in **Chapter 5**, we discuss the short- and long-term impacts of this work on studying MCC in CF and other airway diseases.

2.0 Development of a Physiologically-Based Model of Airway Mucociliary Clearance and Paracellular Surface Liquid Absorption

2.1 Introduction

As discussed in the previous chapter, we are interested in developing a more granular model of MCC in the airways that can capture localized differences between individuals. While the focus of this work is on applications to CF, focal MCC defects may be important to other muco-obstructive airway diseases, such as COPD and asthma. In this chapter, we build a mathematical model describing MCC of radiolabeled isotope in 2D based on the use of planar images of the measurement technique. To keep the model generic, and thus increase future applicability to other diseases, we focus in this chapter only on healthy individuals. This allows us to use average airway anatomy to define mucus flow in a physiologically meaningful way.

2.1.1 System Definition

In order to make the system more granular, we wanted to break the nuclear images down into smaller ROIs. Previous work has done static analysis on particle retention on a pixel-by-pixel basis [36]. This represents one extreme form of granularization, where whole lung analysis would be the other extreme. Because study participants occasionally shift slightly over the course of imaging, we felt a pixel-by-pixel analysis would be too susceptible to these disturbances. As described in Chapter 1, whole lung or even central and peripheral ROIs are not granular enough to capture localized defects. Instead we chose to define our ROIs, so they were somewhere between these two extremes. To do that, we divided the right lung into a 16 × 8 (rows × columns) grid mesh. We selected this grid size, so that we could draw direct comparisons with traditional ROIs (see Figure 3 in Chapter 1). Specifically, rows 5-12, columns 1-4 correspond to the traditional central ROI, while the remaining grids within the lung area correspond to the peripheral ROI. From there, we derived flow constraints based on airway geometry to describe flow of mucus into and out of each of these grids.

We chose to use the geometric mean between the posterior and vertically mirrored anterior images. While the posterior view is commonly used for MCC studies [35, 37], recent guidelines from the International Society for Aerosols in Medicine [38] recommend using the geometric mean, where possible, for determining aerosol deposition. We opted to use it for MCC analysis as well because it can be shown mathematically that it is less sensitive to particle motion into and out of the imaging plane.

The intensity measured by the gamma camera ($I(r)$) is proportional to the inverse of the squared distance between the camera and particle (r^2), as shown in Equation 2.1.

$$I(r) \propto \frac{1}{r^2} \tag{2.1}$$

We can express the intensity measured by each camera as follows:

$$I_p\left(r\right) = \frac{\beta}{r^2} \tag{2.2}$$

$$I_a\left(r\right) = \frac{\beta}{\left(R-r\right)^2} \tag{2.3}$$

Where I_p and I_a are the measured posterior and anterior intensities from a particle, r is the distance of the particle from the posterior camera, R is the distance between the two camera heads, and β is a proportionality constant. The geometric mean of the measurements is thus:

$$I_{gm}\left(r\right) = \sqrt{I_p I_a} = \frac{\beta}{r\left(R-r\right)} \tag{2.4}$$

To understand how these intensities are affected by motion into and out of the imaging plane, we can compare the derivatives with respect to distance to the posterior camera, r. We will consider the posterior intensity and the geometric mean of the anterior and posterior images, however, this easily could be applied to the anterior intensity by simply flipping which camera is used as a reference (i.e. define r as the distance to the anterior camera instead).

$$\frac{dI_p}{dr} = \frac{d}{dr}\left(\frac{\beta}{r^2}\right) = \frac{-2\beta}{r^3} \tag{2.5}$$

$$\frac{dI_{gm}}{dr} = \frac{d}{dr}\left(\frac{\beta}{r\left(R-r\right)}\right) = \frac{-\beta\left(R-2r\right)}{r^2\left(R-r\right)^2} \tag{2.6}$$

We can then show that $|\frac{dI_p}{dr}| > |\frac{dI_{gm}}{dr}|$, indicating that the posterior intensity decreases more than the geometric mean, as the distance from the posterior camera increases. Note: $\beta > 0$ and $r > 0$.

$$\left|\frac{-2\beta}{r^3}\right| > \left|\frac{-\beta\left(R-2r\right)}{r^2\left(R-r\right)^2}\right| \tag{2.7}$$

$$\frac{2}{r} > \frac{R-2r}{\left(R-r\right)^2} \tag{2.8}$$

$$1 > \frac{r\left(R-2r\right)}{2\left(R-r\right)^2} \tag{2.9}$$

We can then re-express this in terms of an anterior distance, r_a and a posterior distance, r_p.

$$1 > \frac{1}{2}\left(\frac{r_p}{r_a}\right)\left(\frac{r_a - r_p}{r_a}\right) \tag{2.10}$$

Considering each set of parentheses, we can see that for all cases, Equation 2.10 is True. When $r_a > r_p$, both sets of parentheses are less than 1, therefore the entire fraction is less than 1. When $r_a = r_p$, the fraction in last parentheses goes to 0, which is less than 1, and when $r_a < r_p$, the last fraction becomes negative and thus less than 1.

2.1.2 Many-Objective Optimization

To track the radiolabeled particles at a more granular level as they clear from smaller airways towards the trachea, we break the nuclear images into a 2D grid, where mucus (and thus particulate) flows between neighboring grids over time. This turns what was a single objective function in Markovetz et al. [35], comparing simulated and experimental whole lung activity, into a multi-objective problem, where simulated and experimental activity are compared for each grid in the network. Ideally, we would like to find a solution that minimizes each of the objectives. Challenges arise, however, when the objectives conflict (i.e. an improvement to one objective causes a worse solution for another objective). That begets the question, of how to best balance trade-offs in objective values between grids.

For multi-objective problems, a solution is considered dominant over another if the solutions are no worse for each individual objective and better for at least one. The set of all non-dominated solutions form what is called a Pareto front [39]. For small, finite parameter spaces, the set of solutions that form this front can be found in time $O\left(n\left(\log n\right)^{d-2}\right)$ for $d = 2$ or $O\left(n\left(\log n\right)^{d-2}\right)$ for $d \geq 3$, where n is the number of objectives and d is the number of parameters [40]. When the parameter space to search is larger, or the variables are continuous, however, it can quickly be infeasible to precisely calculate this front.

For these cases, the solution method depends greatly on the specific problem. For problems with continuous variables, but a small number of objective functions, there are numerous evolutionary algorithms that can be implemented based on the ultimate goal of the multi-objective optimization [41, 42]. When there are more than 3 objectives, however, these methods tend to fail to converge due to the large number of non-dominated solutions [39, 41]. In these instances, scalarization methods are typically required, in which the objectives are combined or reformulated as constraints. This reformulated, possibly constrained, single-objective problem can then be solved using traditional optimization tools [39, 41]. In our case, we have 114 grids within the whole lung area, corresponding to 114 independent parameters and 114 objective functions. As such, we will explore several different schemes for scalarization of the objective functions for each grid.

2.1.3 Paracellular Airway Surface Liquid Absorption (ABS)

While we are primarily interested in studying MCC, there is some evidence that the rate of paracelluar airway surface liquid absorption (ABS) can predict response to inhaled HS in CF individuals [23], particularly in the peripheral ROI. Thus as a secondary outcome, we are interested in exploring localized differences in ABS. To do this, we use two radiolabeled probes: one that can only be cleared by MCC and one that can be cleared by MCC and absorbed paracellularly. Calculating the difference in retention between the two probes, we can get a measure of ABS.

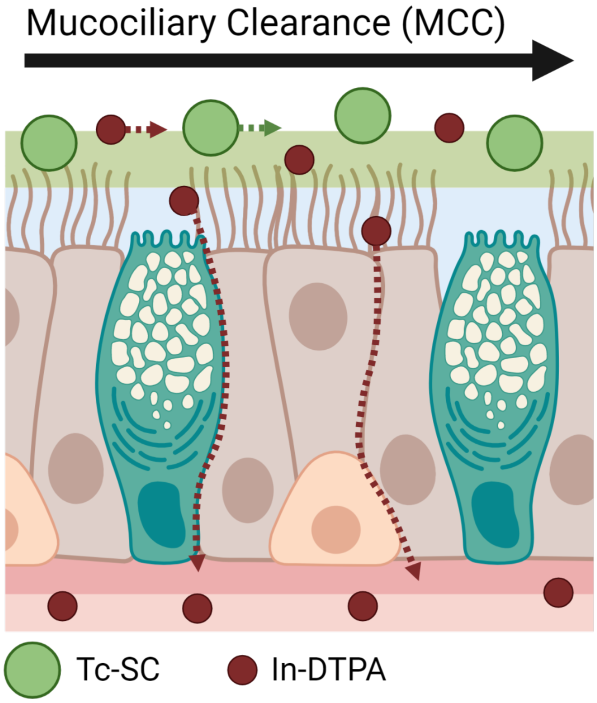


Figure 6: Depiction of two-probe nuclear imaging study. Technetium-99m labeled sulfur colloid (Tc-SC) only clears through MCC. Indium-111 labeled diethylene triamine pentaacetic acid (In-DTPA), which is much smaller, clears through MCC and is absorbed paracellularly.

2.2 Methods

2.2.1 Two-Probe Nuclear Imaging Study

Study subjects inhaled two radiolabeled probes – Technetium-99m labeled sulfur colloid (Tc-SC) and Indium-111 labeled diethylene triamine pentaacetic acid (In-DTPA) – prior to 80 minutes of sequential gamma camera imaging. All subjects had a study day where they inhaled isotonic saline from 10-20 min during the study (images were collected throughout inhalation). CF subjects came in for an additional study day, where they inhaled hypertonic saline. The order of interventions was randomized for CF subjects. Demographic information of this study can be found in Table 2. It should be noted that all CF participants were enrolled prior to the approval of the ivacaftor/tezacaftor/elexacaftor combination CFTR modulator. As shown in Figure 6, Tc-SC is too large to be absorbed and is thus used as a surrogate measure of MCC. In-DTPA is cleared both through MCC and ABS. Using the normalized difference between Tc-SC and In-DTPA retention, we can estimate a measure of paracellular fluid absorption (ABS).

2.2.2 Image Processing

The images were initially processed using ImageJ (1.52v). An anatomically-based whole lung region of interest (ROI) from [43] was stretched and aligned to fit the right lung of a posterior transmission scan for each subject. This ROI was aligned to the observed activity in the right lung of the posterior view and the mirror image of the anterior view of the nuclear images. Scan images were rotated as needed to best fit the ROI to the right lung outline. The images were clipped to a bounding box around this ROI for all time points, and the intensities for each pixel were exported as text images (tab-delineated text file of the intensities). A full protocol for processing these images in ImageJ can be found in Appendix A.

An automated Python script was written for the remainder of the image processing, and is available on GitHub (`https://github.com/monshap/psanalysis`). Background activity was subtracted from each image. Decay corrections for each of the probes were calculated

Table 2: Demographics of included study participants with lung clearance index (LCI) and pulmonary function data (mean±std). For correctors in CF group, I=ivacaftor, L/I=lumacaftor/ivacaftor, and T/I=tezacaftor/ivacaftor.

		Controls (HC)	**Cystic Fibrosis (CF)**		
Number of Subjects		11	23		
Age (years)		22 ± 3	30 ± 14		
Female/Male		4/7	13/10		
Corrector		N/A	**I**	**L/I or T/I**	**None**
			4	7	11
LCI		7.3 ± 0.89 (n=10)	11 ± 2.9 (n=20)		
			IS Day	**HS Day**	
Pulmonary Function Tests (% predicted)	**FEV$_1$**	102 ± 12	69 ± 25	69 ± 25	
	FVC	106 ± 13	84 ± 22	80 ± 25	
	FEF$_{25\text{-}75}$	88 ± 20	48 ± 36	48 ± 33	

within the gamma camera used for imaging. From there, the geometric mean of the posterior and mirrored anterior images was calculated to mitigate motion along the posterior-anterior axis. These cropped images were then divided into 16 x 8 (l x w) equal sized grids. This grid layout was chosen to allow for direct comparison with literature values for central and peripheral ROIs, where the central ROI would correspond to rows 5-12, columns 1-4. The activity in each grid was then divided by the total activity initially deposited in the whole lung to normalize for differences in total deposition between participants. Subjects without a total initial activity of at least 500 were excluded. Only one participant was excluded for this reason.

To assess the heterogeneity within the grids, relative to traditional ROIs, the variance and skew of pixel intensities were calculated across individual grids in each of the ROIs (central and peripheral) and for the whole ROI. A homogeneous ROI or grid should have a low variance and skew close to 0. The more homogeneous the ROI/grid, the more accurate a model treating that unit as a well-mixed compartment can perform.

2.2.3 2D Model Development

2.2.3.1 Regions with no MCC

Not all of the Tc-SC was cleared by the end of 80 min of imaging. This could be due to a few different factors. The first is simply that the time-scale over which it could clear from the lung is longer than the length of the study. This is likely the case for some of the observed activity in larger airways since there are a large number of smaller airways feeding into these that clear at slower rates. A second reason is that some of the aerosol deposits in alveoli, which don't have cilia and thus cannot clear Tc-SC via MCC. Finally, particularly for CF participants, some aerosol lands on dehydrated mucus that inhibits MCC and thus is not cleared over the course of the study.

In the population level model from [35], this was taken into account by fitting a parameter, fraction of functional ciliated area ($FFCA$) simultaneously with the clearance rate coefficients. However, this contributed to the identifiability challenges that led to the assumption of no MCC in the peripheral region. To avoid this, but still account for stagnant particles, the lower 25% quartile of Tc-SC activity in each grid was assumed to be non-clearable and subtracted from the overall activity. The dynamic MCC model was then only fit to the remaining, "clearable" activity.

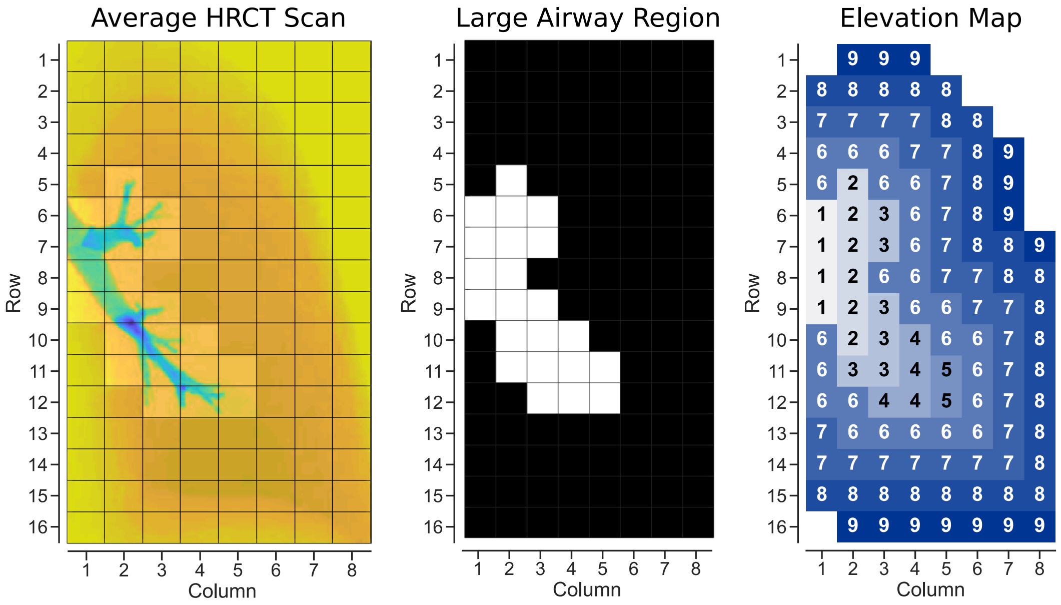
2.2.3.2 Regions with MCC

Next, we modeled the flow of mucus (and thus Tc-SC) between neighboring grids. This required constraints on where mucus could flow in 2D and we wanted to base these around the known physiology. To do this, we used a 2D anterior view of anatomically-averaged high resolution computed tomography (HRCT) scans from [44]. We mirrored the image and divided the right lung into the same 16 x 8 grid as our nuclear imaging scans, as shown in Figure 7. From here, there were 4-5 generations (branch points from the trachea) that could be visibly distinguished. We defined a large airway region (LAR) that consisted of any grid overlaying one or more of these visible airways, as shown in Figure 7. Physiologically, we know that mucus within the further generations of the large airways progressively feed into larger and larger airways until it is cleared to the trachea. To mimic this process, we defined flow constraints within the LAR, such that mucus could flow to neighboring grids that were closer or equal distance (measured as a city-block distance, allowing diagonals) from the left edge. Because the airways are too small to distinguish outside this region, we assume that mucus flows to neighboring grids in the direction of the LAR since we know it must pass through the LAR to clear from the whole lung. To capture all these constraints in a simplified way, we define an elevation map for the grid system, as shown in Figure 7. For each grid, mucus can only flow to equal or lower elevation. We assumed that mucus from a grid flowed in any permitted direction at equal rates.

With these flow constraints in hand, we can define ordinary differential equations (ODEs) to describe the change in clearable particulate concentration in each grid over time. We assume that any response to intervention (i.e. inhaled IS or HS) occurs faster than the time-scale of imaging and that the effect is constant over the duration of the study. Thus the MCC rate leaving a grid should be constant from 10 min through the end of the measurement window. Assuming the clearable particles are well mixed within a 2D grid, the amount of activity leaving at any point in time should be proportional to the concentration of particles at the current time. This is mathematically described using a first-order kinetic rate law, shown in Equation 2.11, where $C(t)$ is the concentration of particulate in the grid at time, t, and k is the MCC rate leaving the grid. For our purposes, we treat concentration as radioactive counts/(pixel^2) because we are not considering motion into or out of the plane. Since all the grids for an individual are the same area of pixels, the denominator is constant with respect to time.

$$\frac{dC}{dt} = -kC(t) \tag{2.11}$$

Performing a mass balance around each grid, we can model the change in concentration of particles, $\hat{C}_{i,j}$ over time for any grid (row i, column j) using Equation 2.12, where $N_{i,j}$ are the (row, column) coordinates of the neighboring grids that are permitted to flow into grid



Row	1	2	3	4	5	6	7	8
1		9	9	9				
2	8	8	8	8	8			
3	7	7	7	7	8	8		
4	6	6	6	7	7	8	9	
5	6	2	6	6	7	8	9	
6	1	2	3	6	7	8	9	
7	1	2	3	6	7	8	8	9
8	1	2	6	6	7	7	8	8
9	1	2	3	6	6	7	7	8
10	6	2	3	4	6	6	7	8
11	6	3	3	4	5	6	7	8
12	6	6	4	4	5	6	7	8
13	7	6	6	6	6	6	7	8
14	7	7	7	7	7	7	7	8
15	8	8	8	8	8	8	8	8
16		9	9	9	9	9	9	9

Figure 7: Process of generating flow constraints. 2D projection of HRCT scan from [44] was divided into a 16 x 8 grid (left). Grids containing visible large airways were selected to define a large airway region (LAR), shown as a black and white mask in the middle panel. Inside the LAR, elevation was defined as the city-block distance to the trachea, located at the left edge. Outside the LAR, it was defined based on the city-block distance to the LAR (right). Flow between grids was constrained, so that mucus could only flow to neighboring grids of equal or lower elevation.

(i, j), k_n and $k_{i,j}$ are the MCC rates leaving the corresponding grid, and $D_{i,j}$ are the total number of directions mucus from grid (i, j) can flow in the grid system.

$$\frac{d\hat{C}_{i,j}}{dt} = \left[\sum_{n \in N_{i,j}} k_n \hat{C}_n(t)\right] - D_{i,j} k_{i,j} \hat{C}_{i,j}(t) \tag{2.12}$$

The initial concentration in each grid at time, $t = 10$ min, is estimated as the average measured concentrations at times, $t = 8, 10, 12$ min.

2.2.3.3 ABS model

We assume that the deposition pattern of In-DTPA is the same as that of Tc-SC both in terms of the amount deposited to each grid and how much of that grid is clearable by MCC. Thus, the difference in activity between the In-DTPA and Tc-SC can be assumed to be a result of ABS. Similar to the assumptions made for MCC, we assume that ABS is constant over the span of the study window and thus the change in concentration of In-DTPA due to ABS will be proportional to the current concentration of In-DTPA. Since ABS occurs we fit a rate coefficient for each grid and model the concentration of In-DTPA in each grid by simply adding a first-order term for ABS to the MCC model:

$$\frac{d\hat{C}_{i,j}}{dt} = \left[\sum_{n \in N_{i,j}} k_n \hat{C}_n(t)\right] - D_{i,j} k_{i,j} \hat{C}_{i,j}(t) - k_{i,j}^{ABS} \hat{C}_{i,j}(t) \tag{2.13}$$

2.2.4 Formulation as a Multi-Objective Optimization Problem

We will compare the results of three different formulations of an overall objective function, which are summarized in Table 3. The first objective function uses the sum (across grids) of the sum (across time) of squared residuals. The second objective takes the L_∞-norm of each of the individual objectives. Practically, this is implemented by minimizing the value of a slack variable, which is constrained to be greater than or equal to the objective of every grid. The third, which we will refer to as a normalized L_∞-norm objective, is similar to the second objective, but will use Chebychev scalarization [39], where an "ideal" objective value is subtracted from each grid. This ideal value is the value of the optimized objective function for just that grid. In other words, if we just optimized for one grid, it would be the value of that objective function. This may perform better in the face of unexpected disturbances (i.e. patient shifts or interference from activity flowing down the esophagus) because it would not sacrifice performance in other grids to try to minimize the error due to these disturbances.

Table 3: Summary of overall objective functions compared

Number	**Description**	**Mathematical Expression**
1	L_2-Norm	$\min_{\boldsymbol{k}} \sum_{i=1}^{16} \sum_{j=1}^{8} \sum_{t \in T} \left(\hat{C}_{i,j}(\boldsymbol{k}, t) - C_{i,j}(t)\right)^2$ $T = \{10, 12, ..., 80\}$
2	L_∞-Norm	$\min_{\boldsymbol{k}} s(\boldsymbol{k})$ s.t. $s(\boldsymbol{k}) \geq g_{i,j}(\boldsymbol{k})$ $\forall i \in \{1, ..., 16\}, j \in \{1, ..., 8\}$ $g_{i,j}(\boldsymbol{k}) = \sum_{t \in T} \left(\hat{C}_{i,j}(\boldsymbol{k}, t) - C_{i,j}(t)\right)^2$ $T = \{10, 12, ..., 80\}$
3	Normalized L_∞-Norm	$\min_{\boldsymbol{k}} s(\boldsymbol{k})$ s.t. $s(\boldsymbol{k}) \geq g_{i,j}(\boldsymbol{k})$ $\forall i \in \{1, ..., 16\}, j \in \{1, ..., 8\}$ $g_{i,j}(\boldsymbol{k}) = \sum_{t \in T} \left(\hat{C}_{i,j}(\boldsymbol{k}, t) - C_{i,j}(t)\right)^2 - g^*_{i,j}$ $T = \{10, 12, ..., 80\}$

We fit the MCC model parameters for all healthy controls by optimizing each of these objective functions. To provide a holistic view of the quality of fit for each of these objective functions, we calculate the mean absolute error (MAE) for individual grids, as well as for the whole lung.

2.3 Results

One key aspect from traditional methods of analyzing planar scintigraphy that we aimed to address with this new model was the heterogeneity within each of the compartments or ROIs. In Figure 8, we compare the variance of pixel intensity, which corresponds to radioactivity, in traditional central and peripheral ROIs with that of all the grids in our 16 x 8 system. Though we might expect a greater decrease in variance with the smaller areas of each grid, this also decreases the sample size of pixels used for the estimate of the variance. The skew of the pixels in the grids, shown in Figure 9, is much lower than that of the central and peripheral ROIs.

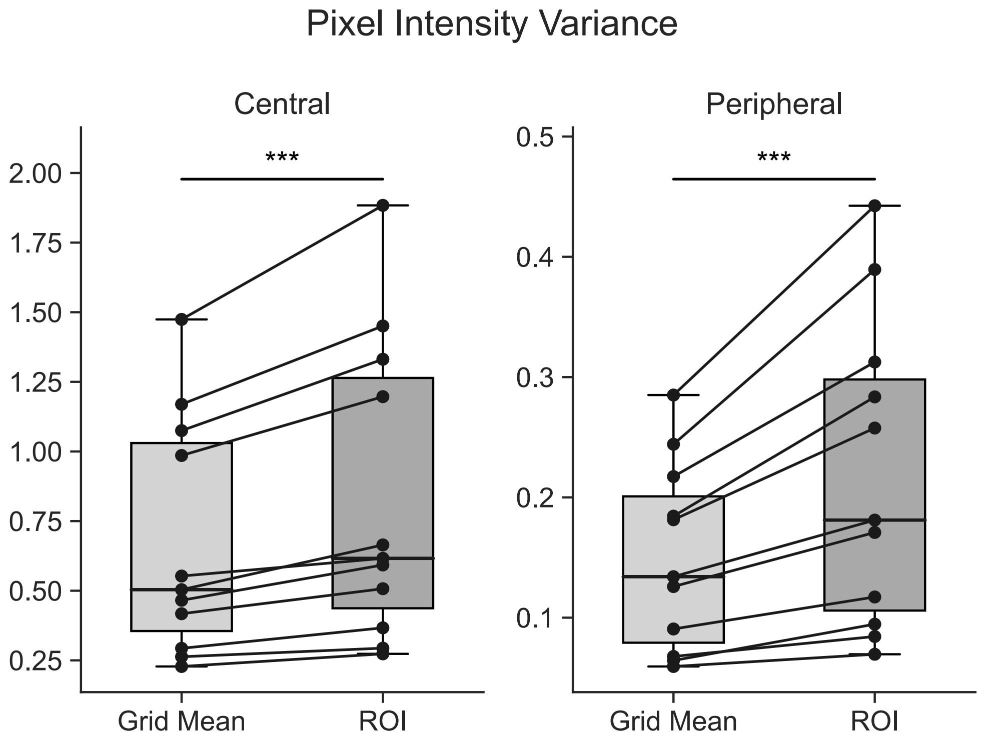


Figure 8: Boxplots of the variance of pixel intensity in the overall central ROI and the mean for individual grids in that region (left) and for the peripheral ROI (right). Solid black lines show change in variance between the grid mean and the overall ROI for each individual. Grid mean vs ROI were compared using a paired Wilcoxon signed-rank test (***: $p < 0.001$).

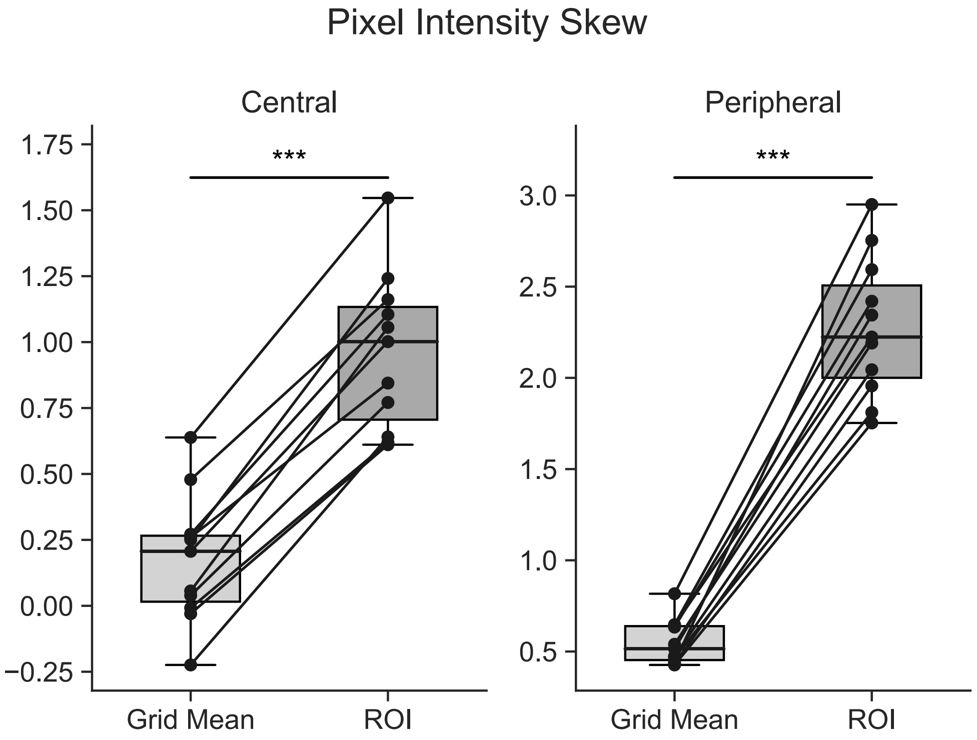


Figure 9: Boxplots of the skew of pixel intensity in the overall central ROI and the the mean for individual grids in that region (left) and for the peripheral ROI (right). Solid black lines show change in skew between the grid mean and the overall ROI for each individual. Grid mean vs ROI were compared using a paired Wilcoxon signed-rank test (***: $p < 0.001$).

The median non-clearable activity across healthy subjects is presented in Figure 10. While the non-clearable activity in the more distal grids is most likely in the alveoli, the non-clearable activity overlaying the large airways may also include mucus that would eventually clear – just not over the duration of our study.

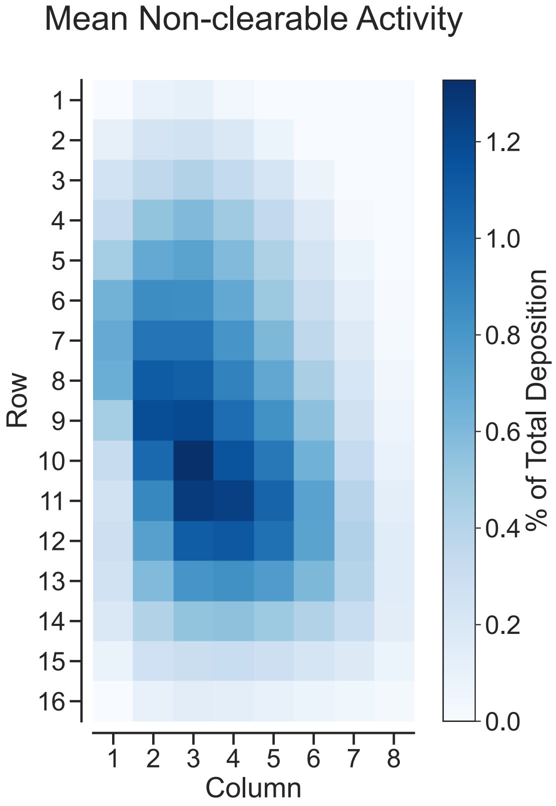


Figure 10: Mean non-clearable activity, defined as the lower 25% quartile of concentration of Tc-SC in each grid across the 80 minute study, for healthy subjects.

Each of the overall objective functions were fit to Tc-SC retention data from the same 11 healthy controls. For each, the MAE was calculated between the model and experimental data for each individual grid. The mean and maximum grid MAE was calculated for each individual and these are summarized in the first two columns of Table 4. The MAE was also calculated between the model simulations of the whole lung and the experimental data, which are presented in the third column of Table 4.

The normalized L_2-norm objective function performs well under each of these metrics, while the other objective functions perform poorly for at least one of the metrics. For this

Table 4: Comparison of error metrics (mean±std) for each of the tested objective functions fit to Tc-SC retention for healthy subjects.

Objective Function	**Mean Absolute Error (MAE)**		
	Mean Per Grid	**Max Grid Error**	**Whole Lung**
L_2-Norm	0.085 ± 0.016	0.24 ± 0.064	1.9 ± 0.68
L_∞-Norm	0.14 ± 0.027	0.34 ± 0.21	11 ± 3.4
Normalized L_∞-Norm	0.11 ± 0.021	0.28 ± 0.12	7.7 ± 3.5

reason, the model optimized using the normalized L_2-norm objective was selected for use with the ABS model. To better understand the performance of this model, sample trajectories from three healthy controls for the whole lung and three selected grids are shown in Figure 11.

Next, the ABS model was fit to the In-DTPA retention data. The fraction of activity in each grid that is non-clearable through MCC was assumed to be the same as for Tc-SC. This was a necessary assumption to fit parameters, however, it may be confounded by faster ABS in the alveoli than in the airways due to the higher relative surface area. The parameters for MCC rates from the L_2-norm objective were used and ABS rate parameters were fit for each grid. As shown in Figure 12, the fitted model cannot capture the trends nearly as well. This suggests that one or more of the assumptions made may not hold true.

We hypothesized that this was related to our assumption of identical deposition of both Tc-SC and In-DTPA. To assess the difference in deposition, we compared initial activity in each grid between the two isotopes, as shown in Figure 13. While the distribution of differences across all grids is very tight about 0, there seems to be a trend towards more Tc-SC deposition in the upper lobe of the lung than In-DTPA, and vice versa in the lower lobe. Though these differences are small individually, when all 114 grids are combined the overall change may be substantial. Further, while there is no way to assess the proportion of In-DTPA that lands in non-clearable spaces independently of Tc-SC, if deposition was identical, it would be impossible for HC10 to clear more Tc-SC (Figure 11) than In-DTPA (Figure 12).

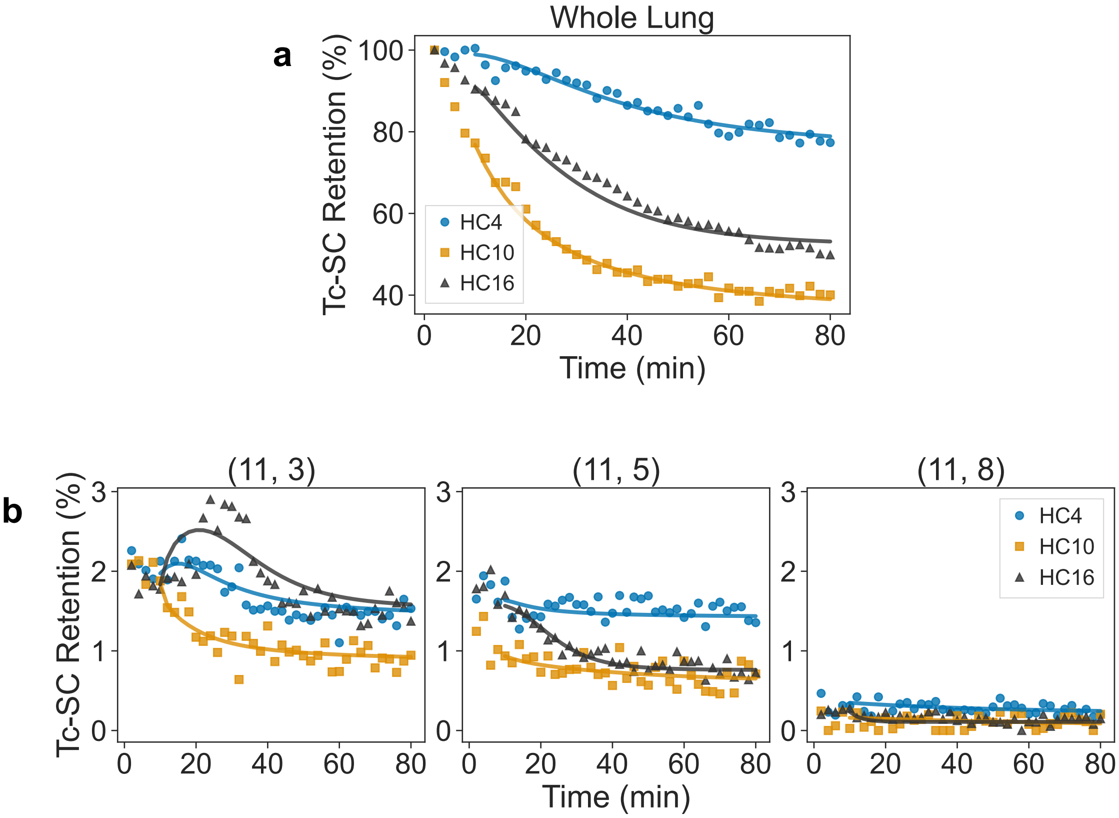


Figure 11: Comparison between fitted model (solid lines) and data (markers) at the whole lung scale (a) and in three example grids from row 11, columns 3, 5, and 8, which are at different elevations (b). Colors and markers correspond to three individual healthy subjects (HC4, 10, and 16)

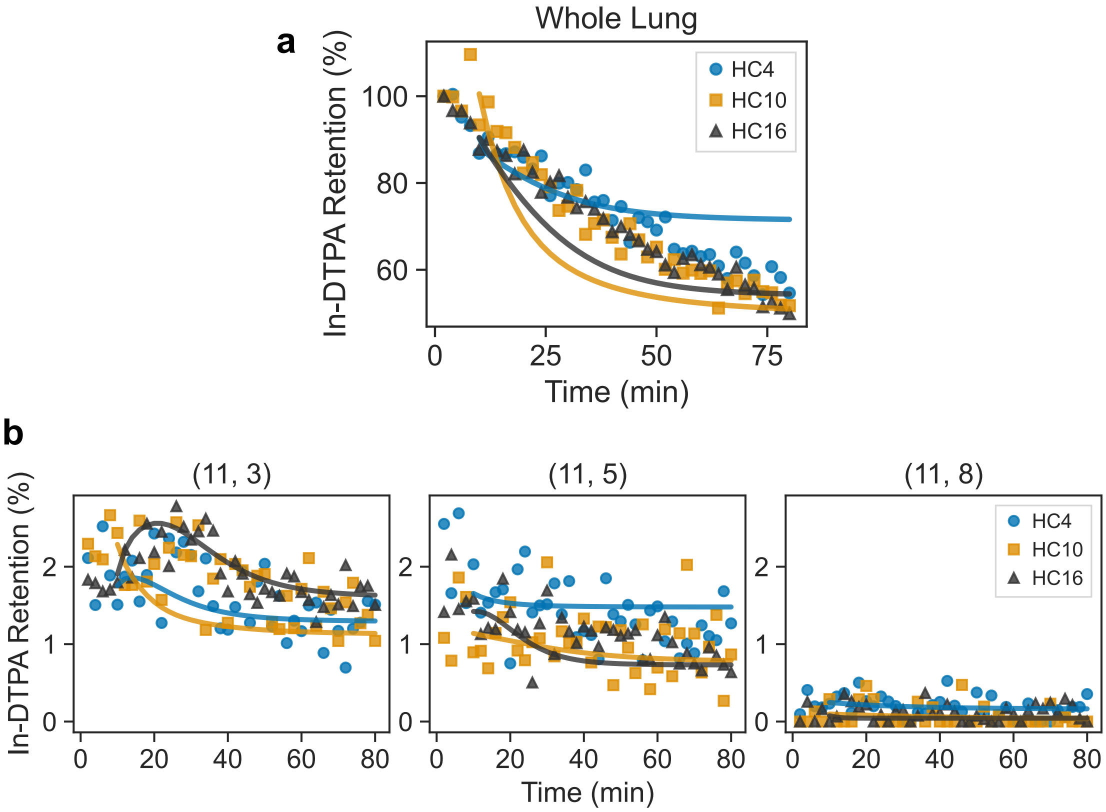


Figure 12: Comparison between fitted model (solid lines) and In-DTPA retention (markers) at the whole lung scale (a) and in three example grids from row 11, columns 3, 5, and 8 (b). Colors and markers correspond to three healthy individuals (HC4, 10, and 16).

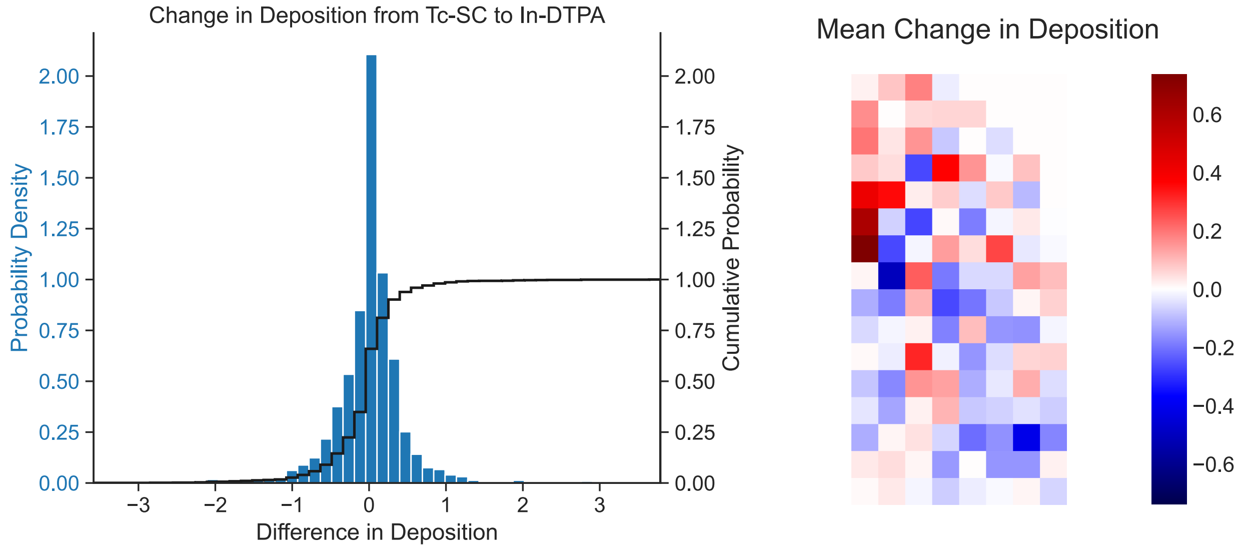


Figure 13: Probability density of the difference between Tc-SC and In-DTPA deposition for each grid (left) and the mean difference across healthy controls (right).

2.4 Discussion

The model developed in this section is able to accurately capture MCC dynamics observed in planar scintigraphy and improves upon previous methods in several ways. First, it is able to fit trends observed on an individualized basis, not just at the population level. This is essential for creating a tool for personalized medicine. Without an accurate way to describe individual response, we cannot determine how well a person responds to inhaled treatment.

Moreover, it can describe very localized dynamics. Previous models and measurement techniques lumped large areas of the lung into just two regions [32, 33, 35]. This was limiting in that both regions had heterogeneous deposition patterns, which muddied the description of clearance from those zones. With this new model, each compartment is much more homogeneous, as indicated by Figures 8 and 9. The variance in the central ROI is generally much greater than that of the peripheral ROI. This is consistent with visual "hot spots" localized to large airways that were observed in Bennett et al. [36].

While the variance is only slightly lower in the grids relative to the entire ROIs, it is still statistically significant. In contrast, the change in skew is quite dramatic. Particularly in the peripheral ROI, the skew in the whole ROI is more than 4 times that of the individual grids, indicating a much longer right tail relative to the mean of the distribution. The greater homogeneity in these grids allows for the description of more precise dynamics. In addition, by implementing anatomically-based flow patterns, we can fit MCC rates to individual grids and still recapitulate what is observed at the whole lung scale, as shown in Figure 11.

The ability to accurately capture these dynamics was enabled by the use of an appropriate overall objective function. The normalized L_∞-norm performed better than the unnormalized formulation. As alluded to, this is likely due to measurement disturbances, such as interference from swallowed aerosol, that are irrelevant to the underlying MCC process. In cases where there is such a disturbance, the unnormalized objective will accept slightly worse fits for several other grids in order to minimize the large disturbance in one grid. Since the large error is not related to the physiology the model is based on, this results in poor agreement between simulated and experimental whole lung activity. Similarly, the L_∞-norm objectives can be insensitive to small errors, especially in the more peripheral regions because the magnitude of error is higher in the more dynamic LAR.

While this was successful in describing MCC dynamics, it performed somewhat poorly when applied to ABS dynamics. This is likely due to the assumption of equal deposition in non-clearable space of Tc-SC and In-DTPA. While there are some participants for which this is clearly not the case (e.g. HC10, where more Tc-SC is cleared than In-DTPA), there are others where it is unclear whether or not this holds. Without this assumption, however, we are unable to determine how much of the inhaled In-DTPA cannot be cleared through MCC. Because In-DTPA can be absorbed paracellularly, we cannot use the same method for determining the amount of non-clearable activity as was used for Tc-SC. As a result, we cannot have any certainty in how much the change in In-DTPA retention is caused by ABS vs MCC. For this reason, we chose not to proceed with reducing the model of ABS.

This highlights one of the primary shortcomings of planar scintigraphy, which is that it is a 2D representation of a 3D system. While a single-photon emission computed tomography

(SPECT) scan would take too long to be useful in measuring the fast MCC dynamics in the large airways, there may be a suitable middle ground. For example, alternating between two camera angles could improve our ability to determine where along the airway tract the radiolabeled particles deposit. In future, such a setup may help distinguish activity that deposits in the alveoli from activity that just has not had sufficient time to reach the trachea during the study, thus providing a clearer picture of the underlying system we are modeling. This in turn might provide a better basis for how much In-DTPA is in the alveoli and thus cannot clear through MCC.

While this model provides a much clearer picture of how mucus is transported across the lung, it suffers from high model complexity. In particular, the large number of parameters fit for each subject makes interpretation of those parameters difficult. Additionally, due to individual differences in deposition, not all individuals have enough clearable activity in each grid to accurately inform the parameter values for all grids. Thus, it is prudent to reduce the overall scale of our model in a rational manner that will preserve the physiological basis. This will be focus of the next chapter.

3.0 Rational Model Reduction for Improved Patient-Specific Identifiability

3.1 Introduction

As mentioned previously, not all of the parameters could be uniquely identified with certainty for the model developed in Chapter 2. While it is structurally identifiable – meaning that with sufficient data, of the right kind, unique solutions could be found for each parameter, it is not practically identifiable (i.e. there is not enough information given the actual data we have). Thus, in this chapter, we focus on methods for reducing the complexity of the model. To start, we discuss the development of a new tool in Python called, plepy, which estimates likelihood-based confidence intervals for path-constrained systems. Path constraints define dynamic regions that a system must lie within. While not strictly necessary for our particular model, enabling the use of these constraints allows this tool to be used for a broader array of problems. Next, we take steps to decrease the number of fitted parameters while retaining the physiological constraints embedded in the model. After this rational reduction of parameters, we assess the identifiability of the remaining parameters.

3.1.1 Importance of Parameter Identifiability

While model parameters may provide excellent agreement between model and data, it is important to understand how much uncertainty there is in the value of these parameters. For instance, if a parameter can increase by 1,000 fold and the other parameters in the model can compensate to keep the error between model and data low, then we cannot have any certainty as to what the true value should be. In that case, we cannot draw any useful conclusions about the parameters for an individual, defeating the point of modeling the system in the first place. To be useful therefore, we need to ensure model parameters are uniquely identifiable for most individuals.

There are several methods of analyzing parameter identifiability for a system. For linear, time-invariant systems, we can analytically test for structural identifiability by calculating the observability matrix [45] from the state-space representation of the dynamic system (Equation 3.3).

If it is observable (rank($\mathcal{O}$) $=$ n, where n is the number of unknowns), then given sufficient data all of the parameters in the model can be uniquely identified. This is often a first test, where possible, because it is a necessary condition for parameters to be practically identifiable, meaning that given the actual data unique solutions can be found for each parameter. If the matrix is not of full rank, the system is structurally non-identifiable, meaning that regardless of how much data is gathered and how high the quality of that data, the parameters cannot ever be uniquely identified. In this case, additional assumptions must be made or steps taken to reduce the model before trying to fit model parameters. This analysis can only be applied to linear, time-invariant systems, however, which limits its use. Additionally, even if the system is structurally identifiable, there is no guarantee that appropriate data can actually be gathered to identify these parameters. In these cases, other methods must be used.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{3.1}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{3.2}$$

$$\mathcal{O} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{3.3}$$

One such method is through likelihood-based confidence intervals. The primary assumption for this method is that the measurement errors are Gaussian in nature and thus minimizing the sum of squared errors of the residuals is equivalent to maximizing the log-likelihood of the model given the data [46]. Thus, two model versions – the original model and an alternate model with a parameter of interest fixed to a value other than the fitted one – can be compared using a likelihood ratio test. The distribution of this test statistic asymptotically approaches a χ^2 distribution with 1 degree of freedom for large sample sizes [47]. Thus, it can be used to approximate a threshold value for the sum of squared errors for a given Type I error level, above which the alternate model can be rejected. Confidence limits can then be placed on individual parameters by finding the values above and below the fitted value that cause this threshold to be crossed. A depiction of these confidence intervals is shown in Figure 14.

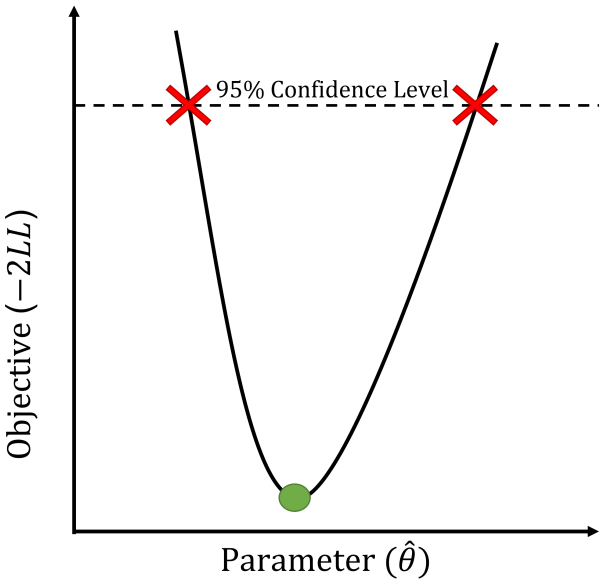


Figure 14: Depiction of a likelihood-based confidence interval. Green circle is the fitted parameter value, the dashed line represents the 95% confidence threshold, the solid curve represents the minimized objective value when the parameter is fixed to the corresponding value, and the X symbols indicate the lower and upper values of the confidence interval.

Several tools have been developed to generate confidence limits using this method [48, 49], however, they would require substantial reformulation of the model from Chapter 2. Additionally, they lack the ability to implement path constraints. Path constraints are limits on the value a state can take on over a specific time region. For example, there may be known physiological limits on a state, such that any model that exceeds these limits is aphysiological. While this is a common issue that arises in physiologically-based models, it can also arise due to safety, cost, or other physical constraints. Later in this chapter, we will present a new profile likelihood tool in Python that also enables the implementation of path constraints.

3.1.2 Measures of Identifiability

When analyzing the identifiability of parameters using likelihood-based confidence intervals, there are several possible outcomes. The best case is that confidence limits can be placed about the maximum likelihood estimate as in Figure 15A, such that there is a unique value that minimizes the sum of squared errors. In such instances, it is considered to be practically identifiable, meaning that given the model structure and data, a unique parameter can be found with a specified degree of certainty.

Another possible outcome is that there is a unique minimum value, however, beyond a certain value, the sum of squared error stops increasing, as in Figure 15B. This is a case of practical non-identifiability, where the parameters are theoretically identifiable, but the quantity and/or quality of data limits our ability to bound the parameter to a specified degree of certainty. There are two paths forward in this case: the first is to gather more data, ideally under different conditions from the original data, and the second is to reduce the complexity of model, such that model parameters can be uniquely identified from the existing data. While the former may provide better results in the case that this is possible (e.g. changing experimental conditions *in vitro*), we will focus on the second option in this work since obtaining additional planar scintigraphy data is not possible in the context of the current study.

Finally, identifiability analysis may demonstrate that certain model parameters can take on any value without increasing the sum of squared error between the model and data, as shown in Figure 15C. In this case, the model is also structurally non-identifiable. While this can be tested for *a priori* for linear, time-invariant systems, for nonlinear or time-varying systems, likelihood-based confidence intervals can be used to determine this. In this case, model reduction techniques can be applied to reduce the complexity of the model, and thus increase the identifiability of remaining parameters.

3.1.3 Image Processing Tools

While there are many traditional tools that can be used to reduce the scale of an ODE model based solely on the mathematical properties, we want to retain the physiological

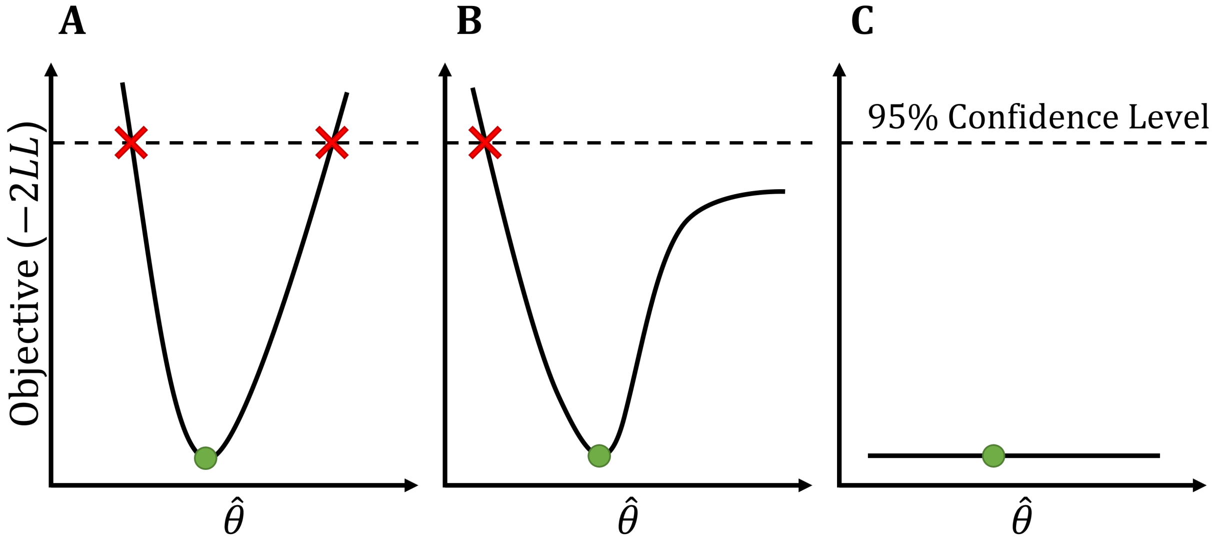


Figure 15: Depiction of likelihood profiles for a practically identifiable parameter (A), a practically non-identifiable parameter (B), and structurally non-identifiable parameter (C).

information that we have embedded in our model through flow constraints. Further, we expect based on human anatomy that neighboring grids will have similar sized airways. Rather than use mathematics alone to reduce this model, we will take advantage of the fact that our data and model are derived from images. This allows us to borrow common image processing tools and apply them to our parameter space.

There are two classes of tools that we are particularly interested in: 1) image denoising, and 2) image segmentation. Some of the individual grid parameters for many subjects are practically non-identifiable because there simply is not enough clearable activity in the grid to ground the best-fit parameter value. Thus, when we consider the MCC rates for a grid across all the healthy subjects, we end up with a number of values that are not meaningful. By applying image denoising techniues, we hope to minimize the impact of these non-identifiable parameters. Once we have clean data to work with, we are then interested in finding connected regions that have similar parameter values across healthy subjects. For this, we can apply image segmenation tools to the parameter values for each grid.

3.1.3.1 Image Denoising

There are numerous techniques available for denoising images, each carrying their own assumptions. Many of these techniques try to detect pixels that are very different from their neighbors via convolution matrices [50, 51]. What this means is that an individual pixel value in the denoised image actually depends on all of the pixels around it. For images with a large number of pixels, this can be really effective. With our system, however, we have a small array of grids and airway size could change drastically from one grid to another. Using one of these convolution methods would thus hamper the distinction of specific airway size classes. We do, however, have the benefit of having multiple measurements for each grid (i.e. the fitted parameters for each of the healthy controls). This allows us to use feature extraction techniques, such as principal component analysis (PCA).

PCA is an eigendecomposition of the covariance matrix of a data set. When the orthonormal eigenvectors are sorted by their corresponding eigenvalues from largest to smallest, they are ordered by the amount of variance in the data that can be described by that vector [50]. In the case that all eigenvectors are used, this equates to a change of basis of the original data into a different coordinate system. Because the eigenvectors are orthogonal, this has the added benefit of decoupling correlated data. An example of this is shown in Figure 16. In many cases, however, the higher order principal components (PCs) are discarded. The discarded PCs correspond to low variance directions – often due to measurement noise. The retained eigenvectors can thus capture high variance directions, often correspond to important features, without as much noise [51].

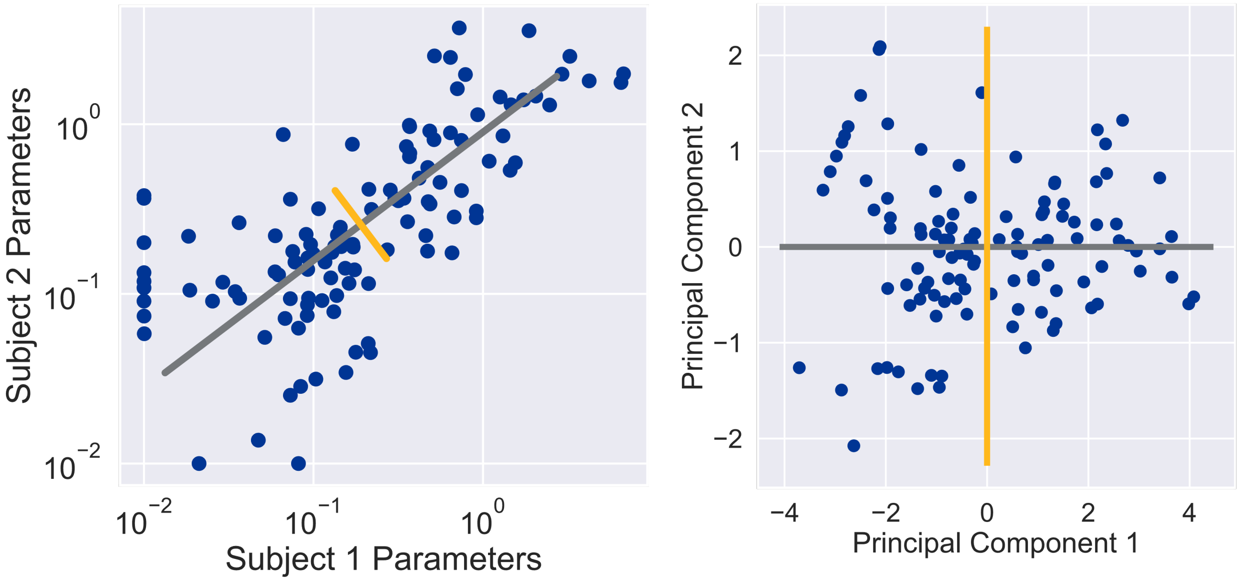


Figure 16: Example of how PCA can be used as a change of basis to describe the same data using linearly independent coordinates. Parameters for two sample subjects are highly correlated, as shown on left. Principal components (blue and yellow lines) capture directions of maximum variance. The same data expressed in terms of principal components (right) is no longer correlated.

A classic example of how PCA has been used for feature extraction was an early facial recognition method called, "eigenfaces" [52]. In that case, PCA was performed on a set of equal sized headshots, where the eigenvectors corresponded to pixel arrays (eigenfaces) capturing the directions of highest variance across the headshots. Using just the first 7 eigenfaces, they were able to accurately identify subjects from the training data under different lighting or head orientation and detect new faces [52].

In an analogous way, we can apply PCA to model parameters from our healthy controls to identify "eigenlungs" that capture the directions of high variance across our controls. In our case, we would expect parameter arrays that are highly patient-specific to be captured in the first few eigenlungs. In contrast, parameters that are frequently non-identifiable and thus get set to one of the parameter bounds would appear in eigenlungs corresponding to low variance. Thus, if we use a small subset of eigenlungs to describe the parameters of the healthy controls, we can effectively mitigate the impact of non-identifiable parameters.

3.1.3.2 Image Segmentation

We can also use tools developed for image segmentation. Image segmentation is an active area of research within medical imaging for accurately identifying different organs or tissue regions. Many of the tools for image segmentation use graph clustering techniques [53–55], meaning that pixels or voxels that are clustered together must be connected. One of the simplest of these techniques, which has been successfully applied to other medical images such as diffusion MRI tractography [53], is constrained hierarchical clustering. With this technique, a similarity metric ranging from 0 (no similarity) to 1 (identical) is defined between each node. Connectivity constraints are then applied by setting non-adjacent node similarities to 0. From there, the two most similar nodes can be merged into a cluster. A new similarity is calculated between that cluster and each other node. Then this process is repeated until all nodes are combined into one cluster (this assumes the graph is completely connected). This forms a dendrogram, showing where each node/cluster is combined. For a selected number of clusters, this can then be used to define which cluster each node belongs to. While most image segmentation techniques use pixel color or intensity, we can apply the same technique to grid parameters.

3.2 Profile Likelihood Estimator in Python

Profile likelihood is a method of analyzing parameter identifiability based on maximum likelihood. Assuming noise in the system is Gaussian, it can be shown that minimizing the sum of squared errors (SSE) is equivalent to maximizing the likelihood. Furthermore the SSE objective function is equal to $-2 * log(\mathcal{L})$. Uncertainty about the maximum likelihood estimate (MLE) of a parameter can be defined by the chi-squared distribution with single degree of freedom. Traditional profile likelihood tools work by fixing one parameter to a value slightly greater than (or less than - depending on which confidence limit you are searching for) the MLE value. The other parameters are then re-optimized to maximize the likelihood of the system and the log-likelihood is re-calculated. This is repeated over and over until the other parameters in the system cannot sufficiently compensate for changes in the fixed parameter and the log-likelihood drops below a threshold confidence level (defined by the chi-squared distribution). This is then considered one bound on the confidence limit and the process is repeated in the opposite direction. In some cases though, the other parameters in the system can indefinitely compensate for changes in the fixed parameter. In this case, the system is either practically or structurally non-identifiable.

Several limitations exist to using these traditional tools. Firstly, if the parameter of interest is non-identifiable, many many calculations are run only to come to the conclusion that the parameter is non-identifiable. This is often a waste of computational resources with little added information about the system. Additionally, to the best of our knowledge, the tools available only allow users to place constraints on parameter values, rather than states in the system. This requires users to pre-define parameter regions that satisfy path-constraints on the states of the system. While this is not terribly arduous for linear systems, for non-linear systems, there are often multiple regions that are feasible for any given parameter, if the other parameters of the system fall within a certain range. Traditional tools would therefore require a user to find confidence limits on parameters in each of these regions to find overall confidence limits.

To resolve these issues, we developed an open-source tool in Python called, Profile Likelihood Estimator in Python (plepy). This tool leverages the power of Python Optimization

Modeling Objects (PYOMO) to implement path constraints by converting the system of ordinary differential equations into discrete algebraic equations [56]. Using this framework, users can directly define path-constraints and the overall bounds on the parameters of the system.

Previous work from our group tried to implement the profile likelihood tool developed in [57] within this framework, however, it had limited capabilities. Specifically, it failed for many stiff problems because it would take larger and larger adaptive steps away from the optimum value in parameter space because there was little decrease in likelihood. Eventually, it would pass the parameter confidence limit and the likelihood would drop far below the threshold specified. However, because the step size had gotten so large, there was a huge window over which the precise confidence limit could have occurred. If the adaptive step size was removed for stiff problems, tens of thousands of iterations could be taken without reaching the confidence limit that we knew existed.

To improve the efficiency of the search for confidence limits, rather than taking steps away from the MLE parameter value, the parameter is first fixed to the bound in either direction for that parameter and the other parameters are re-optimized. This way, if the log-likelihood remains above the confidence threshold of interest, there is no need to calculate additional steps within that region. If it is below the confidence threshold, however, our tool uses a binary search to find the confidence limit of the parameter to within a user-specified degree of accuracy. Since it can also be useful to analyze how other parameters adjust to compensate for the fixed parameter, once the bounds are defined, intermediary values of the parameter can be estimated using the same process. This process is summarized in the algorithm shown in Algorithm 1.

Algorithm 1 Binary search for confidence limit

```
  **Initialize:**
      $x_{in} \leftarrow x_{opt}$
      $x_{out} \leftarrow x_{bound}$
      $x_{mid} \leftarrow x_{bound}$
      $y_{opt} \leftarrow \mathbf{g}(x_{opt})$                    ▷ $\mathbf{g}(x)$ is the minimized objective function
      $L_t \leftarrow \ln(y_{opt}) + \frac{1}{2}\mathbf{Z}(\alpha)$          ▷ $\mathbf{Z}(\alpha)$ is the inverse survival function for $\chi^2_{df=1}$
  $y_{mid} \leftarrow \mathbf{g}(x_{mid})$
**Ensure:** $y_{mid}$ is feasible                ▷ If $y_{mid}$ is infeasible, enter binary search for max
                                                   feasible solution.
  $L_{mid} \leftarrow \ln(y_{mid})$
  **if** $L_{mid} < L_t$ **then**
      $x_{CI} \leftarrow \infty$
  **else**
      **while** $(x_{out} - x_{in}) > \delta_{tol}$ **do**                ▷ $\delta_{tol}$ is a minimum step size
          $x_{mid} \leftarrow \frac{1}{2}\left(x_{out} + x_{in}\right)$
          $y_{mid} \leftarrow \mathbf{g}(x_{mid})$
          $L_{mid} \leftarrow \ln(y_{mid})$
          **if** $L_{mid} > L_t$ **then**
              $x_{out} \leftarrow x_{mid}$
          **else**
              $x_{in} \leftarrow x_{mid}$
          **end if**
      **end while**
      $x_{CI} \leftarrow x_{mid}$
  **end if**
  **return** $x_{CI}$
```

3.3 Methods

To reduce the flexibility of the model, the first step we took was to consider only allowing mucus to flow to grids of equal elevation if their own elevation was below some threshold value. We defined this value as the threshold elevation and varied it from 2-9, re-optimizing the objective function (sum of squared errors across time and space) at each threshold elevation. We then compared the objective values at each step with that of the full model developed in Chapter 2. We looked for a sharp increase in error or "knee" to use as the threshold elevation for the remainder of the model reduction.

We then sought to apply PCA across the parameters for the healthy controls. PCA can only be applied to matrices with normally distributed data, so we first needed to process our parameters. We initially formed our data matrix by flattening the 2D matrix of MCC rate parameters from each individual into one vector per individual, where the row corresponds to a specific grid coordinate. Each of these makes up one column of our data matrix. The initial parameter distribution for a given individual was strictly positive and approximately log-normal in shape, as shown in Figure 17. Therefore, we took the natural log of the original parameters and then transformed them using Equation 3.4, where $\mathcal{G}(\mu, \sigma)$ is the log of the initial distribution, which is approximately normally distributed with mean, μ and variance σ^2.

$$\mathcal{Z}(0,1) \approx \frac{\mathcal{G}(\mu, \sigma) - \mu}{\sigma} \tag{3.4}$$

From there, the eigenvalue decomposition was performed on the transformed dataset across all healthy controls. This was implemented using the scikit learn package in Python [58]. To reduce the effect of non-identifiable parameters on model reduction, we discarded PCs that captured 5% or less of the variability in the data across healthy controls. The remaining PCs were visualized in 2D to create "eigenlungs" corresponding to the directions of greatest variability. This can provide intuition as to which areas are most important to capture to be able to describe the MCC dynamics of an individual.

We then performed constrained hierarchical clustering on the grids based on similarity of the first two eigenlungs. We used the Ward metric for similarity, which minimizes the within

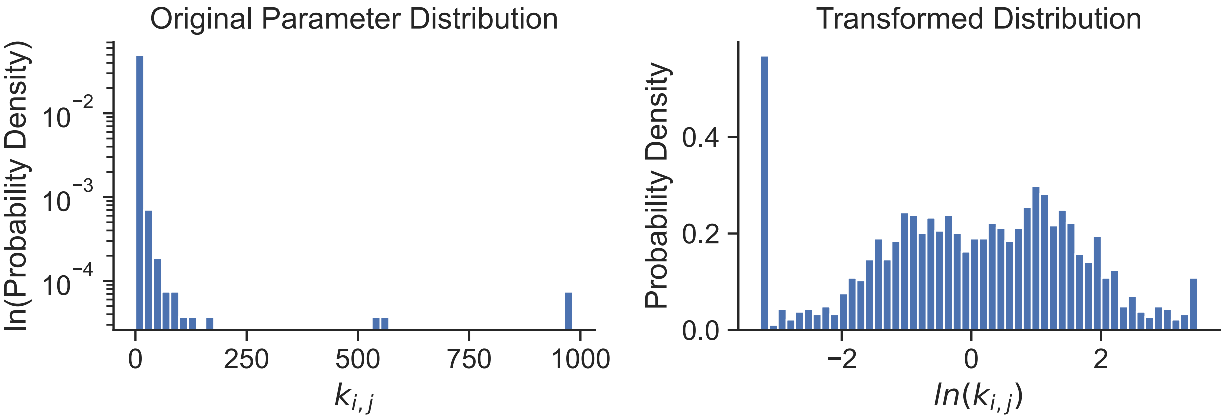


Figure 17: Original distribution of MCC rate parameters for healthy controls (left) and the transformed distribution after processing (right).

cluster variance [59]. To implement the connectivity constraints, an adjacency matrix was calculated. The adjacency matrix is a pairwise matrix for each set of grids that has a value of 1, if they are neighbors, and 0 otherwise. The adjacency matrix was then applied as a mask on the similarity matrix, such that only neighboring grids had a value greater than 0. This was implemented using scikit learn in Python [58].

To identify the minimum number of clusters necessary to adequately capture the MCC dynamics of an individual, the number of clusters selected from the dendrogram formed by the constrained hierarchical clustering was increased from 2 to 10. For each set of clusters, the objective function from Chapter 2 was re-optimized setting all MCC rate coefficients for a given cluster to the same value. The flow constraints were defined based on the threshold elevation determined previously. The change in overall objective from the full model was calculated for each individual for each number of clusters. We then looked for an elbow in the error, where a decrease in cluster number would greatly increase the error and an increase would have little effect.

Pearson's correlation coefficients were calculated between each parameter to evaluate any linear dependence between them. If there was a significant correlation, one of the parameter pair could be estimated from the other using a linear model rather than fitting it independently.

After selecting an appropriate number of clusters and fitting MCC rate coefficients for each cluster, 95% confidence intervals were calculated for each parameter for each individual. The proportion of individuals with identifiable lower and upper confidence limits and the range of those intervals was calculated for each parameter.

3.4 Results

3.4.1 Threshold Elevation

The change in sum of squared error when different elevations are used as a threshold, below which mucus can flow to equal elevation grids, is shown in Figure 18. While the magnitude of change is relatively small using each elevation as a threshold, there is a knee in the error change at a threshold elevation of 4. We chose this for a threshold elevation going forward with the remainder of the model reduction.

3.4.2 Eigenlungs

Next, we performed PCA across healthy controls. Only the first two PCs captured more than 5% of the variance in the data and combined account for 74% of the overall variance in the data, as presented in Figure 19a. These two PCs are depicted as eigenlungs in Figure 19b. They roughly correspond to axes of small vs large airway and upper vs lower lobe.

3.4.3 Hierarchical Clustering

Discarding all but the first two eigenlungs of the healthy controls, we then applied constrained hierarchical clustering. The dendrogram formed for the entire grid network is shown in Figure 20a, where each of the rows corresponds to an individual grid and the two columns of the heatmap are the associated values of the first two eigenlungs. Each branch point (ver-

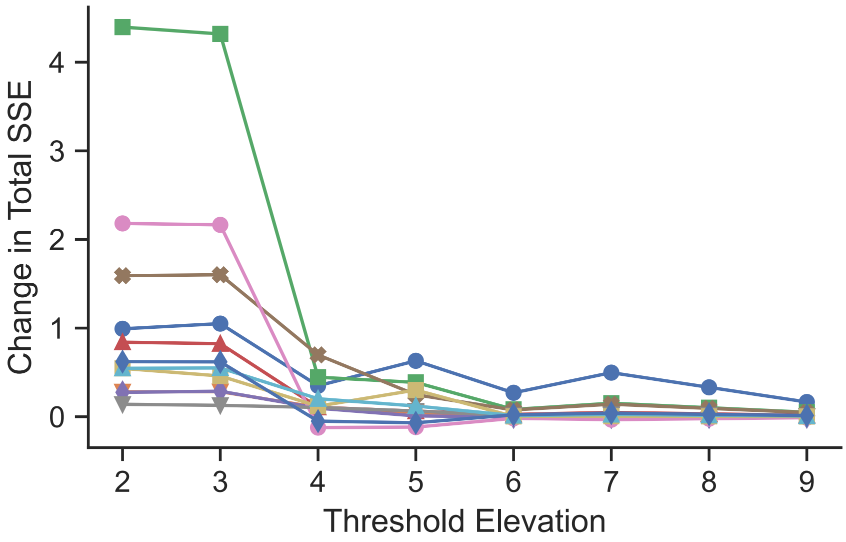


Figure 18: Change in total sum of squared error (across all time points and grids) between model and MCC scan data from original model versus threshold elevation, below which mucus can flow to equal elevation grids. Each line represents a healthy control

tical lines) on the dendrogram represents the bifurcation of the parent cluster (horizontal line from the left) into two smaller clusters (horizontal lines on the right).

To select an appropriate number of clusters for the reduced model, the number of clusters was increased from 2 to 10 (working from left to right on the dendrogram). For each set of clusters, a single MCC rate parameter was fit for each cluster without changing any of the flow constraints (i.e. mucus still flowed to lower or equal elevation grids based on the threshold elevation of 4). The change in total sum of squared error relative to the full model was calculated for each set of clusters and is compared in Figure 21. There was a knee in the error at 5 clusters, which corresponds to those shown in Figure 20.

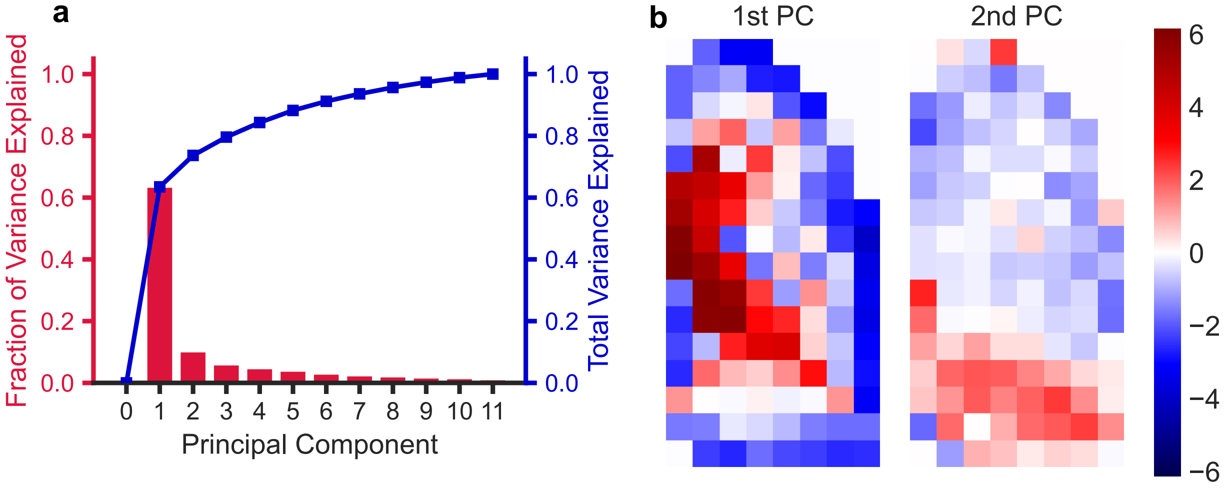


Figure 19: a) Principal component (PC) analysis of log-normalized clearance rate coefficients across healthy subjects. PCs are orthogonal and ordered by the amount of variance in the data they explain. b) The first two PCs, represented spatially. The scale bar indicates the magnitude and direction of the components from the median. Parameters of all healthy subjects can be described as a linear combination of these two PCs to within 74% accuracy

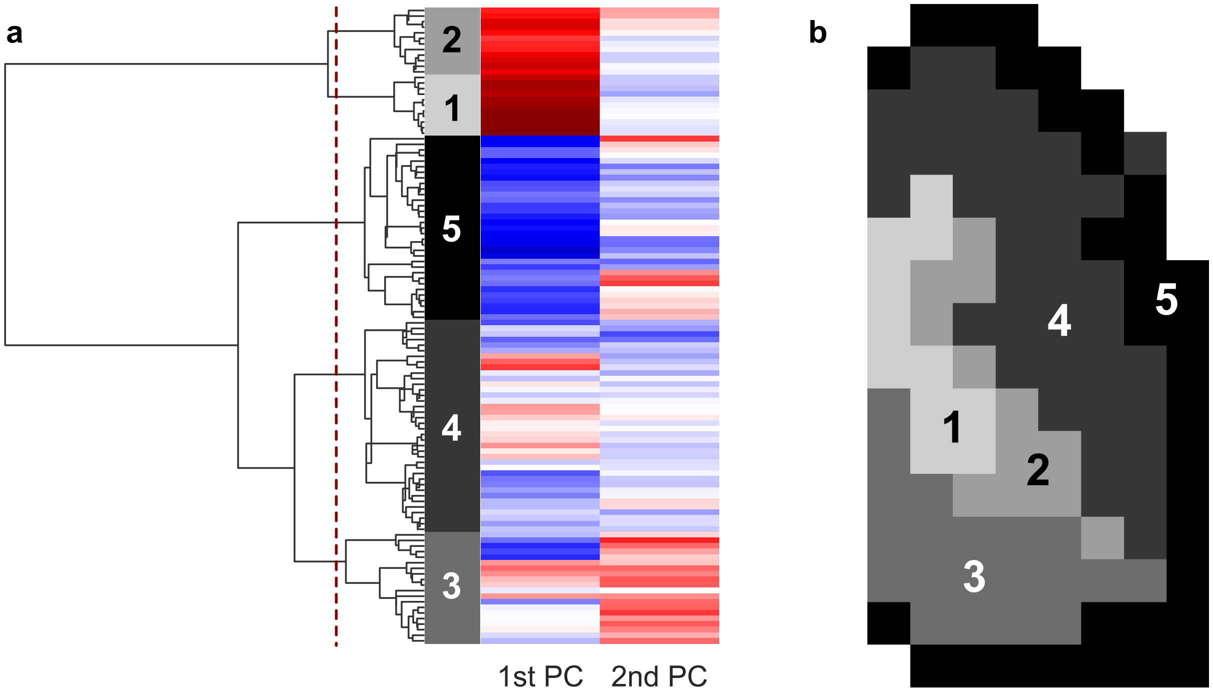


Figure 20: a) Dendrogram and heatmap showing hierarchical clustering of 1st and 2nd principal components of healthy subject clearance rate coefficients and b) spatial layout of selected clusters. Each colored row of the heatmap represents a grid in terms of principal components (columns). Each vertical line on the dendrogram represents the merging of the next most similar cluster or grid. The dashed line shows where the dendrogram was truncated to form the selected clusters

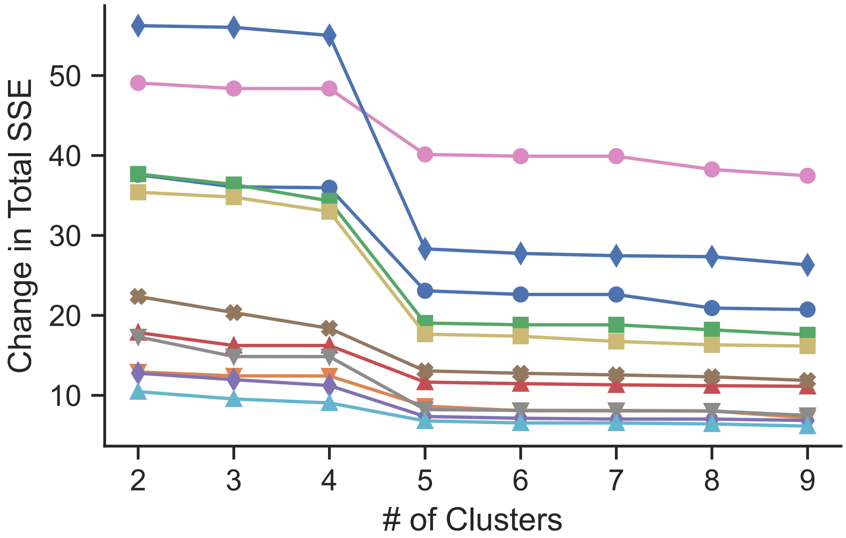


Figure 21: Change in total sum of squared error (across all time points and grids) between model and MCC scan data from original model versus the number of clusters used in the reduced models. Each line represents a healthy control

3.4.4 Parameter-Parameter Correlation

In order to minimize the number of fitted parameters, a linear model could be used to estimate one parameter from a highly correlated pair. Correlation coefficients were calculated between each of the fitted cluster parameters, as shown in Figure 22. There were no significant correlations between the fitted parameters.

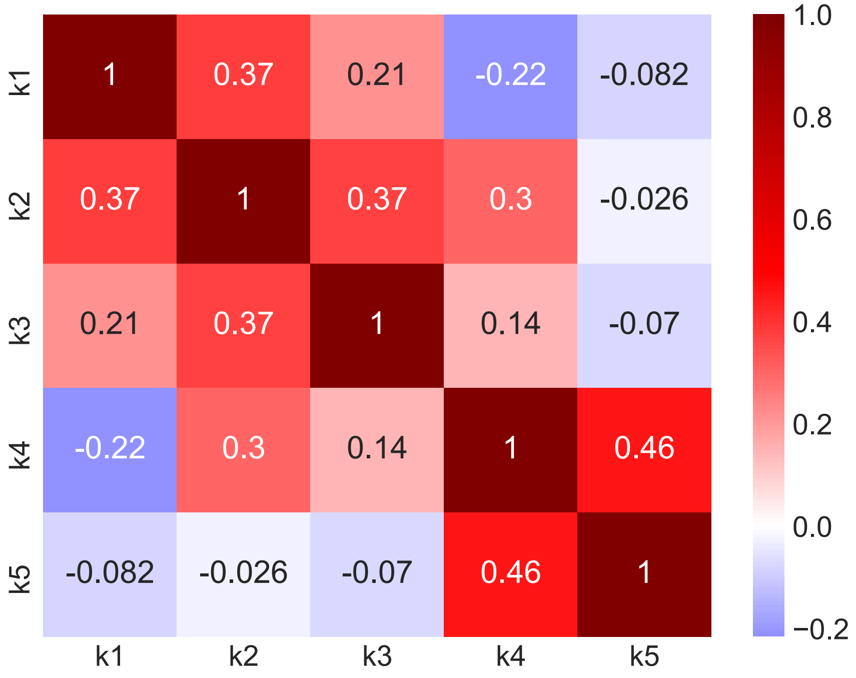


Figure 22: Pearson's correlation coefficients between each of the fitted parameters. Rows and columns (top to bottom, left to right) correspond to the MCC rate coefficients for healthy subjects for each cluster (k_1-k_5). There are no significant correlations.

3.4.5 Model Identifiability

Likelihood based 95% confidence intervals were generated for the fitted parameters for each healthy individual. For each cluster, at most 2 of the 11 healthy controls had a bound that was non-identifiable. The range of those confidence intervals was higher in clusters 1 and 2, which roughly overlap the large airways, and smaller in clusters 3, 4, and 5. It should be noted that for cluster 2, the large average interval size was due almost entirely to one individual who had a range of 996 min^{-1}. The median interval for cluster 2 was 4.80 min^{-1} and if that individual is excluded, cluster 2 has a mean ± std of 14.5 ± 14.7.

Table 5: Number of participants with identifiable lower and upper 95% confidence limits and the range of those intervals (mean±std) for each of the 5 clusters of the reduced model.

Cluster	**Lower Limit**	**Upper Limit**	**Range (min^{-1})**
1	11/11 (100%)	9/11 (81.8%)	15.4 ± 7.00
2	11/11 (100%)	10/11 (90.9%)	111 ± 311
3	10/11 (90.9%)	10/11 (90.9%)	1.95 ± 1.39
4	11/11 (100%)	9/11 (81.8%)	2.02 ± 0.535
5	11/11 (100%)	11/11 (100%)	1.02 ± 0.434

3.5 Discussion

By methodically reducing the scale and complexity of the model developed in Chapter 2, we were able to improve the identifiability and interpretability of the fitted parameters while preserving the physiological basis embedded by the flow constraints. The total number of free parameters was reduced from 114 down to 5 with only an 8.7% increase in mean grid MAE (from 0.085±0.016 to 0.092±0.016). Using likelihood-based confidence intervals, we were also able to determine that there were unique minima for the parameters of this model for most individuals.

Interestingly, the 5 clusters identified from the hierarchical clustering, as shown in Figure 20b, roughly correspond to anatomical features. Clusters 1 and 2 overlay the area in 2D where we would expect to find the main bronchus and bronchus intermedius, respectively [44]. Clusters 3 and 4 approximately correspond to where the lower and upper lobes of the lung would be, respectively. While we might expect there to be 3 clusters – one for each lobe of the right lung, there is significant overlap between the lobes when projected onto a 2D plane. It is likely that any difference in MCC rate between the lower and middle or middle and upper lobes would be confounded by this overlap. Finally, though it is not a distinct anatomical region, cluster 5 corresponds to the most distal regions of the lung from a 2D perspective.

From this, we can identify 5 key regions on planar scintigraphy scans that are most important for capturing patient-specific clearance. The eigenlungs from the PCA in Figure 19b really highlight where previous methods were successful and what key regions they were missing. The first eigenlung primarily separates the grid space into large airway regions and distal regions from an anatomical perspective. While this supports other evidence that there is consistently faster MCC in the large airways than in smaller ones, it also indicates that the magnitude of that difference in MCC rates is highly variable across healthy individuals. This supports the common practice of dividing the lung into a central and and peripheral ROI. However, this also suggests that there is more nuance required to adequately describe the MCC behavior for an individual. The most similar analysis of this is from Alcoforado et al. [43], where they used the same HRCT scans from Greenblatt et al. [44] to define anatomically-based central and peripheral ROIs. Although they were only used to measure endpoint changes in MCC and heterogeneity of aerosol deposition, the anatomically-based ROIs captured patient-specific regions much more closely than the traditional box-shaped central ROI. A large aspect that has been missing from standard MCC analysis of planar scintgraphy scans is the difference between upper and lower lobe areas, which is illustrated well in Figure 19b. This distinction was shown to be the second most variable feature between individuals.

While these are key areas, it should be noted that these are not simply new ROIs that can be treated statically. Recall that within each of these clusters, we are still implementing flow constraints on a grid-by-grid basis. We still need to model mucus as it flows across each of these clusters toward the large airways and up to the trachea in order to accurately describe the system dynamics, however, we can now capture these dynamics with a small number of parameters.

The clusters identified also highlight the utility of using constrained hierarchical clustering rather than clustering based on similarity alone. While it is likely that the MCC dynamics could be captured just as well with another method, the resulting clusters could include grids that are scattered across the lung. That would make interpretation of the reduced model much more difficult due to the lack of physiological meaning. By enforcing connectivity of the clusters, we reduce the formation of spurious clusters. Because there is a physiological rationale as to why neighboring grids would behave similarly (namely they contain similar sized airways), they can also provide spatial insight into the MCC behavior of an individual.

When we consider a patient-specific model, we want to make sure we have confidence in the unique value of those parameters to be able to draw useful conclusions. To that end, we have demonstrated that the parameters of the reduced model are uniquely identifiable for most subjects. One exception, which resulted in most of the non-identifiable confidence limits in Table 5, is where there is very little aerosol deposited to a certain region. This is completely unsurprising and to be expected. Practically, no model will be able to uniquely identify clearance rates without a strong enough signal from a particular region. In the case that a parameter is non-identifiable due to low signal, the average clearance rate coefficient for that cluster can be used to describe the MCC dynamics for that individual with minimal impact on error. When analyzing parameters across a population, these parameters could then be excluded to avoid confounding any underlying relationships in the parameters that actually are identifiable.

One limitation to this work is the small sample size of healthy controls. With one participant excluded from this analysis for low overall activity, we are only working with 11 individuals. Though the high degree of parameter identifiability and small change in overall error leads us to believe this model will be sufficient to describe MCC dynamics across all individuals, there is a possibility that with additional data from new participants, these results could change slightly. Specifically, the eigenlungs associated with a larger population may have slightly different values and therefore could lead to slight changes in the clusters selected. Because the drop in variability after the first eigenlung is very pronounced, we would expect very little change in that with additional participants and thus would expect little, if any change in clusters 1 and 2. Though the precise boundaries of the other clusters might change slightly, we would not expect large changes in the number or location of clusters. Based on the change in error observed in Figure 21, we would expect to still need at least 5 clusters. There may, however, be enough variability over a larger population to make the use of 6 clusters more appropriate. Alternatively, the potential increase in variability could lead to the use of an additional eigenlung, which could influence the boundaries of the clusters.

4.0 Patient-Specific Modeling of Hypertonic Saline Response in Cystic Fibrosis

4.1 Introduction

Because CF is a rare disease, most single site clinical trials are performed on very small sample sizes. Of the 283 completed clinical studies in the US (listed on ClinicalTrials.gov as of April 2023) that exclusively studied CF participants, the median enrollment was 42. Low enrollment limits the ability to adjust statistics for co-variates, and often results in low-quality evidence supporting a therapy or inconclusive results. Those trials that are larger, are typically multi-site studies. While this can greatly increase the statistical power, they are often more costly and any deviations in protocol between sites can confound these results.

A number of clinical trials have studied the effect of HS on MCC in CF individuals [31, 35, 60]. While most conclude that there is a therapeutic effect, the magnitude and duration of the response has been shown to vary greatly between individuals [61]. Depending on the actual response for an individual, there may or may not be a sufficient response to justify the lengthy process of administering HS and sanitizing the equipment after. There is no clear method for determining who will benefit most from HS or how to determine the effect on a per-patient basis.

We were interested in building a tool to better estimate patient-specific response to HS. In addition to the nuclear imaging that was used to develop the MCC model, participants in the study donated human nasal epithelial cells (HNEs), performed pulmonary function tests, a sweat chloride test, and a multi-breath washout study. Additional *in vitro* experiments were performed on the HNEs to assess the electrophysiology of the cells, the volume of the airway surface liquid (ASL), paracellular absorption, and the diffusion time through the mucus layer, amongst other things. From these *in vivo* measurements, an *in silico* model of ion and water transport across the epithelial tissue in CF [23] was previously developed in our group. We used these measurements as features for a statistical model to estimate the change in MCC rate coefficients and non-clearable activity from Chapter 3. This could provide clinicians a tool to aid in decision making about whether or not to include HS as part of a CF individual's treatment plan based on their specific expected benefit.

4.1.1 Thin-Film Model

Prior work from our group developed a tissue-scale model of water and ion transport across the epithelium based on the *in vitro* measurements from a subset of these study participants [23, 62]. The model simulated sodium, chloride, potassium, and water transport across the cell both transcellularly and paracellularly. The permeability and maximal fluxes of relevant ion channels and pumps were fit to Ussing chamber experiments and the water absorption was fit to 48 hours of ASL volume measurments. To distinguish between para- and transcellular pathways, Tc-DTPA – an *in vitro* equivalent to In-DTPA – retention was also measured over the same timeframe. The parameters were fit using a tool called APT-MCMC, which uses a Markov chain Monte Carlo technique with ensembles of affine-invariant samplers and parallel temperature swapping to quickly sample uncorrelated regions of parameter space [63]. A comparison of the best-fit parameter set for CF vs non-CF participants showed significantly increased permeabilities for transceulluar water channels, the epithelial sodium channel (ENaC), paracellular Tc-DTPA and significantly decreased permeability of alternative chloride channels (ACC) [62]. We hypothesized that by including these patient-specific *in silico* parameters for those who had them available, we could better estimate individual response to inhaled HS in the CF group.

4.1.2 Feature Selection

Because the initial set of variables, also called features, considered was larger than the number of participants, fitting linear model coefficients for all of them would result in gross overfitting. Reducing the number of features in the models and using cross-fold validation decreases the risk of overfitting, increasing the likelihood of producing accurate predictions when applied to new data sets. We use a combination of two methods to select the features for each statistical model: least absolute shrinkage and selection operator (LASSO) regularization and recursive feature elimination (RFE). LASSO adds an L_1 regularization term to the typical objective function minimized in linear regression, as expressed in Equation 4.1. This penalizes large magnitude coefficients, driving coefficients that do not also sufficiently reduce the L_2 term to zero [64].

$$\min_{\hat{\beta}} \frac{1}{2} \frac{\|\mathbf{y} - \mathbf{X}\hat{\beta}\|_2^2}{N} + \alpha \|\hat{\beta}\|_1 \tag{4.1}$$

RFE is a backwards elimination wrapper method that recursively removes the least important feature, as defined by the method it wraps around[65, 66]. In this case, the magnitude of the coefficients from LASSO regression provide a measure of relative importance [64]. To further ensure the appropriate features are selected, k-fold cross-validation can be implemented at each elimination step by using the mean importance across all folds to select a feature to remove.

4.2 Methods

4.2.1 Apply Reduced Model to CF Participants on IS and HS

First, we fit MCC rate coefficients for each of the clusters of the reduced model from Chapter 3 for CF participants on both IS and HS study days. We compared the clearance rates for CF participants on the IS day (CFIS) vs healthy controls (HC) using a two-sample Kolmogorov-Smirnov non-parametric test to determine which, if any, parameters were significantly different in the CF group. Patient-specific response in MCC parameters between the CF participants on their IS vs HS (CFHS) study days was compared using a paired Wilcoxon signed-rank test.

4.2.2 Statistical Model

The primary outcome we were interested in studying was whether the change in MCC rate coefficients and non-clearable activity could be estimated using other more readily available measurements. To that end, we built multivariable linear regression models to estimate each of these response variables. We broke our input variables into several groups based on how the measurements were gathered to assess how accurate of predictions we could get using different subsets of available data. These variables were broken into 3 groups: clinical (Table 6), *in vitro* (Table 7), and *in silico* (Table 8).

Table 6: Summary of clinical input variables considered for multivariate linear models (n=23, except †: n=20)

Variable	Abbreviation	Units	Description
Age	—	years	Demographics
Height	—	cm	
Weight	—	kg	
Body mass index	BMI	kg/m^2	
Forced expiratory volume in 1 sec	FEV_1	% predicted	Pulmonary function test (PFT)
Forced expiratory flow between 25% and 75% of vital capacity	$\text{FEF}_{25\text{-}75}$	% predicted	
Forced vital capacity	FVC	% predicted	
Change in FEV_1 from IS to HS day	ΔFEV_1	% predicted	PFT change
Change in $\text{FEF}_{25\text{-}75}$ from IS to HS day	ΔFEF_{25-75}	% predicted	
Lung clearance index†	LCI	—	Multibreath washout test measurement; Marker of lung inhomogeneity
Sweat Chloride	Sweat Cl	μM	Biomarker of CFTR function

In the case that measurements were not available for an individual, they were imputed using a weighted average of the values for the 3 nearest neighbors based on the remainder of parameter space. We chose to do this due to the small sample size, however it should be

Table 7: Summary of *in vitro* input variables considered for multivariate linear models. IR=Isotonic Ringer's, HR=Hypertonic Ringer's

Variable	**Abbreviation**	**Units**	**Description**
Normalized apical liquid volume absorption rate (n=16)	Norm. V_{ap}	%/day	Measure of liquid absorption rate
Absorbed Tc-DTPA (n=22)	Cell ABS	%/day	Marker of paracellular liquid absorption
Fluorescence recovery after photobleaching (n=10)	FRAP	τ/τ_{saline}	Marker of mucin concentration
Change in Norm. V_{ap} from IR to HR (n=16)	Δ(Norm. V_{ap})	%/day	Change in liquid absorption rate under HR conditions
Change in Cell ABS from IR to HR (n=16)	Δ(Cell ABS)	%/day	Change in paracellular liquid absorption under HR conditions

noted that this could introduce some bias in our model, particularly for measurements with missing values for a large number of participants.

4.2.3 Feature Selection

To avoid including redundant variables in our statistical model, we took a few steps to reduce the number of input variables or features included. All variables were centered to mean 0 and normalized to a standard of deviation of 1 prior to performing the remainder of feature selection. We calculated Pearson's correlation coefficients between each of the available features to determine their linear independence. For those that were highly correlated, only one of the two parameters was selected for use in the remainder of feature selection. In the

Table 8: Summary of *in silico* input variables considered for multivariate linear models (n=9)

Variable	Abbr.	Units	Description
Trans-apical water permeability	P_{wca}	μm/min	Transcellular water transport
Trans-basolateral water permeability	P_{wcb}	μm/min	Transcellular water transport
Paracellular water permeability	P_{wab}	μm/min	Paracellular water transport
ENaC permeability	P_{enac}	μm/min	Transcellular Na^+ transport
ACC permeability	P_{acc}	μm/min	Non-CFTR transcellular Cl^- transport
BKCa permeability	P_{bkca}	μm/min	Ca^+-activated K^+ transport
Basolateral Potassium permeability	P_{bK}	μm/min	Transcellular K^+ transport
Basolateral Chloride permeability	P_{bCl}	μm/min	Transcellular Cl^- transport
Paracellular Sodium permeability	P_{pna}	μm/min	Paracellular Na^+ transport
Paracellular Chloride permeability	P_{pcl}	μm/min	Paracellular Cl^- transport
Paracellular Potassium permeability	P_{pk}	μm/min	Paracellular K^+ transport
Paracellular DTPA permeability	P_{pdt}	μm/min	Paracellular small molecule transport
Maximum molar flux through NKCC	J_{nkcc}	mmol/(min·m^2)	Co-transport of Na^+, K^+, and Cl^-
Maximum molar flux through Na-K-ATPase	J_{nakp}	mmol/(min·m^2)	ATP-activated transport of Na^+ and K^+

case that one of the parameters had more available measurements (i.e. fewer missing values), this was preferentially selected out of the two.

Next, recursive feature selection was performed with 5-fold cross validation using scikit learn in Python [58] for each of the response variables (MCC rate coefficients and non-clearable activity change). We repeated the same process using the change in Tc-SC concentration at 10 minutes as the output variable to enable the simulation of the predicted MCC dynamics. During this process, LASSO regression was performed for each fold with the lowest coefficient getting removed during each iteration of the recursive feature elimination. The estimated values from these cross validated models were then compared to the fitted values for each individual to assess the accuracy of the multivariable model.

4.2.4 Validation with Organ-Scale Model

While our primary outcome was to predict the parameter values for the MCC model from Chapter 3, we can also evaluate the impact errors in these predictions have on the resulting MCC dynamics. Since the change in non-clearable activity was estimated on a cluster basis rather than for individual grids, we assumed the change in activity predicted was proportional for all grids in a cluster. From there, we simulated MCC retention using the predicted MCC rate coefficients, Tc-SC concentration at 10 min, and non-clearable activity for each individual and compared the average grid MAE and whole lung MAE to that of the fitted MCC model.

4.3 Results

4.3.1 Response to HS from Fitted Parameters

The distribution of fitted MCC rate coefficients for each group (HC, CFIS, and CFHS) are presented in Figure 23. HC and CFIS parameters were compared using non-parametric two-sample Kolmogorov-Smirnov tests to determine if there were any significant differences between the values for the groups. Only cluster 3, which approximately overlays the smaller airways of the lower lobe, had a significantly different parameter distribution in the CF group than in HCs. The distribution of MCC rate coefficients between CFIS and CFHS groups was compared using a paired Wilcoxon signed-rank test. There were significant increases from CFIS to CFHS for MCC rate coefficients in all clusters. Some of the parameters for the CFHS group even exceeded the range observed in the HCs.

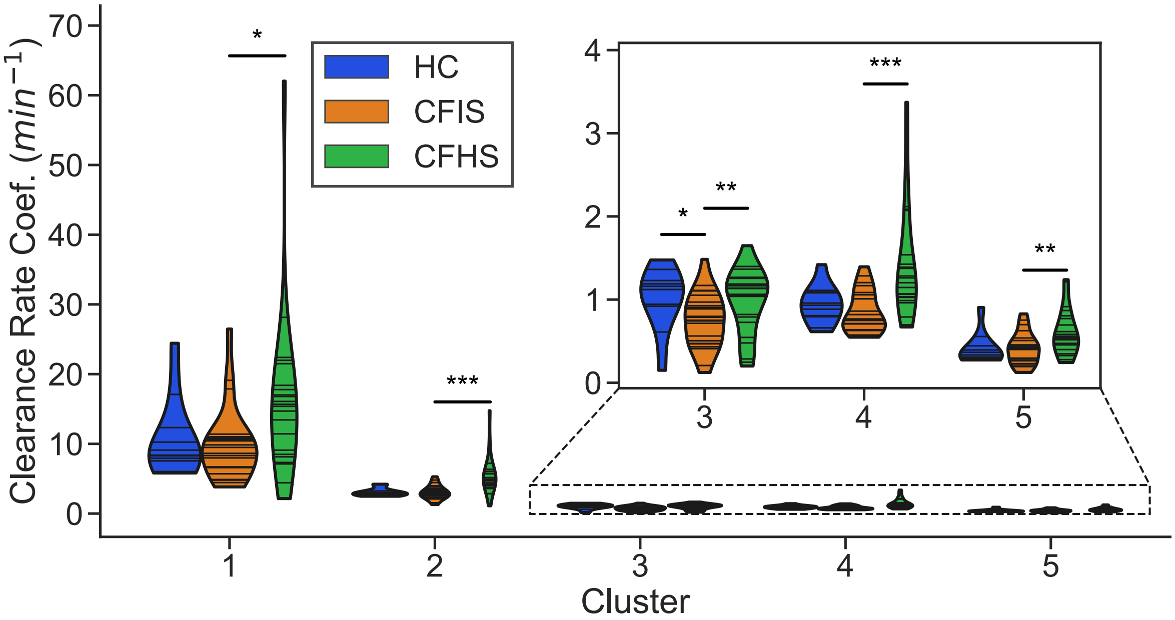


Figure 23: Clearance rate coefficient distribution by subject subgroup (HC=healthy controls including inhalation of isotonic saline (IS), CFIS=CF including inhalation of IS, CFHS=CF including inhalation of hypertonic saline (HS)) and cluster. CFIS and CFHS are paired data from the same subjects on two different study days. For each violin, the width of the shaded region shows the estimated probability density of the corresponding parameter values; each horizontal line corresponds to the fitted parameter value for an individual. HC vs CFIS were compared using a two-sample Kolmogorov-Smirnov statistical test and CFIS vs CFHS were compared using a paired Wilcoxon signed-rank test (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$)

The distribution of non-clearable activity in each cluster was also estimated for each group and is shown in Figure 24. The same statistical tests were used to compare the distributions between groups. The only significant difference in non-clearable activity between the HC and CFIS group was in cluster 5, which corresponds to the most distal regions in 2D. In contrast, the distribution of non-clearable activity decreased significantly from the IS to HS days in all clusters except cluster 5. The overall range of observed values in CF participants was also much larger than in the HC group.

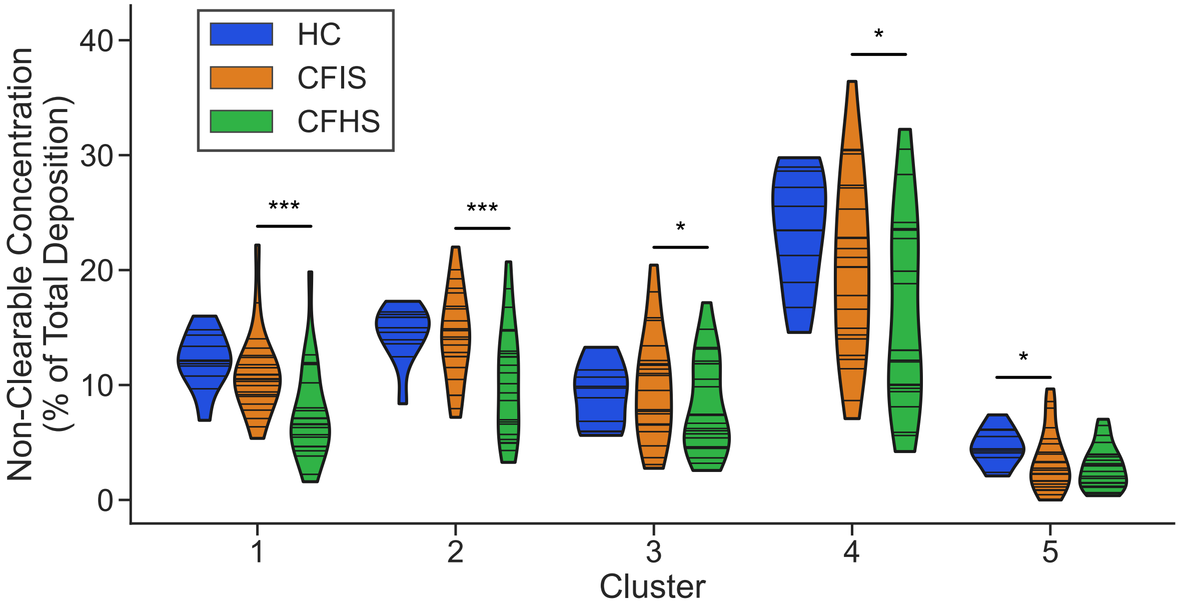


Figure 24: Distribution of non-clearable activity by subject subgroup (HC=healthy controls including inhalation of isotonic saline (IS), CFIS=CF including inhalation of IS, CFHS=CF including inhalation of hypertonic saline (HS)) and cluster. CFIS and CFHS are paired data from the same subjects on two different study days. For each violin, the width of the shaded region shows the estimated probability density of the corresponding non-clearable activity; each horizontal line corresponds to the value for an individual. HC vs CFIS were compared using a two-sample Kolmogorov-Smirnov statistical test and CFIS vs CFHS were compared using a paired Wilcoxon signed-rank test (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$)

4.3.2 Feature Correlations

The Pearson's correlation coefficients for all of the initially included input variables was calculated and any pair with a coefficient magnitude greater than 0.67 had one variable removed. Height and weight were strongly correlated (r=0.69), as well as weight and BMI (r=0.94). Because BMI implicitly includes both of these measurements, height and weight were removed. FEV_1 was highly correlated with FVC (r=0.92) and $FEF_{25\text{-}75}$ (r=0.82). Because FEV_1 is more commonly reported as a PFT, we chose FEV_1 out of these PFTs. The change from IS to HS study day in $FEF_{25\text{-}75}$, however, only showed weak correlation with FEV_1 or its change between study days, so it was included. The change in normalized apical volume absorption rate (ΔNorm. V_{ap}) was highly correlated (r=-0.78) with the value under IR conditions, so only the IR condition was included. Four *in silico* parameters were removed because they had strong correlations with *in vitro* measurements, which contain less uncertainty: P_{pna} with FRAP (r=-0.95), P_{wca} with Norm. V_{ap} (r=-0.68), P_{bk} with Cell ABS (r=0.79), and P_{enac} with FRAP (r=-0.90).

4.3.3 Linear Regression Estimates for Organ-Scale Parameters

RFE with 5-fold cross validation was performed, using LASSO regression coefficients to determine relative feature importance for all of the response variables: change in MCC rate coefficient, initial deposition, and non-clearable activity for each cluster from IS to HS study day. The resulting multivariable models when including all non-correlated input variables are summarized in Tables 9-10. For the selected features and coefficients for the submodels using only clinical and *in vitro* features, please refer to Appendix C. The relative importance was calculated for the features selected in each model as the magnitude of the LASSO regression coefficients and are presented in Figures 25-26.

Parity plots comparing the estimated change in values with the actual change in values are presented in Figures 27-29 for the MCC rate coefficients, initial deposition, and non-clearable activity, respectively. An adjusted coefficient of variation (Adj. R^2) was calculated for each fold during cross-validation and are shown on the corresponding parity plots.

Table 9: Coefficients and intercept for multivariable linear regression models to estimate change in MCC rate coefficients from IS to HS days.

		Response Variable (min^{-1})				
		Δk_1	Δk_2	Δk_3	Δk_4	Δk_5
Feature	**Units**	**Coefficient**				
FEV_1	% of predicted	-0.498	-0.133	-0.0208	-0.0120	—
P_{pcl}	μm/min	—	—	-0.382	-0.185	-0.114
Sweat Cl	μM	—	—	0.0174	—	4.44×10^{-3}
P_{bkca}	μm/min	—	—	0.463	—	—
LCI	—	-2.58	-0.606	—	—	-0.0153
ΔCell ABS	% cleared/day	—	—	—	-0.0281	-0.0207
FRAP	τ/τ_{saline}	8.89	1.72	—	—	—
J_{nkcc}	mmol/(min·m^2)	—	—	-0.0248	—	—
P_{acc}	μm/min	—	—	-1.39	—	—
BMI	kg/m^2	—	—	-0.0391	—	—
ΔFEV_1	% of predicted	—	—	—	—	0.0211
Age	years	—	—	—	—	0.0100
$\Delta\text{FEF}_{25\text{-}75}$	% of predicted	-0.137	—	—	—	
Norm. V_{ap}	%/day	—	5.71	—	—	—
J_{nakp}	mmol/(min·m^2)	—	—	—	—	8.13×10^{-3}
Intercept	—	34.9	9.03	1.71	1.28	-0.798

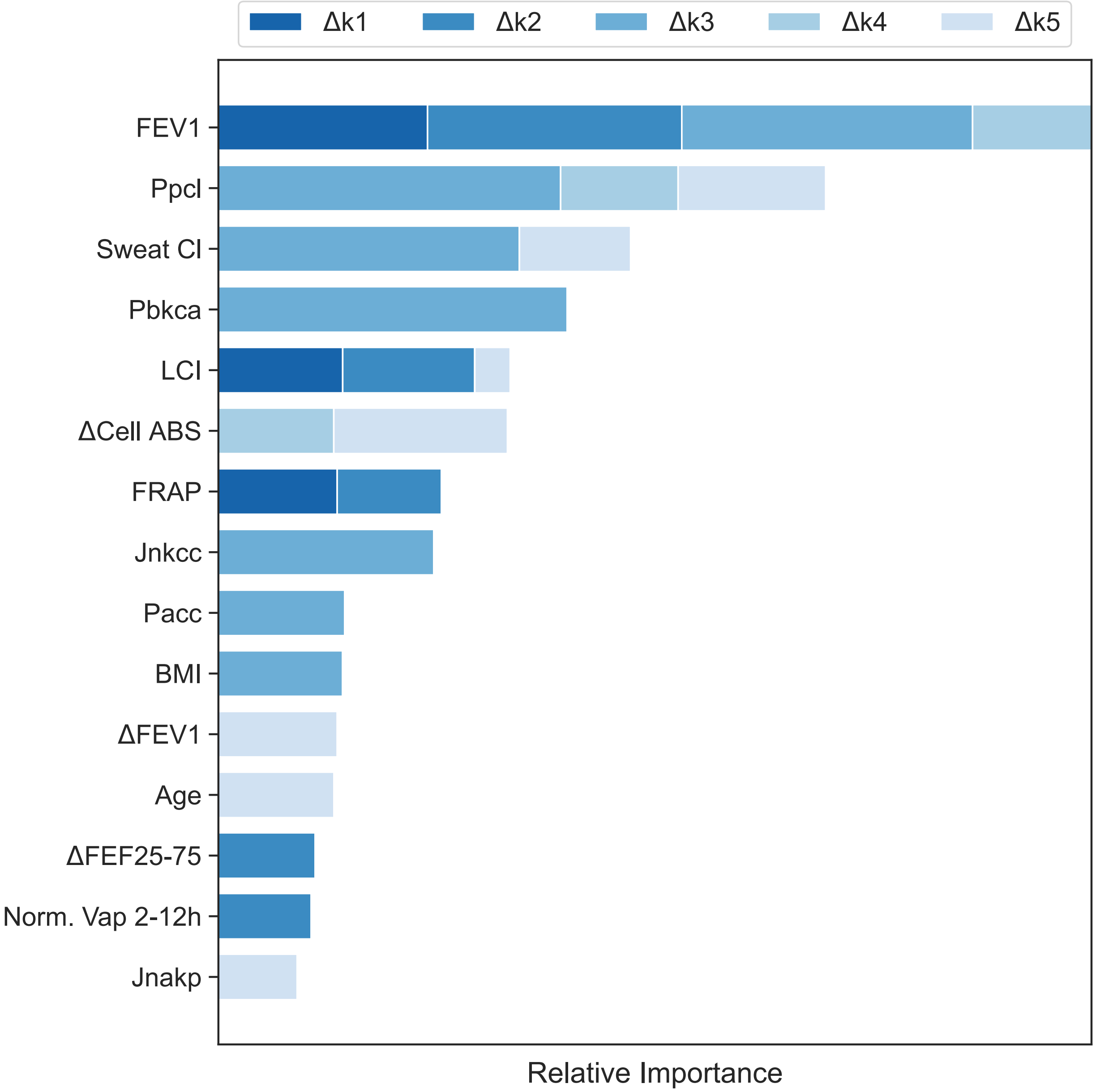


Figure 25: Relative importance of included features in predicting the change in MCC rate coefficients from IS to HS days in each cluster. Relative importance is determined by the relative magnitude of LASSO regression coefficients.

Table 10: Coefficients and intercept for multivariable linear regression models to estimate change in non-clearable activity from IS to HS days.

		Response Variable (% Total Activity)				
		ΔN_1	ΔN_2	ΔN_3	ΔN_4	ΔN_5
Feature	**Units**	**Coefficient**				
FRAP	τ/τ_{saline}	-2.44	-2.35	1.97	-7.16	-1.94
$\mathrm{P}_{\mathrm{acc}}$	μm/min	20.0	12.6	-7.14	14.4	—
BMI	$\mathrm{kg/m^2}$	0.279	—	0.250	-0.427	-0.223
Age	years	-0.328	—	—	-0.248	
LCI	—	1.08	—	—	1.47	-0.165
$\mathrm{P}_{\mathrm{wcb}}$ $(\times 10^{-4})$	μm/min	-4.32	-2.73	—	2.82	—
Cell ABS	% cleared/day	—	-0.126	0.303	-0.0977	—
$\mathrm{P}_{\mathrm{pcl}}$	μm/min	—	—	1.75	—	-0.569
FEV_1	% of predicted	—	—	0.0705	—	-0.0577
Norm. V_{ap}	%/day	—	—	—	21.4	4.03
ΔCell ABS	% cleared/day	—	—	0.310	0.158	—
$\Delta\mathrm{FEV}_1$	% of predicted	—	—	-0.465	—	—
Sweat Cl	μM	-0.0659	—	—	-0.0867	—
P_{pk}	μm/min	2.13	1.38	—	—	—
$\Delta\mathrm{FEF}_{\text{25-75}}$	% of predicted	—	—	0.220	-0.110	—
$\mathrm{P}_{\mathrm{bkca}}$	μm/min	—	—	—	-3.22	—

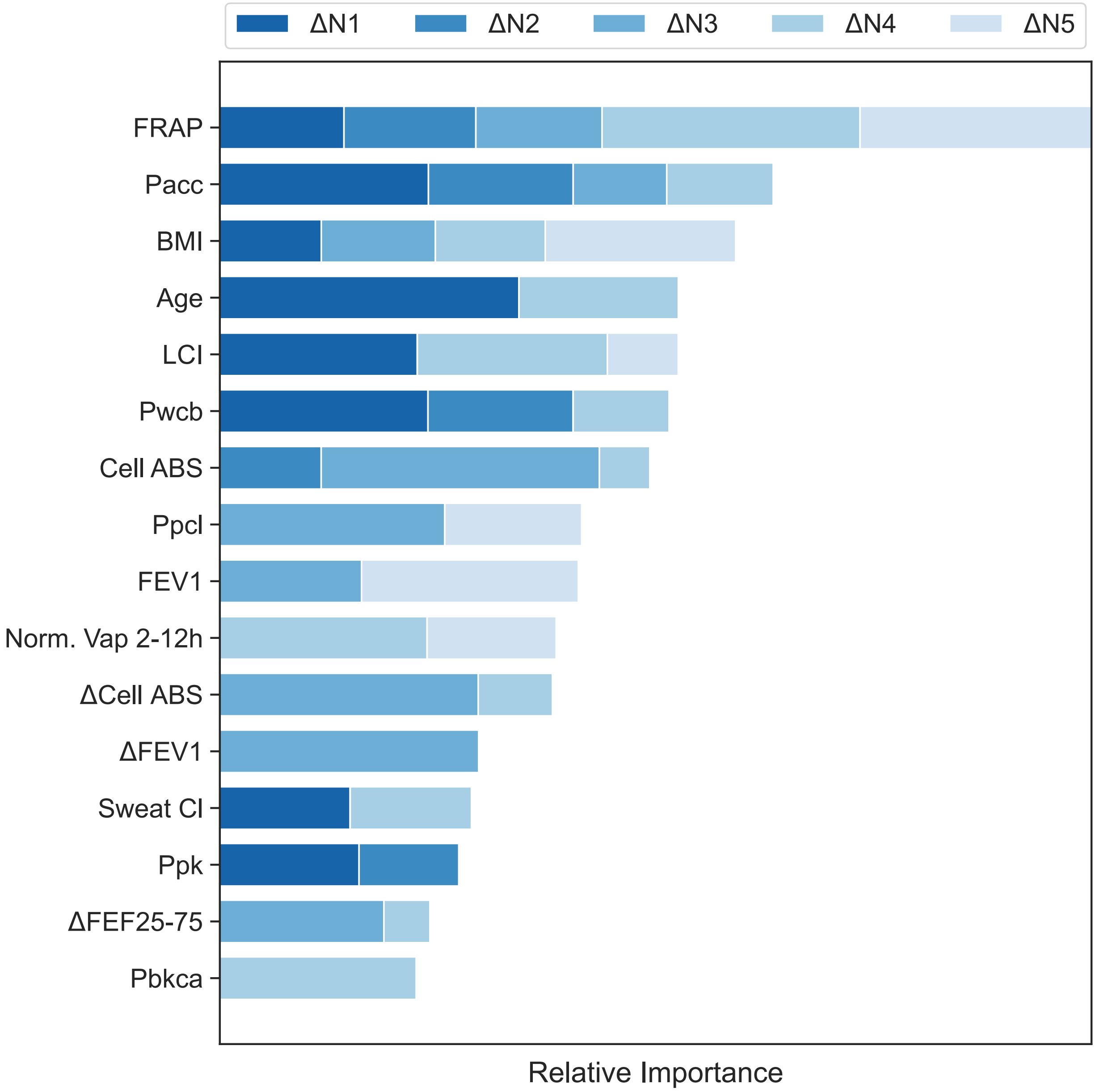


Figure 26: Relative importance of included features in predicting the change in non-clearable activity from IS to HS days in each cluster. Relative importance is determined by the relative magnitude of LASSO regression coefficients.

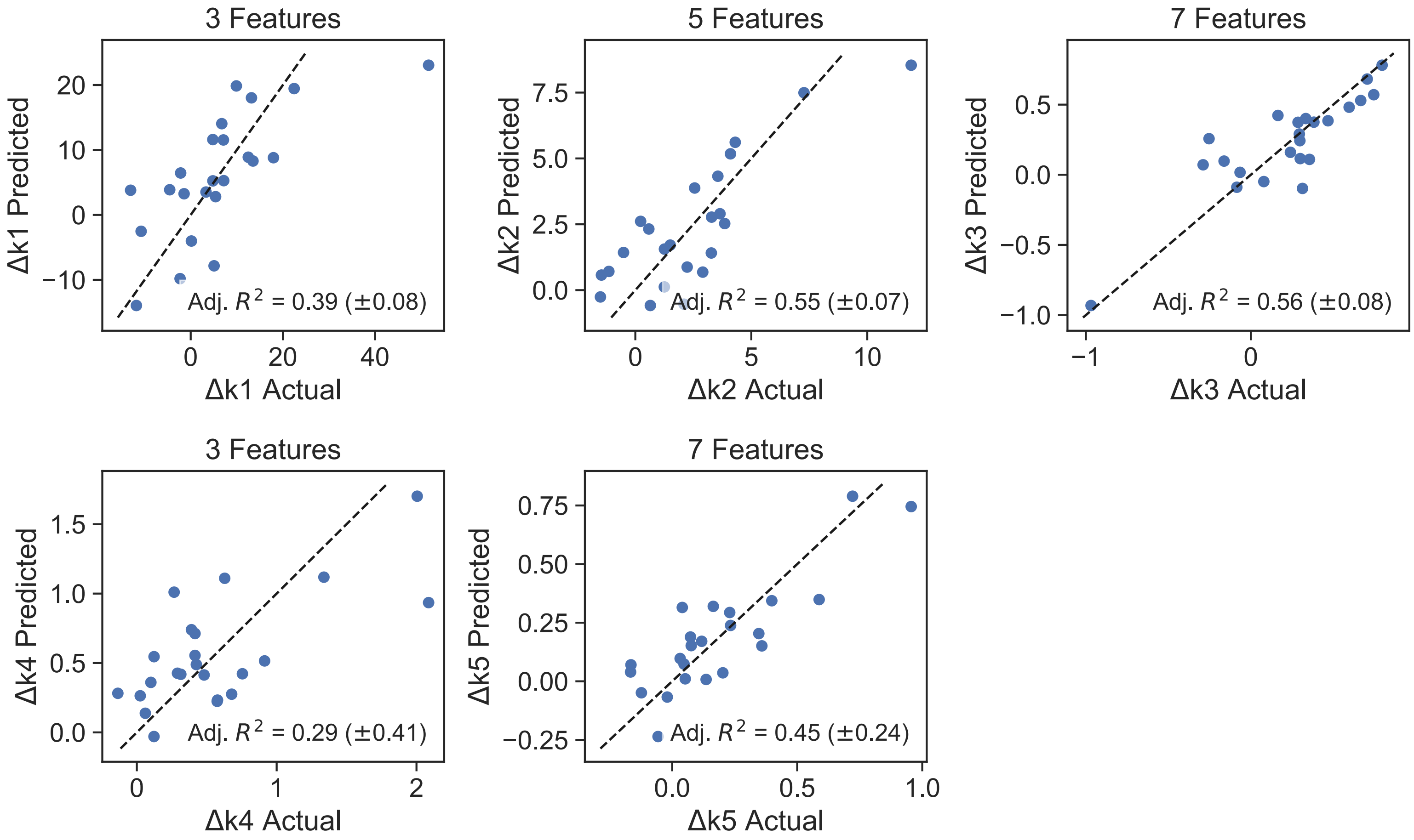


Figure 27: Comparison between the estimated (y-axes) and actual (x-axes) change in MCC rate coefficients from IS to HS study days for each cluster. Each circle corresponds to a CF individual and the dashed line indicates parity between estimated and actual values.

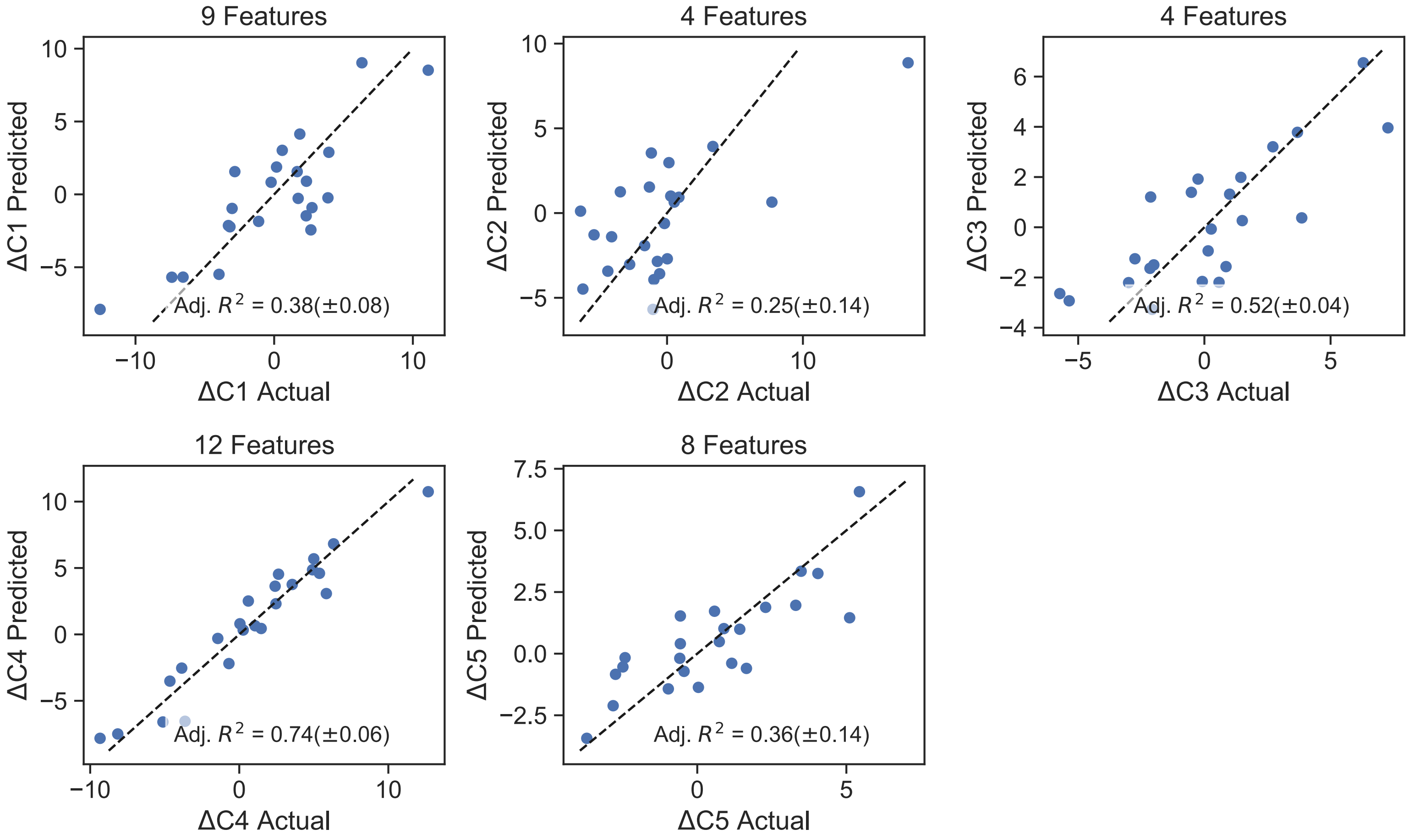


Figure 28: Comparison between the estimated (y-axes) and actual (x-axes) change in initial Tc-SC deposition from IS to HS study days for each cluster. Each circle corresponds to a CF individual and the dashed line indicates parity between estimated and actual values.

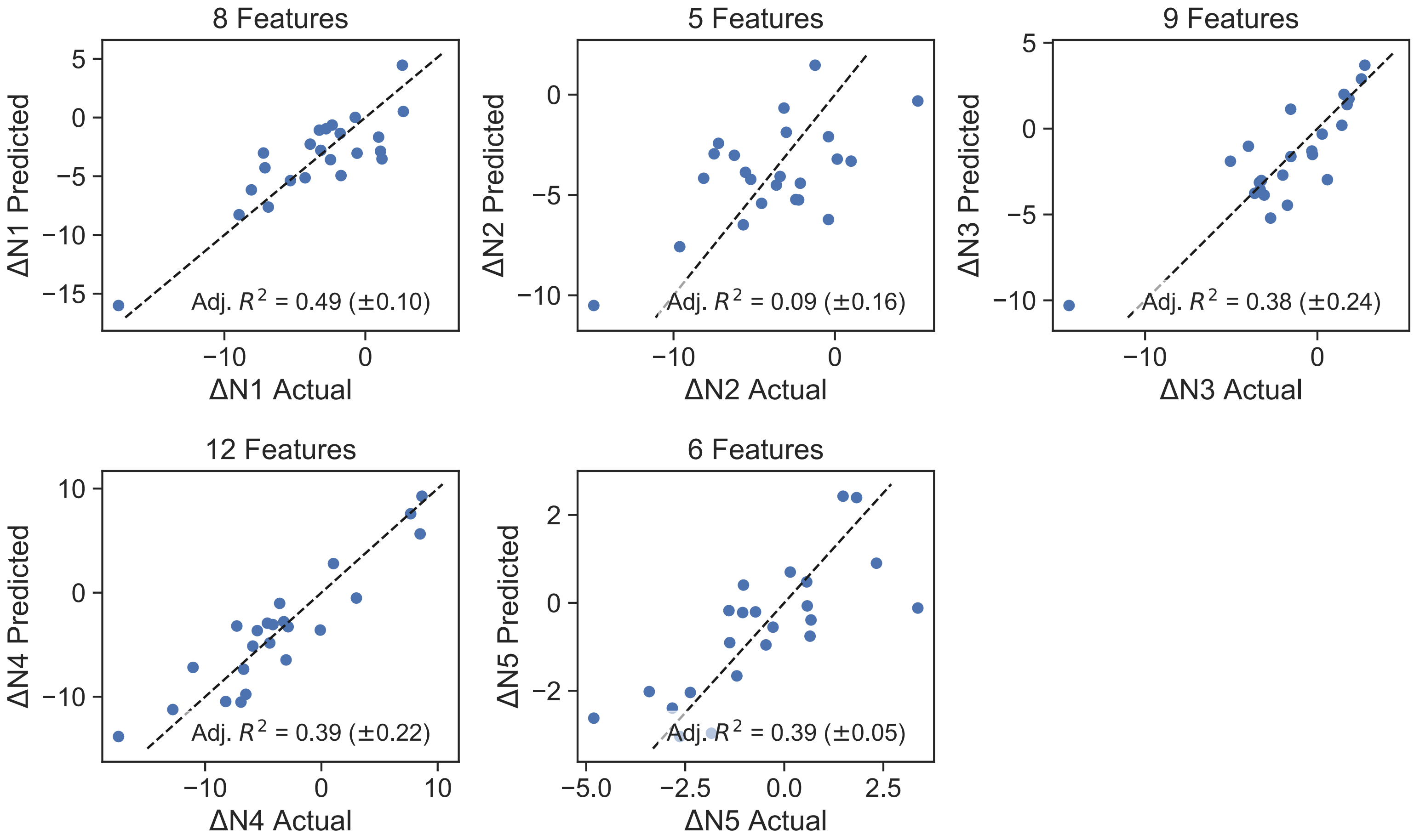


Figure 29: Comparison between the estimated (y-axes) and actual (x-axes) change in non-clearable activity from IS to HS study days for each cluster. Each circle corresponds to a CF individual and the dashed line indicates parity between estimated and actual values.

4.3.4 Comparison of Model Dynamics

MCC dynamics were simulated using the IS day parameters for each CF individual and the predicted changes in those parameters, deposition pattern, and non-clearable activity. The mean grid MAE for the predicted trajectories relative to the measured HS day Tc-SC were compared for the three different subsets of variables used to generate statistical models. These are summarized in Table 11. Whole lung trajectories for the model including all variables are shown in Figure 30 for the participant with the largest and smallest whole lung MAE to illustrate qualitatively the range of error. Error in the amount of non-clearable activity appears to have a large impact. When the actual values are used for non-clearable activity and initial deposition, but the MCC rate coefficients are predicted, the grid MAE and whole lung MAE drop to 0.12 ± 0.037 and 4.6 ± 2.1, respectively.

Table 11: Model error including different subsets of features

Included Variables	**Mean Grid MAE**	**Whole Lung MAE**
All (Clinical, *In Vitro*, and *In Silico*)	0.18 ± 0.046	7.6 ± 2.9
Clinical + *In Vitro*	0.19 ± 0.057	9.0 ± 4.8
Clinical	0.20 ± 0.051	9.7 ± 6.6

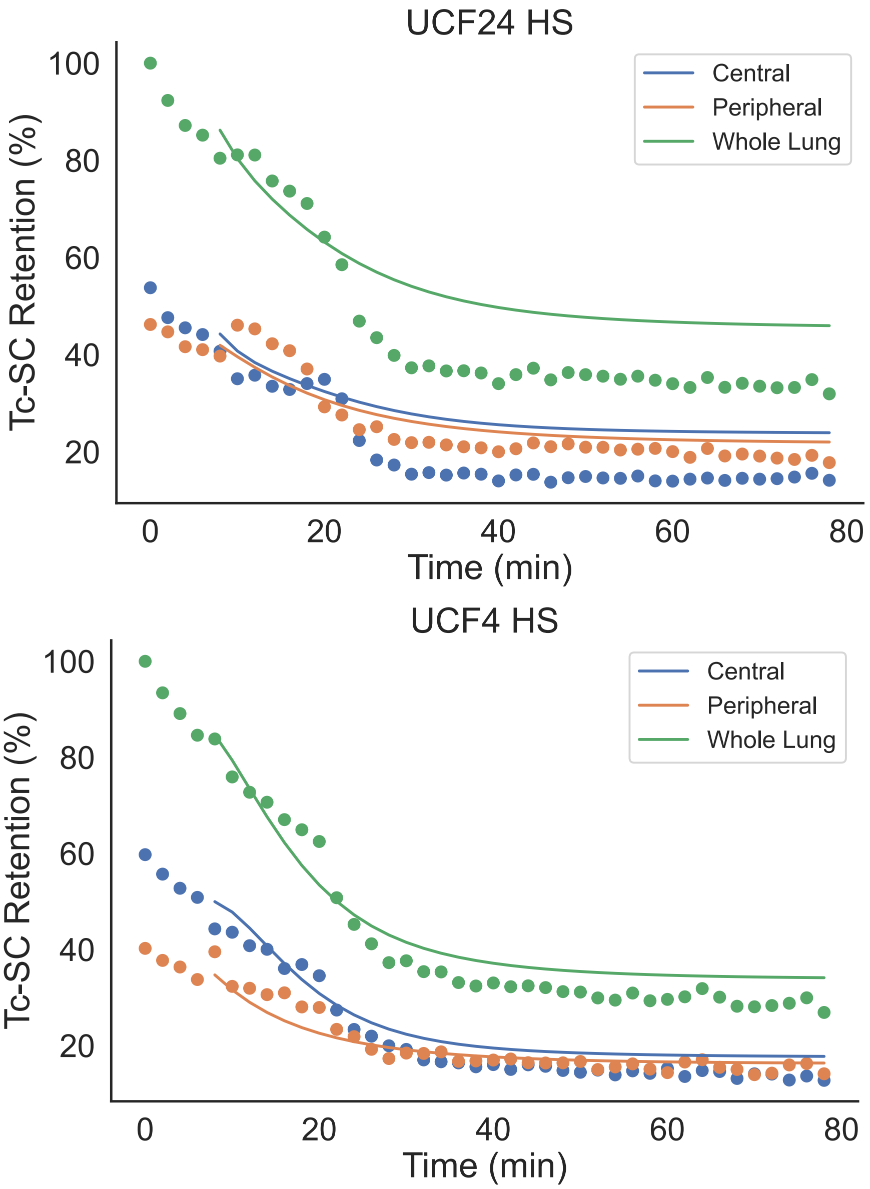


Figure 30: Simulated whole lung MCC trajectories using predicted values for the CF individuals with the largest (top) and smallest (bottom) MAE between simulation and measured Tc-SC retention.

4.3.5 Binary Classification from Regression Models

While the values estimated from the linear regression models for the change in clearance rate coefficients after inhaled HS were pretty accurate, the estimated change in the non-clearable activity was less accurate for many clusters. Though this causes large errors for some subjects in the simulated versus measured MCC dynamics, it may still be useful in classifying participants as responders or non-responders. To evaluate this potential, we considered binary classifier models for the change in clearance rate coefficient and non-clearable activity for each cluster. A response for the change in clearance rate coefficients was considered to be anything greater than 0, while a response for the change in non-clearable activity was considered to be anything less than 0. Receiver operating characteristic (ROC) curves were generated by calculating the true positive rate (TPR) and false positive rate (FPR) for different threshold values of the estimated response variables to classify responders (positive) versus non-responders (negative) and are presented in Figures 31 and 32. These highlight trade-offs between detecting more of the positive results and misclassifying more of the negative. For these curves, a perfect classifier would have a single point at (0, 1) and have an area under the curve (AUC) of 1. A classifier with no skill would be linear with slope 0.5 (shown as a dashed line in Figures 31 and 32) and would have an AUC of 0.5. The higher the AUC is for a classifier, the higher its skill. The average AUCs were 0.83 for classifying response of rate coefficients and 0.87 for classifying response of non-clearable activity to HS. ROC curves that cross the no skill line are caused by large gaps in data and small sample size (i.e. switching one individual from a positive to a negative has a large impact).

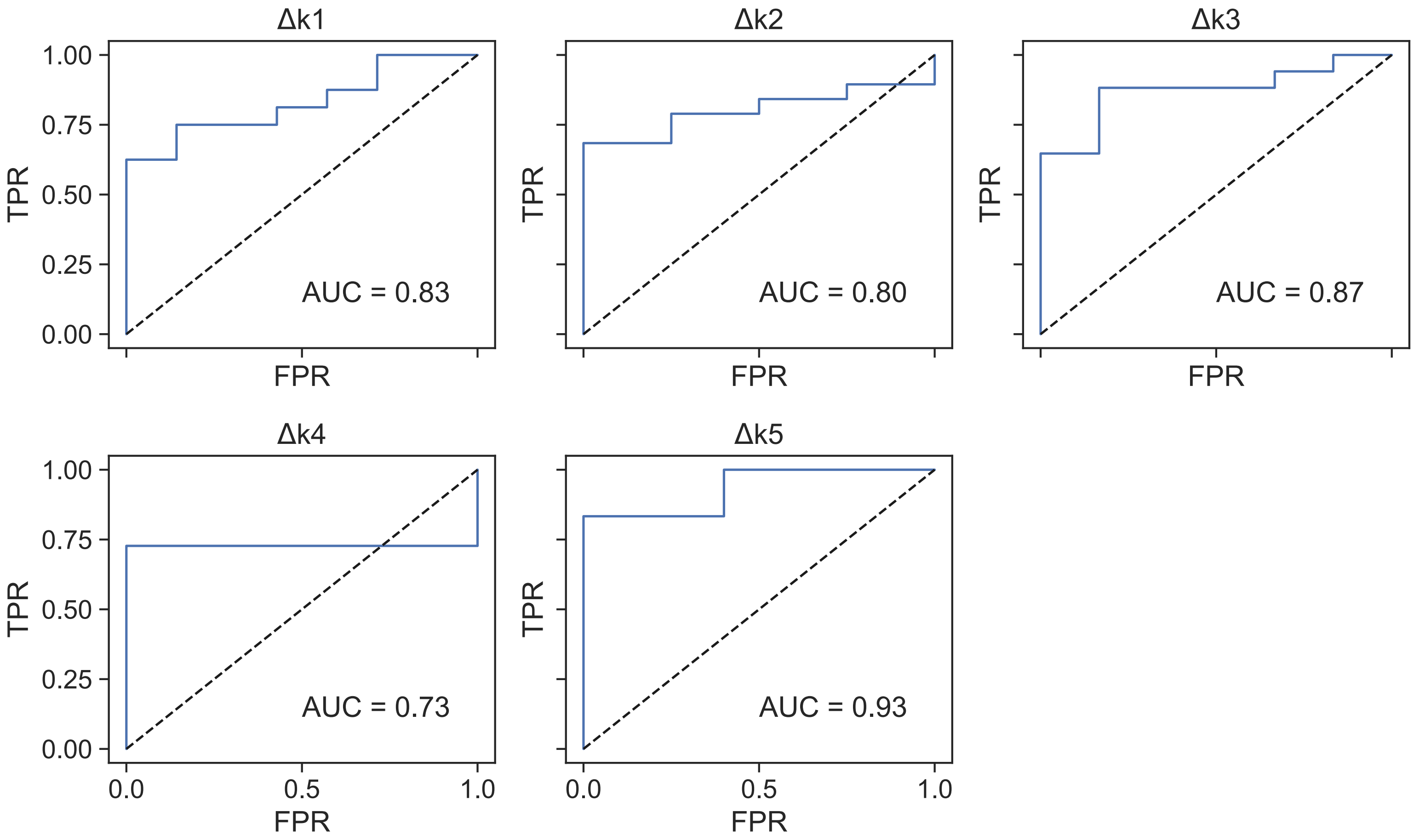


Figure 31: ROC curves for binary classification of response of rate coefficients to HS for each cluster. The solid blue line represents the binary classifier for different threshold values. The dashed line represents a classifier with no skill.

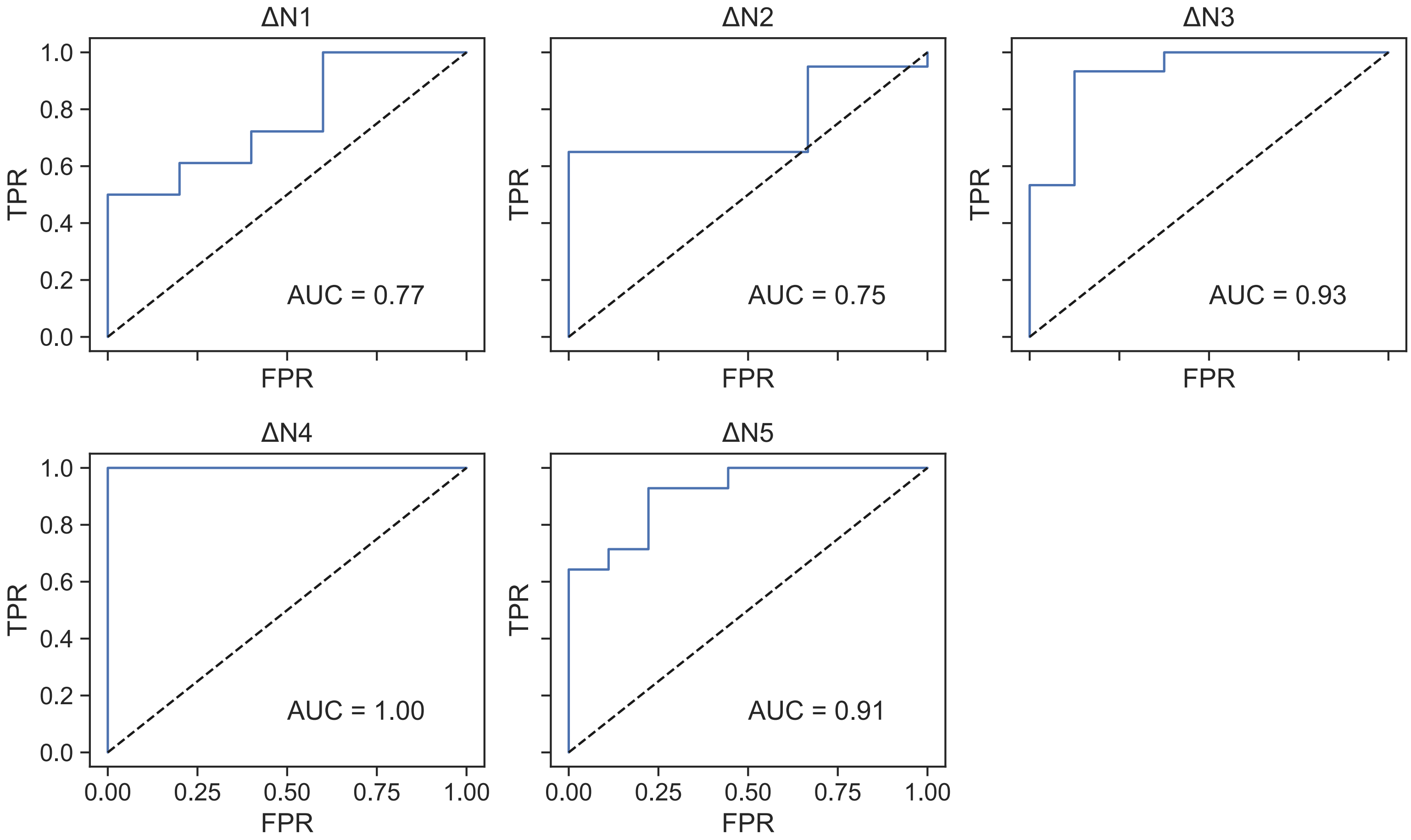


Figure 32: ROC curves for binary classification of response of non-clearable activity to HS for each cluster. The solid blue line represents the binary classifier for different threshold values. The dashed line represents a classifier with no skill.

4.4 Discussion

4.4.1 Comparison Between CF and Healthy Individuals

We found that of the five characteristic regions identified in Chapter 3, the distribution of clearance rate coefficients was only significantly different between healthy controls and CF subjects in one of those regions — the one corresponding to the lower lobe. Previous work describing this data showed no significant difference in whole lung MCC between CF and HC, when adjusted for percent central deposition [34]. This points to our model as possibly a more sensitive tool for detecting localized changes to MCC. While previous studies have consistently found evidence of worse disease in the right vs left lung, the results are highly varied as to whether there is any difference in disease severity between lobes [67–70]. Although those studies did not specifically look at MCC as a marker of regional disease severity, one would expect regions of depressed MCC to accompany those with high degrees of bronchiectasis, air trapping, inflammatory markers, and bacterial loads. These results suggest that MCC is slowed most in the middle or lower lobe in CF, which is consistent with findings of increased air trapping in the middle and lower lobes relative to the upper lobe in CF [70, 71].

One reason we may not see significantly depressed MCC in other regions may be related to how we process the non-clearable activity. Since the clearance rate coefficients are only fit to mucus that clears over the course of the 80 minute study window, it is possible that in regions with high degrees of mucus plugging, the radiolabeled aerosol simply gets trapped and never clears. If this were the case, one would expect to see a significant increase in the amount of non-clearable Tc-SC relative to the healthy controls, which we did not observe (see Figure 24). It is alternatively possible that due to decreased ventilation, no aerosol deposits in regions with very severe disease. Since all subjects had some clearable mucus observed over the course of the experiment in each cluster, however, we think the impact of this on our results is minimal.

4.4.2 Response to HS in CF Group

Clearance rate coefficients increased significantly after inhaled HS in every cluster in the CF group. This is consistent with studies showing increases in whole lung MCC in CF subjects following inhaled HS [31, 33, 34, 61, 72, 73]. Interestingly, some of these coefficients exceed the clearance coefficients observed in healthy subjects. While this would seem to suggest HS could enhance MCC capabilities of CF subjects beyond those of healthy subjects, it is important to note that these coefficients only capture the speed of mucus clearance and not the overall amount or the duration of the effect. Thus, although HS may increase the speed of mucus clearance, it may still be insufficient to clear accumulated mucus as well as the ongoing rate of mucus production in CF. As for the increased speed, this may be linked to the concentration of mucins in CF versus healthy subjects. Indeed, previous work from this study showed a significant correlation between fluorescence recovery after photobleaching (FRAP) in human nasal epithelial cells – a measure of diffusion time through mucus – and the same subject's change in adjusted whole lung MCC after inhaled HS [34]; it indicates that higher concentrations of mucins may be associated with greater response to HS. This notion is consistent with observations that although CF subjects demonstrated both acute and prolonged increases in whole lung MCC after inhaled HS [31, 33, 72], healthy subjects only show an acute response lasting less than 4 hrs [32, 74]. It may be that mucin concentrations are depleted after HS treatment in healthy controls, but due to the increased initial concentration and rate of production of mucins, we continue to see an effect in CF subjects for much longer.

Significant decreases in non-clearable activity were also observed between baseline and HS study days in all but cluster 5 (the most peripheral region). This suggests that HS can acutely increase the total amount of mucus that can be cleared in addition to the speed of clearance. The difference was most pronounced in the regions overlapping the mainstem bronchus and bronchus intermedius. Since these regions also show the fastest MCC rates, the combination results in a dramatic increase in the observed acute whole lung MCC. The increase in clearable activity in clusters 3 and 4 is more moderate, suggesting that rehydration of mucus may be less effective in these regions. This could be due to differences in the proportion of the surface that is ciliated in the large airways, relative to the small airways [15]. Thus, more of the cluster has the potential to exhibit rehydration from HS.

4.4.3 Outcomes of Statistical Model

Multivariable linear regression models were fit in order to estimate individualized changes in MCC rate coefficients and non-clearable activity from IS to HS study days using matched clinical, *in vitro*, and *in silico* variables. The accuracy was not excellent (highest adjusted R^2 of 0.56 and 0.49 for the rate coefficients and non-clearable activity, respectively), indicating that nuclear imaging is still the most precise way to measure individual response to HS. However, they were sufficiently accurate to classify CF participants as "responders" or "non-responders" for each cluster. While further training on a larger population, as well as out-of-sample validation, would be necessary to have enough confidence in these classifiers to implement clinically, this highlights the feasibility of predicting patient-specific response without the need for nuclear imaging. This is important for the development of a screening tool since nuclear imaging requires a lot of time, access to nuclear imaging facilities and personnel, and exposes participants to radioactivity. Although the *in vitro* and *in silico* measurements take longer overall in order for cell lines to grow out, the burden on the part of a CF participant would be much less; the nasal scraping to harvest the cells only takes a few minutes and the participant would not be exposed to radioactivity.

Because the L_1 term of LASSO regression penalizes non-zero coefficients, we can use the magnitude of those coefficients to assess the relative importance of different features in estimating the response variables. These models are not mechanistic in nature, so we cannot derive any causal relationships. However, by analyzing the collection of features that are important, we can get an idea of what factors contribute to heterogeneity of individual response.

For predicting the change in MCC rate coefficients, FEV_1 was found to be an important feature for all but the most peripheral cluster. FEV_1 is a common marker for large airway function [75] and has a negative coefficient for these clusters. It makes sense that a participant with higher baseline large airway function would see less of a response since there would likely be less dehydrated mucus to rescue. The magnitude of the coefficients also decrease the more peripheral the cluster, which makes sense because FEV_1 is more sensitive to changes in large airway function than small airways.

The change in rate coefficients for clusters 1 and 2, which correspond to the large airways, were also largely impacted by LCI, with a negative coefficient, and FRAP, with a positive one. LCI is a marker of ventilation inhomogeneity, is more sensitive to small airway changes, and has been suggested for use as an early marker of airway disease [76, 77]. The negative relationship here seems counter intuitive since one might expect higher delivery of HS to the large airways with decreased small airway ventilation and therefore a larger response for the large airway coefficients. One reason for this result may be due to the link between higher LCI and disease severity. Prolonged mucus dehydration may have led to an accumulation of mucus, such that even with HS, cilia cannot clear the mucus any faster. The effect of FRAP makes more sense intuitively since it is thought to be related to the concentration of mucins in the mucus [34, 78, 79]. Drawing water into the mucus through inhalation of HS would decrease this mucin concentration. For an individual with already low concentrations, this would have a diminishing benefit, while an individual with high mucin concentration may become much less viscous.

The change in MCC rate coefficient in cluster 3, which corresponds to the area of the lower lobe, has a number of important features related to chloride and potassium transport. While the complexity of tissue-scale ion transport inhibits direct interpretation of these coefficients, as a collection, they suggest a higher dependence of the response in this region to differences between individuals in osmotic driving forces. This may be due to thinner layers of mucus in the smaller airways, such that mucus properties, such as viscosity and mucin concentration, are less important in establishing a change in MCC speed. Interestingly, this was the same cluster that was found to be significantly different between the HC and CFIS groups, as shown in Figure 23. The CFIS distribution is much broader than the HC group, but the CFHS distribution has the same shape as the HC. This points to osmotic imbalances perhaps playing a larger role in depressed MCC in this region than in other regions.

In predicting the change of non-clearable activity from IS to HS days, FRAP was an important feature in every cluster, with the upper lobe and far peripheral clusters exhibiting more of a dependence. This makes sense since highly concentrated mucus can cause cilia to collapse. By rehydrating mucus with HS, cilia function can be restored, thus decreasing the amount of mucus that is non-clearable. The one cluster where the coefficient has the opposite sign is cluster 3. Again, this may indicate a response in this cluster primarily driven by osmotic forces across the membrane, rather than mucus properties. This notion is further supported by the high relative importance of Cell ABS, $\mathrm{P}_{\mathrm{pcl}}$, and Δ(Cell ABS). Collectively, these indicate the change in non-clearable activity after HS in this region may be related more to a rehydrated PCL than to the mucus layer above it. This could also explain why the response in clearance rate coefficients after HS does not increase beyond the range observed in HCs. If there is less accumulated mucus in this region, there may be a decrease in the concentration of mucins after HS, thus limiting how much the speed of clearance could increase. This is consistent what Bennett et al. [32] proposed as to why healthy individuals did not exhibit prolonged response to HS.

Finally, we were interested in understanding which variables were necessary for predicting these variables to better understand which measurements to prioritize in future studies. We looked at this by considering different subsets of features that were considered for the models. In particular, we found that there was little increased error when *in silico* parameters were not considered in terms of predicting change in MCC rate coefficients or non-clearable activity. The largest decreases in adjusted R^2 were from 0.56 to 0.35 in cluster 3 and and from 0.39 to 0.21 in cluster 4 for the rate coefficients and non-clearable activity, respectively. Additionally, binary classifiers based on the predicted response variables had average AUCs of 0.83 (same as when *in silico* parameters were included) and 0.80 (a decrease of 0.07) for the change in MCC rate coefficients and non-clearable activity, respectively. The only place where removing *in silico* parameters had a large impact was in predicting the change in Tc-SC concentration at 10 minutes, which was used to simulate the predicted HS day MCC dynamics. While all the adjusted R^2 values for predicting concentration at 10 minutes were above 0.25 when including *in silico* parameters, only two clusters (3 and 5) reached that level when *in silico* parameters were removed. Since a screening tool would likely not include nuclear imaging, concentrations and non-clearable activity for an IS baseline day would not be available for generating those simulations, regardless.

5.0 Summary & Future Work

5.1 Contributions

5.1.1 New Methods for Analyzing MCC *In Vivo*

Prior work has primarily focused on static measurements across large areas of the lung to assess MCC capabilities of an individual [10, 30–34, 43, 74, 80]. However, these cannot account for interpatient variability in deposition and lump very heterogenous areas of the lung together. Attempts at analyzing MCC at a more granular level did not account for activity transported from more distal regions of the lung to more proximal airways [36]. Conversely, a previous dynamic model of MCC at the population level uses two large regions of interest (ROIs) that are very heterogeneous when applied to individuals and thus cannot accurately capture patient-specific MCC [35].

The work in this dissertation lays the groundwork for a much more detailed analysis of MCC than was previously possible. This was accomplished by breaking nuclear imaging scans into a grid of much smaller regions of interest than are typically used. Rather than just look at endpoint measurements in these ROIs, a dynamic model was developed based on average airway anatomy and known physiology of how mucus clears up the airway tree. This allowed for the detailed description of MCC across the entire lung. Additionally, the fitted parameters from this model are not confounded by the deposition pattern of the aerosol as it is with traditional whole lung MCC analysis. It therefore provides metrics of MCC that get at underlying mechanisms rather than differences in uncontrollable factors in the study.

Additionally, applying this model to extract patient-specific clearance rate parameters does not require any changes to typical planar scintigraphy protocols. This is important because it means that the analysis can be performed retrospectively to gain new insight from previous imaging studies or added to future studies. It has been largely automated in Python and the manual steps in ImageJ are similar to standard analysis methods, enabling end-users who are not well-versed in mathematical modeling to take advantage of the tool.

This increases the useful information that can be obtained with minimal cost. This provides a path to re-analyzing data from old studies to greatly expand our existing understanding of MCC both in health and in various diseases.

5.1.2 Insight Into Key MCC Regimes

We were able to accurately capture MCC dynamics of all study participants with only 5 clearance rate coefficients. This was enabled by first identifying directions of high interpatient variability through PCA and then clustering areas of low intrapatient variability (via hierarchical clustering of eigenlungs). This provides new insight into nominal variability in MCC studies and identifies areas of the lung that may vary independently in their rate of MCC.

For example, the first eigenlung, representing the direction of maximum interpatient variability across healthy controls, was in the magnitude of difference between the large and small airway MCC rates. Thus while there is consistently faster MCC in the large airways, how much faster it is than in the small airways is highly patient-specific. This is important because while it is commonly acknowledged that differences between participants in central ROI vs peripheral ROI deposition confound whole lung MCC measurements [30, 34], this work indicates that the relative speed of clearance between the large and small airways is also a confounding factor.

Additionally, until now, little work has gone into understanding differences in MCC between different lobes. The second eigenlung separated the MCC rates between upper and lower lobes, which indicates that how different the speed of MCC is in these two regions differs greatly between individuals. Importantly, for some individuals, the difference in speed may be very large. Traditional ROIs, which lump upper and lower lobes together into the same ROI, would therefore be confounded in individuals with large differences in speed between the upper and lower lobes.

Furthermore, all of these independently varying clusters were identified within a group of healthy controls. Without understanding the sources of variability inherent within healthy individuals, we cannot determine whether there exist meaningful differences in disease groups.

Specifically for CF, although there is a physical basis for depressed MCC, measurements of whole lung MCC were not significantly different between CF participants and healthy controls [37]. This work highlights a potential cause for this discrepancies. When our model was applied to healthy controls and CF individuals, we found a significant decrease in MCC rates in the lower lobe cluster in CF relative to the healthy controls, but not in any other clusters. If only the whole lung was considered, there would be no significant difference in MCC [34].

Though the statistical power was small due to the low number of healthy controls studied, this difference seems to be present when considering the response of CF participants to HS. Particularly, when analyzing relative feature importance in estimating response to HS, those identified as important in the lower lobe cluster were very different from other clusters. In particular, features associated with mucus properties were found to be less important and those associated with osmotic pressure more important in the lower lobe cluster than in other clusters. Overall, this suggests that MCC impairment functions differently in the lower lobe than it does in the upper. These findings would not be possible to identify without assessing regional differences in MCC.

5.1.3 Statistical Model to Estimate HS Effectiveness for Individuals from Nasal Epithelial Cells

Statistical models using clinical, *in vitro*, and thin-film *in silico* parameters from the same CF participants from our study as inputs were developed that could recapitulate the change in organ-scale *in silico* parameters between IS and HS study days. Because there were more possible input variables, or features, than study participants, a subset of features were selected for each statistical model. These were selected using recursive feature elimination based on LASSO regression coefficients. The magnitude of a LASSO coefficient corresponds to the relative importance of that feature in estimating the response variable (in our case the MCC rate coefficients or non-clearable activity in a cluster). While we could not infer any causality from these coefficients, they were analyzed collectively as covariates.

While the regression models were only moderately accurate (adjusted R^2 values between

0.09 and 0.56), when a threshold value was used to define participants as "responders" or "non-responders", binary classifiers based on these regression models performed quite well (area under ROC curves between 0.73 and 1.0). Though additional studies, as outlined in the next section, must be done before this could truly be used as a screening tool, this serves as a useful proof-of-concept to identify those most likely to benefit from HS treatment with non-invasive measurements.

5.2 Future Work

5.2.1 Model Refinement

While the dynamic and statistical models developed in this work are adequate to capture MCC behavior in healthy controls and CF participants and demonstrate the feasibility of a screening tool for predicting patient-specific HS efficacy, there are a number of refinements that have the potential to improve the model accuracy.

5.2.1.1 Grid Selection

One area that could be further optimized is in the grid network used to define flow constraints. We selected a grid of 16 x 8 equal area rectangles in order to draw direct comparisons with prior studies. However, there may be better dimensions that could more accurately capture airway size classes. While dividing the images into a different shape grid would be trivial (was automated within the Python portion of the image analysis), defining physiologically meaningful flow constraints would be less so. To do this, the LAR and grids outside the whole lung area would need to be redefined in the new grid space and a new elevation map generated. From there, flow constraints could be automatically defined based on the new elevation map. The remainder of the model reduction steps would also need to be repeated for this new grid network. While sweeping through all possible grid sizes would require substantial time both computationally and manually, it may improve the accuracy sufficiently to make the undertaking worthwhile.

Additional alterations could include unequal area grids or a different geometry network. Unequal area grids could allow for finer description of dynamics in regions that are very heterogeneous while reducing model complexity in areas that are homogeneous. This might provide better parsimony than the model in this work, however would require substantial reworking of the model. Similarly, a different geometry network, such as equilateral triangles or rhombi, might better capture the geometry of the airways. Because the main bronchus is at about a 35° angle from the trachea [81], the large airway branching is not oriented in the same diretion as the grid network and its edge is not smoothly defined. A geometry that is oriented in the same direction as the large airways may better capture this edge and therefore be more accurate in separating large airway and small airway dynamics in these regions.

5.2.1.2 Separate Non-Clearable Activity Clusters

Because we calculated the non-clearable activity in each grid directly from the nuclear images, we did not focus on clustering grids based on the similarity of their non-clearable activity. Based on the error estimating the change from IS to HS day of non-clearable activity in the statistical models, however, this may be a worthwhile direction to pursue. Since the MCC rate coefficients are independent of the initial deposition in each grid, it makes sense that non-clearable activity – a quantity dependent on deposition – would vary differently in space from these rate coefficients. Thus, the same clustering methods that were applied to the MCC rate coefficients could be applied to non-clearable activity. This would create clusters of homogeneous non-clearable activity across individuals. While there is no guarantee these would be more accurately estimated from the available features, it is reasonble to think it would. More homogoenous regions of non-clearble activity would likely lead to less spurious variability in the response of each region to HS.

5.2.1.3 Integrate Additional Camera Angles

One of the primary limitations to this work is the inability to determine the depth of radiolabeled particles in the imaging plane. While generating a full 3D image using SPECT would take too long to capture fast large airway clearance dynamics, an approach somewhere between a single camera angle and rotating a full 180° would provide more information about particle deposition in 3D while still capturing fast clearing dynamics.

The primary interest would be to classify existing measures of non-clearable activity into activity that deposited in the alveoli or on non-motile cilia versus activity that is in the large airways and just has not had sufficient time to clear within the 80 min study window. Since truly non-clearable activity (i.e. because of where it deposited) would be stationary over the course of the study taking images at additional angles towards the end of the study would provide just as much information as if they were taken throughout the study. Thus, by only taking images at the end of the study, we could glean additional useful information without reducing the information about fast clearing MCC dynamics from the early time points.

One major challenge this would add, however, is that signal from the left lung would start to interfere with the signal in the right lung. One possible solution would be to extend the model to the left lung, using the right lung model as a baseline. A different LAR would need to be defined since there are only two lobes in the left lungs, but otherwise the same process could be followed. Because of the physiologically-based flow constraints and appropriate parameter bounds based on the right lung, we may be able to get away with the error induced by interference from the stomach. Alternatively, we could attempt to correct for activity in the left lung based on an estimated area of overlap.

5.2.2 Applications to Other Lung Diseases

Probably the most obvious extension of this work would be to apply the model developed to nuclear imaging from individuals with other airway diseases. There is evidence of defective MCC in numerous other groups, such as individuals with asthma [10, 80], COPD [11, 82], primary ciliary dyskinesia [9], and those infected with SARS-CoV-2 [12]. By applying this model to nuclear imaging studies from those populations, we could gain better understanding of how MCC is impaired, as well as provide a tool to measure MCC response to therapies.

Because asthma and COPD are much more prevalent diseases, there is the potential for much larger studies this model could be applied to and thus greater statistical power to detect regional changes, as well as subtypes of disease patterns. For COPD in particular, recent work has identified two stratified subtypes of disease progression: one where small airway dysfunction preceded large airway abnormalities (70.4% of participants), and one with the reverse order (29.6% of participants) [83]. As such, using regional assessment of MCC, such as our model, will be of greater importance in studying defects and potential treatments since it could distinguish these two subtypes.

In addition to studying MCC rate coefficient changes in the context of disease, there is also a need to study more healthy controls to better define nominal MCC behavior. With a more precise distribution of healthy MCC rate coefficients and non-clearable activity, we would have more statistical power to determine if individuals lie outside a healthy range. This would be useful even for rare diseases, like CF or PCD, where subtle changes may be important but hard to detect in a small sample size.

5.2.3 Tool for Efficacy Screening of Osmotic Therapies

There were a number of factors that limited our ability to develop an effective screening tool to determine patient-specific response to HS from just a nasal swab, the primary of which was a small sample size. In spite of this, our results showed promise for a future tool developed on a larger data set. Should such a study take place, there are a number of items that would need to be considered.

Firstly, before studying this in CF participants, the dynamic model itself should be validated on additional healthy controls, ideally across multiple centers. Although this may seem burdensome, it may be possible, depending on similarity of study design to perform this analysis on healthy controls from previous studies, thus limiting the additional work necessary to validate this model. Since the screening tool would ultimately not include nuclear imaging, this is an important step to ensure model parameters derived from the images are reproducible.

Next, the CF participants in the study should be separated into two groups: one that will be used to refine the statistical models (in-sample) and one that will be withheld to use as a validation set (out-of-sample). Within the in-sample group, we can use the statistical models from this work to guide which variables are most important to measure to provide strong predictions.

While some of the important variables, such as age, BMI, and FEV$_1$, are readily available from clinical records or are almost always included in study outcomes, some of the features identified are less common. In particular, fluorescence recovery after photobleaching (FRAP) was found to be the most important feature for estimating the response of non-clearable activity to HS, in addition to being an independent predictor of whole lung MCC [34]. Though it is less commonly measured, our work indicates it may provide meaningful predictions of HS response and should be included. Some other features that are not always studied, but were found to be important across multiple clusters were paracellular liquid absorption (Cell ABS) and the lung clearance index (LCI). While many of the *in silico* parameters from Serrano Castillo et al. [23] were important in the regression models, particularly for non-clearable activity, removing them only had a minor effect on the binary classifier accuracy. Due to the specialized knowledge required to obtain these parameters, these would likely not be worth going after.

The methods used in this work (namely recursive feature elimination using LASSO regression with cross-validation) could be used to refine model coefficients and features included for each statistical model. Additionally, specific threshold values should be determined in conjunction with clinicians to define an adequate response to HS to merit including it in treatment. These can then be implemented within the binary classifiers to provide a single classification per cluster of "response" or "no response". Finally, the resulting linear models and binary classifiers should be applied to the validation group to assess the predictive ability of the models. This is a necessary step to provide confidence to the predictions produced from these models.

While this may not extend directly to other osmotic therapies, such as mannitol, since the predictive models are not mechanistic, the collection of important features may still be able to guide studies on future osmotic therapies as to which factors will likely play a role in patient-specific response.

Appendix A Nuclear Image Processing

A.1 ImageJ Processing

A.1.1 Requirements

In order to analyze DICOM files from gamma cameras, you'll need the following:

- ImageJ (1.52v)
- Base whole lung ROI
- `Stack2TextImages.txt` macro
- Transmission image
- Experiment images
- Background images

A.1.2 Define ROI for Subject

A.1.2.1 Transmission Scan

1. Open transmission scan with ImageJ
2. Open Brightness/Contrast window (`Ctrl + Shift + C` or Image > Adjust > Brightness/Contrast and adjust sliders until right lung is distinguishable from the background
3. Open the ROI Manager (Analyze > Tools > ROI Manager), then select More > Open...
4. Select the baseline whole lung ROI you plan to use. It should appear in the ROI Manager and overlaying the transmission scan.
5. Drag the ROI to best line up with the right lung. Use the Scale interface under Edit > Selection to match the outline up with the lung. Remember to click Update on the ROI Manager whenever you make a change to avoid losing your work.
6. When you're satisfied with the ROI, save it for later use.
7. Crop the image to your selection (`Ctrl+Shift+X` or Image > Crop) and save your file as a Text Image (File > Save As > Text Image) named, `TransScan.txt`

A.1.3 Align the ROI to Tc-99m Activity Window

The image stacks are broken down in the following way:

- (1-40): Anterior Tc-99m energy window
- (41-80): Posterior Tc-99m energy window
- (81-120): Anterior In-111 lower energy window
- (121-160): Posterior In-111 lower energy window
- (161-200): Anterior In-111 higher energy window
- (201-240): Posterior In-111 higher energy window

We will align the ROI using the Tc-99m energy windows since the activity in those ranges are higher. We will use both the Posterior and mirrored Anterior images to align the ROI.

1. Open the experiment DICOM (file name should be something like `FLOWANT_Tc99m001_DS ().dcm`).
2. Select Image > Lookup Tables > 16 Colors then adjust using the Brightness/Contrast window (`Ctrl+Shift+C`) until the activity in the lungs is distinguishable from the background noise.
3. Scroll to the 41st image in the stack (t=2 min, Posterior Tc-99m) and select the ROI from the ROI Manager.
4. Drag the ROI to best capture all the activity in the right lung in images 41-80. Click the Update button in the ROI Manager to save these changes.
5. Click outside of the image to unselect the ROI. Then select Image > Transform > Flip Horizontally.
6. Click on the ROI in the ROI Manager. Then use the keyboard arrows to adjust the position to make sure it captures all the right lung activity in images 1-40. If major changes are made, repeat steps 6 7 to make sure the posterior images still line up. Once you are satisfied with the ROI position, save the ROI.
7. Select Edit > Selection > Create Mask. Click on the new window that appears, select the ROI from the ROI Manager, then crop the image to this selection (Ctrl+Shift+X). Save this as a Text Image called WLMask.txt.
8. Once this is complete, close all image windows without saving.

A.1.4 Crop Anterior and Posterior Images to Right Lung

1. Install Stack2TextImages Macro:
   - Select Plugins > Macros > Install...
   - Navigate to and select the `Stack2TextImages.txt` file
2. Re-open the experiment DICOM and select the ROI in the ROI Manager.
3. Crop the stack of images to this ROI (`Ctrl+Shift+X`).
4. Run the Stack2TextImages macro (Plugins > Macros > Stack2TextImages). When prompted, create a directory within your folder of scans called, TcScans, and select it as the destination directory. Leave the default value of `Tc_` when prompted for a file prefix.
5. Close the stack of images without saving.
6. Re-open the experiment DICOM, then flip the images horizontally (Image > Transform > Flip Horizontally).
7. Select the ROI in the ROI Manager and crop to the ROI.
8. Run the Stack2TextImages macro again. This time, when prompted, create a new directory called, AntScans, and select it as the destination directory. Again, leave the default value for the file prefix.
9. Close the stack of images without saving.

A.1.5 Crop Background Scans

Images of the background activity in each energy window are in the same order as the experiment DICOM, but there is a single image for each camera view & energy window.

1. Open the background images (file like `Head1_Tc99m001_DS (#).dcm`).
2. Select the ROI in the ROI Manager and crop the images to it.
3. Run the Stack2TextImages macro. When prompted, create and select a new directory called, BgScans. Change the file prefix to `Bg_` when prompted.
4. Close the stack without saving.
5. Re-open the background images and flip them horizontally (Image > Transform > Flip Horizontally).

6. Select the ROI in the ROI Manager and crop the images to it.
7. Run the Stack2TextImages macro. When prompted, create and select a new directory called, AntBgScans. Again, change the file prefix to `Bg_` when prompted.
8. Close the stack without saving.

A.1.6 Conclusions

You should now have a directory with the following items:

- AntBgScans directory
- AntScans directory
- BgScans directory
- TcScans directory
- `WLMask.txt` text file

If so, you have all the files necessary to run the analysis code in Python, as described in the next section.

A.1.7 Stack2TextImages Macro

```
dir=getDirectory("Choose Destination Directory");
prefix=getString("Filename Prefix", "Tc_");
for(i=1; i<=nSlices; i++){
    setSlice(i);
    saveAs("Text Image", dir+prefix+i+".txt");
}
```

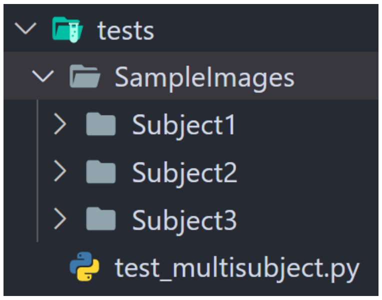
A.2 Python Processing

A.2.1 Requirements

You will need Python 3 and an installed version of the `psanalysis` package (`https://github.com/monshap/psanalysis`). The package itself requires NumPy and SciPy, however, if you install the package using pip, it should automatically check for these requirements.

A.2.2 Multi-Subject Tutorial

For the sake of ease, we will walk through one of the test files (https://github.com/monshap/psanalysis/tree/master/tests), which demonstrates how to process a directory containing subdirectories for three study days. I will point out places where users may wish to customize the analysis. The file structure for this tutorial is shown in Figure 33 and can also be found under the tests subdirectory of this package. We will be walking through the file test_multisubject.py.



```
tests
  SampleImages
    Subject1
    Subject2
    Subject3
  test_multisubject.py
```

Figure 33: File structure for tutorial

A.2.2.1 Import Relevant Packages

The commented out lines were specifically required due to the script's location in the tests subdirectory. You do not need to include them if the package is installed.

```
import os
# import sys

import numpy as np
from scipy.stats.mstats import gmean

# sys.path.append(os.path.abspath("../"))
import psanalysis.analyze_grids as ag
```

A.2.2.2 Configure

The following lines will depend on the specific structure and naming scheme of your files. In this example, Subject1, Subject2, and Subject3 each contain the files extracted from a single study day. If your files are setup differently, you will need to modify the `wd` and `dir_list` variables accordingly. `dir_list` should be a list of strings matching the names of each study subdirectory. tpts should be an array or list of time points images were taken at and `ny` and `nx` should be the number of rows and columns you want in your grid, respectively.

```
wd = os.path.dirname(os.path.realpath(__file__))
tpts = np.arange(0, 80, 2)
dir_list = [f"Subject{i}" for i in range(1, 4)]
(ny, nx) = (16, 8)
nt = len(tpts)
nsub = len(dir_list)
```

A.2.2.3 Initialize a Set of Empty NumPy Arrays

In order, the arrays correspond to the following:

- areas in units of pixels^2 of all grids for a subject (`all_areas`)
- anterior Tc-SC intensity in each grid, time point, and subject (`all_aTc`)
- posterior Tc-SC intensity in each grid, time point, and subject (`all_pTc`)
- anterior In-DTPA intensity in each grid, time point, and subject (`all_aIn`)
- posterior In-DTPA intensity in each grid, time point, and subject (`all_pIn`)

```
all_areas = np.zeros((nsub,))
all_aTc = np.zeros((ny, nx, nt, nsub))
all_pTc = np.zeros((ny, nx, nt, nsub))
all_aIn = np.zeros((ny, nx, nt, nsub))
all_pIn = np.zeros((ny, nx, nt, nsub))
```

A.2.2.4 Loop Through Each Subject

This will fill in the arrays with processed values for each subject. If your file structure is setup differently, make sure the first argument to `ag.PlanarStudy` is the location of a study day directory.

```python
for i, sub in enumerate(dir_list):
    test_dir = ag.PlanarStudy(os.path.join(wd, "SampleImages", sub), tpts)
    test_dir.preprocess_scans(debug=False)
    (a, grid_aTc, grid_aIn, grid_pTc, grid_pIn) = test_dir.gridify(ny, nx)
    all_areas[i] = a
    all_aTc[..., i] = grid_aTc
    all_pTc[..., i] = grid_pTc
    all_aIn[..., i] = grid_aIn
    all_pIn[..., i] = grid_pIn
```

A.2.2.5 Calculate the Geometric Mean

Returns two arrays of shape (ny, nx, nt, nsub) for the Tc-SC and In-DTPA geometric means.

```python
Tc_stack = np.clip(np.stack((all_aTc, all_pTc)), 0, None)
In_stack = np.clip(np.stack((all_aIn, all_pIn)), 0, None)
gm_Tc = gmean(Tc_stack, axis=0)
gm_In = gmean(In_stack, axis=0)
```

Appendix B Code for Models

B.1 Grid Model Fitting

```python
import os
import sys
from math import log

import numpy as np
import pyomo.dae as pdae
import pyomo.environ as pe
from plepy import PLEpy
from scipy.io import loadmat


def sflag(results):
    # Return flag corresponding to solver status
    # 0 = converged to optimal solution
    # 1 = problem may be infeasible
    # 2 = reached maximum number of iterations
    # 3 = sometimes there are other weird errors
    from pyomo.opt import SolverStatus, TerminationCondition

    stat = results.solver.status
    tcond = results.solver.termination_condition
    if ((stat == SolverStatus.ok) and
        (tcond == TerminationCondition.optimal)):
        flag = 0
    elif (tcond == TerminationCondition.infeasible):
        flag = 1
    elif (tcond == TerminationCondition.maxIterations):
        flag = 2
    else:
        flag = 3
    return flag


def get_bool_maps(el, diags=True, equal_flow=True, eq=None):
    """Function for defining flow constraints

    Parameters:
    el (ndarray): Elevation map
```

```
Keywords:
diags (bool): Whether or not to allow flow between diagonal grids,
    True by default
equal_flow (bool): If true, magnitude of flow is equal in all possible
    directions. Else, flow is proportional to difference in elevation.
    True by default
eq (bool type ndarray): ND array defining where to allow flow at equal
    elevation. 1=can flow to equal elevations, 0=can only flow downhill

Returns:
List of boolean arrays corresponding to whether the grids can flow in each
cardinal direction. Order of directions: N, NE, E, SE, S, SW, W, NW
"""

def get_dir_bool(yslice, xslice):
    # flow based on elevation
    el_part = el_bool[1:-1, 1:-1] - el_bool[yslice, xslice]
    # whether to allow flow at equal elevation
    eq_part = np.logical_and(eq_bool[1:-1, 1:-1], eq_bool[yslice, xslice])
    eq_part = eq_part.astype("int")
    combined = (el_part + eq_part)/(ndir/2)
    return combined

# set map properties
ny, nx = np.shape(el)
if not isinstance(eq, np.ndarray):
    eq = np.ones_like(el)
eq_bool = np.zeros((ny+2, nx+2))
eq_bool[1:-1, 1:-1] = eq
if diags:
    ndir = 8
else:
    ndir = 4

# preprocess elevation map
out_lung = el == 0
barrier = int(np.max(el)+1)
el_bool = (barrier+1)*np.ones(((ny+2), (nx+2)))
el[out_lung] = barrier
exit_blocks = np.vstack(
    (
        np.zeros((1, nx+2), dtype=bool),
        np.hstack((el == 1, np.zeros((ny, 2), dtype=bool))),
        np.zeros((1, nx+2), dtype=bool)
    )
)
```

```
    el_bool[exit_blocks] = 0
    el_bool[1:-1, 1:-1] = el

    # subtract elevation from specified direction
    # add 1 if allowing flow between grids of equal elevation
    # denominator is used when flow is proportional rather than equal
    bool_n = get_dir_bool(slice(0, -2), slice(1, -1))
    bool_ne = get_dir_bool(slice(0, -2), slice(2, None))
    bool_e = get_dir_bool(slice(1, -1), slice(2, None))
    bool_se = get_dir_bool(slice(2, None), slice(2, None))
    bool_s = get_dir_bool(slice(2, None), slice(1, -1))
    bool_sw = get_dir_bool(slice(2, None), slice(0, -2))
    bool_w = get_dir_bool(slice(1, -1), slice(0, -2))
    bool_nw = get_dir_bool(slice(0, -2), slice(0, -2))

    # remove negative values
    bool_n[bool_n <= 0] = 0
    bool_ne[bool_ne <= 0] = 0
    bool_e[bool_e <= 0] = 0
    bool_se[bool_se <= 0] = 0
    bool_s[bool_s <= 0] = 0
    bool_sw[bool_sw <= 0] = 0
    bool_w[bool_w <= 0] = 0
    bool_nw[bool_nw <= 0] = 0
    if equal_flow:
        bool_n = bool_n > 0
        bool_ne = bool_ne > 0
        bool_e = bool_e > 0
        bool_se = bool_se > 0
        bool_s = bool_s > 0
        bool_sw = bool_sw > 0
        bool_w = bool_w > 0
        bool_nw = bool_nw > 0
    if not diags:
        bool_ne = np.zeros((ny, nx))
        bool_se = np.zeros((ny, nx))
        bool_nw = np.zeros((ny, nx))
        bool_sw = np.zeros((ny, nx))
    # make numerical
    bool_n = bool_n.astype("float")
    bool_ne = bool_ne.astype("float")
    bool_nw = bool_nw.astype("float")
    bool_s = bool_s.astype("float")
    bool_se = bool_se.astype("float")
    bool_sw = bool_sw.astype("float")
    bool_e = bool_e.astype("float")
    bool_w = bool_w.astype("float")
    return [bool_n, bool_ne, bool_e, bool_se, bool_s, bool_sw, bool_w, bool_nw]


class GridModel(pe.ConcreteModel):
    # Class defining PYOMO model for specified grid model

    ts = list(range(0, 80, 2))
    ts8 = list(range(8, 80, 2))

    def __init__(self, el, clust, fixlow, cdata, eq, wmat=None):
        """Define Concrete PYOMO Model for model & data specified

        Parameters:
        el (ndarray): Elevation map
        clust (ndarray): ND array defining which cluster each grid belongs to.
            To fit a parameter for each grid, specify unique cluster values for
            each grid in lung
        fixlow (ndarray): ND boolean array defining grids to fix the clearance
            rate parameter for
        cdata (ndarray): Concentration data of shape (ny, nx, nt)
        eq (ndarray): ND boolean array defining where to allow flow at equal
            elevation. 1=can flow to equal elevations, 0=can only flow downhill

        Keywords:
        wmat (ndarray): Weight matrix for each grid in overall objective. If
            None, equal weights will be used. None by default.

        Returns:
        Instance of class GridModel

        Methods:
        solve_grids: Use to solve for ideal solution for each grid (see paper
            for details). Must be done *prior* to fitting overall objective.
        solve_overall_sse: Minimize overall objective function
        solve_cluster_sse: Minimize objective within an individual cluster,
            neglecting error in other clusters
        """
        pe.ConcreteModel.__init__(self)
        # generic attributes
        self.ny, self.nx = np.shape(el)
        self.ngrid = self.ny*self.nx
        self.clust = clust
        self.nK = np.max(self.clust)
        self.fixlow = fixlow
        self.nfix = np.sum(self.fixlow > 0)
        self.out_lung = el == 0
        self.bool_map = get_bool_maps(el, eq=eq)
        self.sum_leave = np.clip(np.sum(self.bool_map, axis=0), 1, 8)
        self.bool_n = self.bool_map[0]
        self.bool_ne = self.bool_map[1]
        self.bool_e = self.bool_map[2]
        self.bool_se = self.bool_map[3]
        self.bool_s = self.bool_map[4]
        self.bool_sw = self.bool_map[5]
        self.bool_w = self.bool_map[6]
        self.bool_nw = self.bool_map[7]

        # PYOMO attributes
        # indices
        self.t = pdae.ContinuousSet(bounds=(8, 80), initialize=self.ts8)
        self.i = pe.RangeSet(0, self.ny-1)
        self.j = pe.RangeSet(0, self.nx-1)
        self.K = pe.RangeSet(0, self.nK)
        # fitted parameters
        self.k = pe.Var(self.K, initialize=1.0, bounds=(1e-3, 1e3))
        self.k[self.nK] = 1e-2
        self.k[self.nK].fix()
        fixclust = self.clust[self.fixlow > 0]
        fixmed = self.fixlow[self.fixlow > 0]
        for i in range(self.nfix):
            self.k[fixclust[i]] = fixmed[i]
            self.k[fixclust[i]].fix()
        # states
        self.C = pe.Var(self.t, self.i, self.j, within=pe.NonNegativeReals)
        # derivatives
        self.dCdt = pdae.DerivativeVar(self.C, wrt=self.t)

        # data attributes
        if wmat is None:
            self.wmat = np.ones((self.ny, self.nx))
        else:
            emsg = f"Weight matrix dimensions should be ({self.ny},{self.nx})."
            assert np.shape(wmat) == (self.ny, self.nx), emsg
            self.wmat = wmat
        self.cdata = cdata
        self.C8 = np.clip(np.mean(self.cdata[..., 3:6], axis=2), 1e-3, 100)
        for t in self.t:
            for i in self.i:
                for j in self.j:
                    self.C[t, i, j] = self.C8[i, j]
                    self.dCdt[t, i, j] = -1e-2*self.C8[i, j]

    def _init_cond(self, m, i, j):
        if self.out_lung[i, j]:
            return m.C[8, i, j] == 0.
        else:
            return m.C[8, i, j] == self.C8[i, j]

    def _odes(self, m, t, i, j):
        if self.out_lung[i, j]:
            return m.dCdt[t, i, j] == 0.
        else:
            K = self.clust[i, j]
            rhs = 0.
            rhs = rhs - m.k[K] * m.C[t, i, j]
            if i < self.ny-1:
                rhs = rhs + (self.bool_n[i+1, j] * m.k[self.clust[i+1, j]]
                             * m.C[t, i+1, j]) / self.sum_leave[i+1, j]
            if (i < self.ny-1) and (j > 0):
                rhs = rhs + (self.bool_ne[i+1, j-1] * m.k[self.clust[i+1, j-1]]
                             * m.C[t, i+1, j-1]) / self.sum_leave[i+1, j-1]
            if j > 0:
                rhs = rhs + (self.bool_e[i, j-1] * m.k[self.clust[i, j-1]]
                             * m.C[t, i, j-1]) / self.sum_leave[i, j-1]
            if (j > 0) and (i > 0):
                rhs = rhs + (self.bool_se[i-1, j-1] * m.k[self.clust[i-1, j-1]]
                             * m.C[t, i-1, j-1]) / self.sum_leave[i-1, j-1]
            if i > 0:
                rhs = rhs + (self.bool_s[i-1, j] * m.k[self.clust[i-1, j]]
                             * m.C[t, i-1, j]) / self.sum_leave[i-1, j]
            if (i > 0) and (j < self.nx-1):
                rhs = rhs + (self.bool_sw[i-1, j+1] * m.k[self.clust[i-1, j+1]]
                             * m.C[t, i-1, j+1]) / self.sum_leave[i-1, j+1]
            if j < self.nx-1:
                rhs = rhs + (self.bool_w[i, j+1] * m.k[self.clust[i, j+1]]
                             * m.C[t, i, j+1]) / self.sum_leave[i, j+1]
            if (j < self.nx-1) and (i < self.ny-1):
                rhs = rhs + (self.bool_nw[i+1, j+1] * m.k[self.clust[i+1, j+1]]
                             * m.C[t, i+1, j+1]) / self.sum_leave[i+1, j+1]
            return m.dCdt[t, i, j] == 0.1*rhs

    def _grid_obj(self, m):
        i, j = self.gid
        sse_ij = sum([self.wmat[i, j]*(m.C[self.ts[t], i, j]
                                       - self.cdata[i, j, t])**2
                      for t in range(4, 40)])
        return sse_ij

    def _overall_obj_sse(self, m):
        sse = 0.
        for cl in range(self.nK):
            gids = np.c_[np.where(self.clust == cl)]
            for gid in gids:
                i, j = gid
                sse_ij = sum([(m.C[self.ts[t], i, j] - self.cdata[i, j, t])**2
                              for t in range(4, 40)])
                sse += self.wmat[i,j] * (sse_ij - self.best_sse[i, j])
        return sse

    def _clust_obj(self, m):
        sse = 0.
        for gid in self.gids:
            i, j = gid
            sse_ij = sum([(m.C[self.ts[t], i, j] - self.cdata[i, j, t])**2
                          for t in range(4, 40)])
            sse += self.wmat[i,j] * (sse_ij - self.best_sse[i, j])
        return sse

    def solve_grids(self, solver=None, find_best=True, find_worst=False):
        # Method to find best fit when only considering each grid individually
        # (i.e. error in other grids can be humongous and we don't care)
        emsg = "Neither best nor worst fit selected."
        assert find_best or find_worst, emsg

        # activate constraints
        self.init_cond = pe.Constraint(self.i, self.j, rule=self._init_cond)
        self.odes = pe.Constraint(self.t, self.i, self.j, rule=self._odes)
        # define solver
        if solver is not None:
            self.solver = solver
        else:
            self.solver = pe.SolverFactory('ipopt')
            self.solver.options['linear_solver'] = 'ma97'
            self.solver.options['tol'] = 1e-6
            self.solver.options['max_iter'] = 600
            self.solver.options['acceptable_tol'] = 1e-4
        # numerical discretization
        tfd = pe.TransformationFactory("dae.finite_difference")
        tfd.apply_to(self, nfe=2*len(self.t), wrt=self.t, scheme="BACKWARD")

        # find best and/or worst SSE for each grid
        if find_best:
            self.best_sse = np.zeros((self.ny, self.nx))
        if find_worst:
            self.worst_sse = np.zeros((self.ny, self.nx))
        for cl in range(self.nK):
            self.gid = np.c_[np.where(self.clust == cl)][0]
            if find_best:
```

```
            # minimize objective
                self.grid_obj = pe.Objective(rule=self._grid_obj)
                results = self.solver.solve(self, keepfiles=False, tee=False,
                                            load_solutions=False)
                flag = sflag(results)
                if flag == 0:
                    self.solutions.load_from(results)
                    best_sse = pe.value(self.grid_obj)
                    self.best_sse[self.gid[0], self.gid[1]] = best_sse
                self.del_component(self.grid_obj)
            if find_worst:
                # maximize objective
                self.grid_obj = pe.Objective(rule=self._grid_obj,
                                            sense=pe.maximize)
                results = self.solver.solve(self, keepfiles=False, tee=False,
                                            load_solutions=False)
                flag = sflag(results)
                if flag == 0:
                    self.solutions.load_from(results)
                    worst_sse = pe.value(self.grid_obj)
                    self.worst_sse[self.gid[0], self.gid[1]] = worst_sse
                self.del_component(self.grid_obj)

    def solve_overall_sse(self, presolved=False):
        # Minimize overall objective function (sum of SSE between model & data
        # minus the best possible error for each grid) by varying all grid
        # clearance parameters. See paper for mathematical details.
        if presolved is False:
            # activate constraints
            self.init_cond = pe.Constraint(self.i, self.j,
                                           rule=self._init_cond)
            self.odes = pe.Constraint(self.t, self.i, self.j, rule=self._odes)
            # define solver
            if not hasattr(self, "solver"):
                self.solver = pe.SolverFactory('ipopt')
                self.solver.options['linear_solver'] = 'ma97'
                self.solver.options['tol'] = 1e-6
                self.solver.options['max_iter'] = 600
                self.solver.options['acceptable_tol'] = 1e-4
            # numerical discretization
            tfd = pe.TransformationFactory("dae.finite_difference")
            tfd.apply_to(self, nfe=2*len(self.t), wrt=self.t,
                         scheme="BACKWARD")
        self.overall_obj = pe.Objective(rule=self._overall_obj_sse)
        results = self.solver.solve(self, keepfiles=False, tee=True,
                                    load_solutions=False)
        flag = sflag(results)
        if flag == 0:
            self.solutions.load_from(results)

    def solve_cluster_sse(self, presolved=False):
        # Minimize error for a specific cluster, without regard to error in
        # other clusters. Similar to solve_grids, but for whole cluster.
        if presolved is False:
            # activate constraints
            self.init_cond = pe.Constraint(self.i, self.j,
                                           rule=self._init_cond)
            self.odes = pe.Constraint(self.t, self.i, self.j, rule=self._odes)
            # define solver
            if not hasattr(self, "solver"):
                self.solver = pe.SolverFactory('ipopt')
                self.solver.options['linear_solver'] = 'ma97'
                self.solver.options['tol'] = 1e-6
                self.solver.options['max_iter'] = 600
                self.solver.options['acceptable_tol'] = 1e-4
            # numerical discretization
            tfd = pe.TransformationFactory("dae.finite_difference")
            tfd.apply_to(self, nfe=2*len(self.t), wrt=self.t,
                         scheme="BACKWARD")
        self.obj = pe.Objective(rule=self._clust_obj)
        results = self.solver.solve(self, keepfiles=False, tee=True,
                                    load_solutions=False)
        flag = sflag(results)
        if flag == 0:
            self.solutions.load_from(results)

    def get_grid_sse(self):
        # Calculate error in each grid from optimized solution
        self.solved_error = np.zeros((self.ny, self.nx))
        for i in range(ny):
            for j in range(nx):
                sse_ij = sum([(pe.value(self.C[self.ts[t], i, j])
                               - self.cdata[i, j, t])**2
                              for t in range(4, 40)])
                self.solved_error[i, j] = sse_ij
        return self.solved_error


# define subject labels
hc_labels = [f"UHC{i}" for i in [*range(1, 7), *range(9, 14), 16]]
cr_labels = [f"UP{i}" for i in [*range(1, 10), *range(11, 17)]]
cfis_labels = [f"UCF{i} IS" for i in [1, 3, 4, *range(6, 28)]]
cfhs_labels = [f"UCF{i} HS" for i in [1, 3, 4, *range(6, 28)]]
labels = hc_labels + cr_labels + cfis_labels + cfhs_labels
```

```
nhc = len(hc_labels)
ncr = len(cr_labels)
ncf = len(cfis_labels)
nsub = nhc + ncr + 2*ncf

# conc format: (row, col, t, subject)
raw_gmTc = np.load("GridData_ALL.npz")["gm_Tc"]
gmTc0 = np.sum(raw_gmTc[:,:,0,:], axis=(0,1))
gmTcNorm = np.divide(raw_gmTc, gmTc0, out=np.zeros_like(raw_gmTc),
                     where=gmTc0!=0)*100
conc = gmTcNorm # total concentration of activity
flr0 = np.percentile(conc, 25, axis=2)  # non-clearable activity
no_flr = conc - flr0[:, :, np.newaxis, :]   # clearable activity
ny, nx, nt, nsub2 = np.shape(conc)
assert nsub2 == nsub, "Number of subjects don't match!"
ngrid = nx*ny
ts = list(range(0, 80, 2))

# define elevation map
el = np.load("Emap_16x8_diags_HRCT.npz")["arr_0"]
eq_thresh = 4
eq = el < eq_thresh
out_lung = el == 0
(bool_n, bool_ne, bool_e, bool_se,
 bool_s, bool_sw, bool_w, bool_nw) = get_bool_maps(el, eq=eq)
sum_leave = np.sum([bool_n, bool_ne, bool_e, bool_se, bool_s, bool_sw,
                    bool_w, bool_nw], axis=0)

# load cluster map
# server version:
n_clust = int(sys.stdin.read())
# local version:
# n_clust = 5

# for reduced model:
# prefix = "aggcl"
# link = "ward_pca"
# clmap_name = "{:s}_{:g}_link_{:s}".format(prefix, n_clust, link)

# for full-scale model:
clmap_name = "full_model"

clust = np.load(f"{clmap_name}.npz")["arr_0"]
clust = clust.astype("int32")
nout = np.sum(out_lung)
nin = ngrid - nout
# we're not currently fixing clearance parameters for any grids
```

```
fixlow = np.zeros((ny, nx))
nfix = np.sum(fixlow > 0)
nK = np.max(clust)
nclust = nK - nfix

# for each subject, for each grid, find minimum SSE for that grid, letting all
# parameters vary - sets ideal vector, z*, for Chebychev metric
presolve_grids = False

# save outputs to file (False or name of output file)
save_best_sse = False
save_worst_sse = False
save_params = f"{clmap_name}_params.npz"
save_grid_sse = f"{clmap_name}_grid_err.npz"

# load from previous run (False or name of input file)
load_params = False
load_best_sse = "best_full_hc_sse.npz"
load_worst_sse = False
load_grid_sse = False

all_ks = np.zeros((nK, nsub))
all_best_sse = np.zeros((ny, nx, nsub))
all_worst_sse = np.zeros((ny, nx, nsub))
all_grid_sse = np.zeros((ny, nx, nsub))

if isinstance(load_params, str):
    all_ks = np.load(load_params)["all_ks"]
if isinstance(load_best_sse, str):
    all_best_sse = np.load(load_best_sse)["all_best_sse"]
if isinstance(load_worst_sse, str):
    all_worst_sse = np.load(load_worst_sse)["all_worst_sse"]
if isinstance(load_grid_sse, str):
    all_grid_sse = np.load(load_grid_sse)["all_grid_sse"]

for pat_num, pat_str in enumerate(hc_labels):
    eq = el < eq_thresh
    pat_no_flr = no_flr[..., pat_num]
    pat_conc = conc[..., pat_num]
    print(pat_str)
    pat_gm = GridModel(el, clust, fixlow, pat_no_flr, eq)
    pat_gm.best_sse = all_best_sse[..., pat_num]
    pat_gm.worst_sse = all_worst_sse[..., pat_num]
    if presolve_grids:
        pat_gm.solve_grids()
    if isinstance(save_best_sse, str):
        all_best_sse[..., pat_num] = pat_gm.best_sse
        np.savez(save_best_sse, all_best_sse=all_best_sse)
    if isinstance(save_worst_sse, str):
        all_worst_sse[..., pat_num] = pat_gm.worst_sse
        np.savez(save_worst_sse, all_worst_sse=all_worst_sse)

    # solve overall objective fn - change to match run
    if isinstance(load_params, str):
        for K in range(nclust):
            pat_gm.k[K] = all_ks[K, pat_num]
    pat_gm.solve_overall_sse(presolved=presolve_grids)
    grid_err = pat_gm.get_grid_sse()
    if isinstance(save_grid_sse, str):
        all_grid_sse[..., pat_num] = grid_err
        np.savez(save_grid_sse, all_grid_sse=all_grid_sse)
    ks = np.array([pe.value(pat_gm.k[K]) for K in range(pat_gm.nK)])
    if isinstance(save_params, str):
        all_ks[:, pat_num] = ks
        np.savez(save_params, all_ks=all_ks)
```

B.2 Agglomerative Clustering

```
import numpy as np
import os
from sklearn.cluster import AgglomerativeClustering
from sklearn.decomposition import PCA

from grid_helpers import get_bool_maps


def calc_linkage_matrix(model):
    counts = np.zeros(model.children_.shape[0])
    n_samples = len(model.labels_)
    for i, merge in enumerate(model.children_):
        current_count = 0
        for child_idx in merge:
            if child_idx < n_samples:
                current_count += 1  # leaf node
            else:
                current_count += counts[child_idx - n_samples]
        counts[i] = current_count
    linkage_matrix = np.column_stack(
        [model.children_, model.distances_, counts]
    ).astype(float)
    return linkage_matrix

# grid properties
(ny, nx) = (16, 8)
ngrid = ny*nx
diags = True

# define elevation map
el = np.load("Emap_16x8_diags_HRCT.npz")["arr_0"]
eq = el < 4
out_lung = el == 0
nout = np.sum(out_lung)
nin = ngrid - nout
bn, bne, be, bse, bs, bsw, bw, bnw = get_bool_maps(el, eq=eq)
sum_leave = np.clip(np.sum([bn, bne, be, bse, bs, bsw, bw, bnw], axis=0), 1, 8)

# cluster map
clust = np.zeros((ngrid,))
clust[out_lung.flatten()] = nin
clust[~out_lung.flatten()] = np.arange(0, nin)
clust = np.reshape(clust, (ny, nx))
clust = clust.astype("int32")

# adjacency matrix
adj = np.zeros((nin, nin))
l = [(a, b) for (a, b) in zip(clust[:, 1:].flatten(), clust[:, :-1].flatten())
            if (a < nin) & (b < nin)]
r = [(a, b) for (a, b) in zip(clust[:, :-1].flatten(), clust[:, 1:].flatten())
            if (a < nin) & (b < nin)]
u = [(a, b) for (a, b) in zip(clust[1:, :].flatten(), clust[:-1,:].flatten())
            if (a < nin) & (b < nin)]
d = [(a, b) for (a, b) in zip(clust[:-1, :].flatten(), clust[1:,:].flatten())
            if (a < nin) & (b < nin)]
adj[tuple(zip(*l))] = 1
adj[tuple(zip(*r))] = 1
adj[tuple(zip(*u))] = 1
adj[tuple(zip(*d))] = 1
if diags:
    ul = [(a, b) for (a, b) in
                 zip(clust[1:, 1:].flatten(), clust[:-1, :-1].flatten())
                 if (a < nin) & (b < nin)]
    dl = [(a, b) for (a, b) in
                 zip(clust[:-1, 1:].flatten(), clust[1:, :-1].flatten())
                 if (a < nin) & (b < nin)]
    ur = [(a, b) for (a, b) in
                 zip(clust[1:, :-1].flatten(), clust[:-1, 1:].flatten())
                 if (a < nin) & (b < nin)]
    dr = [(a, b) for (a, b) in
                 zip(clust[:-1, :-1].flatten(), clust[1:, 1:].flatten())
                 if (a < nin) & (b < nin)]
    adj[tuple(zip(*ul))] = 1
    adj[tuple(zip(*dr))] = 1
    adj[tuple(zip(*ur))] = 1
    adj[tuple(zip(*dl))] = 1
adj = adj + np.eye(nin)

# healthy labels
HClabels = ['UHC'+str(i) for i in [*range(1,7), *range(9,14), 16]]
nHC = len(HClabels)

# load parameters
ks = np.load("full_model_params.npz")["all_ks"]
hc_keff = ks[:, :nHC]

# normalize data
hc_keff = np.delete(hc_keff, 8, axis=1) # remove UHC11 for low deposition
hc_logkeff = np.log(hc_keff)
quarts = np.percentile(hc_logkeff, [25, 50, 75])
iqr = quarts[2] - quarts[0]
upper_lim = quarts[1] + 1.5*iqr
lower_lim = quarts[1] - 1.5*iqr
hc_clipped = np.clip(hc_logkeff, lower_lim, upper_lim)
hc_lognorm = (hc_clipped - np.mean(hc_clipped))/np.std(hc_clipped)

# principal components
pca = PCA()
pca.fit(hc_lognorm)
pc_weights = pca.components_
hc_pcs = pca.transform(hc_lognorm)

# initialize agglomerative clustering model
n_clust = 5
n_pcs = 2
link = "ward"
for n_clust in range(2, 10):
    model = AgglomerativeClustering(n_clusters=n_clust, connectivity=adj,
                                    linkage=link, compute_distances=True)

    # fit model to parameter data
    X = hc_pcs[:, :n_pcs]
    model.fit(X)
    link_mat = calc_linkage_matrix(model)

    # format as cluster map
    ext_labels = np.append(model.labels_, model.n_clusters_)
    clust_map = ext_labels[clust]
    # np.savez("aggcl_{:g}_link_{:s}_pca.npz".format(n_clust, link), clust_map)
```

B.3 Statistical Models

```
import os
from itertools import combinations

import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import sklearn.linear_model as lm
from autocolor import color_yaxis
from scipy.stats import beta
from sklearn.feature_selection import RFE, RFECV, f_regression
from sklearn.impute import KNNImputer, SimpleImputer
from sklearn.metrics import (auc, precision_recall_curve, roc_auc_score,
                             roc_curve)
from sklearn.model_selection import LeaveOneOut, RepeatedKFold, cross_validate
from sklearn.preprocessing import StandardScaler

# Load data frame
is2hs = pd.read_json("is2hs_response.json")
nK = 5

# Set figure directory
home = os.path.expanduser("~")
fig_dir = os.path.join(home, "OneDrive - University of Pittsburgh", "Research",
                       "Group Presentations", "Figures", "Stat Model Paper")
if not os.path.exists(fig_dir):
    os.makedirs(fig_dir)

response_vars = [*[f"delta_k{i}" for i in range(nK)],
                 *[f"delta_flr{i}" for i in range(nK)],
                 *[f"delta_C{i}" for i in range(nK)]]
```

```
features = is2hs[[col for col in is2hs.columns if col not in response_vars]]
features = features.drop(["Subject", "Race", "Corrector", "delta_MCC Adj.",
                          "Gender", "Height (cm)", "Weight (kg)", "FRC"],
                         axis=1)
clin_var = ["Age", "BMI", "Sweat Cl", "LCI", "FEV1", "FEF25-75", "delta_FEV1",
            "delta_FEF25-75"]
invitro_var = ["FRAP (mean)", "Norm. Vap 2-12h", "Cell ABS (mean)",
               "delta_Norm. Vap 2-12h", "delta_Cell ABS (mean)"]
insilico_var = ["Pwca", "Pwcb", "Pwab", "Penac", "Pacc", "Pbk", "Pbcl",
                "Pbkca", "Jnakp", "Jnkcc", "Ppna", "Ppcl", "Ppk", "Ppdt"]
high_corr = ["Ppna", "Pwca", "delta_Norm. Vap 2-12h", "Pbk", "Penac",
             "FEF25-75"]
rem_high_corr = lambda x: x not in high_corr
clin_var = list(filter(rem_high_corr, clin_var))
invitro_var = list(filter(rem_high_corr, invitro_var))
insilico_var = list(filter(rem_high_corr, insilico_var))

# Feature sets
v1 = features[clin_var]
v2 = features[clin_var + invitro_var]
v3 = features[clin_var + invitro_var + insilico_var]

# Imupte & scale data
fprefix = "AllVars_"
ver = v3
knn_imputer = KNNImputer(n_neighbors=3, weights="distance")
X = knn_imputer.fit_transform(ver)
xscaler = StandardScaler()
Xscaled = xscaler.fit_transform(X)

nfeat_total = len(ver.columns)
nfold = 5
nalpha = 100

final_nf = {}
final_lm = {}
final_ypred = {}
final_adjr = {}
final_ci = {}
final_yp_ci = {}
final_feat_in = {}
final_bhat = {}
final_mhat = {}

for rv in response_vars:
    yvar = is2hs[rv].dropna()
```

```
yscaler = StandardScaler()
ynew = yscaler.fit_transform(np.array(yvar).reshape(-1, 1)).flatten()
xvar = Xscaled[~is2hs[rv].isna()]
rv_lab = f"$\Delta${rv[6:]}"

# RFE w/ cross-validation to select number of features
lasso = lm.ElasticNet(alpha=0.01, l1_ratio=1.0, max_iter=5000)
feat2score = {}
feat2ci = {}
# determine stopping point
minf = 1
nf = 1
npt = len(ynew)
while (nf < npt-1) & (nf < nfeat_total):
    rfecv = RFECV(lasso, cv=nfold, scoring="neg_median_absolute_error",
                  min_features_to_select=minf)
    rfecv.fit(xvar, ynew)
    nf = rfecv.n_features_
    feat_in = ver.columns[rfecv.support_]
    feat_sort = ver.columns[np.argsort(rfecv.ranking_)]
    Xnew = rfecv.transform(xvar)

    # linear regression w/ reduce feature set, 5-fold cross-validation
    lr = lm.LinearRegression()
    cval = cross_validate(lr, Xnew, ynew, cv=nfold, return_estimator=True)
    total_scores = np.array([L.score(Xnew, ynew)
                             for L in cval["estimator"]])
    mean_score = np.mean(total_scores)
    adj_mean = 1 - ((1-mean_score)*(npt-1)/(npt-nf-1))
    ci_score = 2.216*np.std(total_scores)
    lr.fit(Xnew, ynew)
    ypred = lr.predict(Xnew)
    yp = yscaler.inverse_transform(ypred.reshape(-1, 1)).flatten()
    if nf < npt-1:
        feat2score[nf] = adj_mean
        feat2ci[nf] = (mean_score - ci_score, mean_score + ci_score)
    minf +=1
try:
    nf_select = max(feat2score, key=feat2score.get)
except:
    nf_select = 1

# perform RFECV with selected stopping point
rfecv = RFECV(lasso, cv=nfold, scoring="neg_median_absolute_error",
              min_features_to_select=nf_select)
rfecv.fit(xvar, ynew)
feat_in = ver.columns[rfecv.support_]
```

```
    nf = rfecv.n_features_
    Xnew = rfecv.transform(xvar)

    lr = lm.LinearRegression()
    nfold = 5
    nout = 2
    cval = cross_validate(lr, Xnew, ynew, cv=nfold, return_estimator=True)
    total_scores = np.array([L.score(Xnew, ynew) for L in cval["estimator"]])
    mean_score = np.mean(total_scores)
    adj_mean = 1 - ((1-mean_score)*(npt-nout-1)/(npt-nout-nf-1))
    ci_score = 2.086*np.std(total_scores)
    lr.fit(Xnew, ynew)
    r2 = lr.score(Xnew, ynew)
    ypred = lr.predict(Xnew)
    # calculate intercept + coefficients for non-transformed variables
    sy = yscaler.scale_
    my = yscaler.mean_
    sx = rfecv.transform(xscaler.scale_.reshape(1,-1)).flatten()
    mx = rfecv.transform(xscaler.mean_.reshape(1, -1)).flatten()
    bhat = sy*lr.coef_/sx
    mhat = my - np.dot(mx, bhat)
    Xold = rfecv.transform(X)
    yhat = np.dot(Xold, bhat) + mhat

    yp = yscaler.inverse_transform(ypred.reshape(-1, 1)).flatten()
    yp_ci = 2.074*sy*np.sqrt((1-r2)*(npt-1)/(npt-2))

    final_nf[rv] = nf
    final_lm[rv] = lr
    final_ypred[rv] = yp
    final_adjr[rv] = adj_mean
    final_ci[rv] = ci_score
    final_yp_ci[rv] = yp_ci
    final_feat_in[rv] = feat_in
    final_bhat[rv] = bhat
    final_mhat[rv] = mhat[0]

    print("\n")
    print(" | ".join(["R.V.".center(10), "Coef. ", "Feature"]))
    for f in range(nf):
        if f == 0:
            print(30*"-")
            print(f"{rv:<10} | {bhat[f]:>6.3f} | {feat_in[f]}")
        else:
            print(10*" " + f" | {bhat[f]:>6.3f} | {feat_in[f]}")
    print(10*" " + f" | {mhat[0]:>6.3f} | Intercept")
```

```
# Plots for change in MCC rate coef.
sns.set(context="paper", style="ticks", font_scale=1.5)
fig, axs = plt.subplots(2, 3, figsize=(10, 6))
axs[1, 2].set_visible(False)
# renumber clusters to match new labels
new2old_map = {
    1: 3,
    2: 4,
    3: 0,
    4: 1,
    5: 2
}
for i in range(1, nK+1):
    rv = f"delta_k{new2old_map[i]}"
    rv_lab = f"$\Delta$k{i}"
    yvar = is2hs[rv].dropna()
    yp = final_ypred[rv]
    adj_mean = final_adjr[rv]
    ci_score = final_ci[rv]
    ax0 = axs.flatten()[i-1]
    ax0.plot(yvar, yp, ls="None", marker="o")
    ymin, ymax = ax0.get_ylim()
    ax0.plot([ymin, ymax], [ymin, ymax], ls="--", color="k")
    ax0.set_xlabel(f"{rv_lab} Actual")
    ax0.set_ylabel(f"{rv_lab} Predicted")
    ann = ax0.text(0.25, 0.1,
                   f"Adj. $R^{2}$ = {adj_mean:3.2f} ($\pm${ci_score:3.2f})",
                   transform=ax0.transAxes, size="small",
                   backgroundcolor=(1., 1., 1., 0.6), in_layout=False,
                   linespacing=1.0, va="center")
    ax0.set_title(f"{final_nf[rv]} Features")
plt.tight_layout(pad=0.4, w_pad=0.8, h_pad=1.05)
# plt.show()
figname = fprefix + "Parity_Delta_k.svg"
plt.savefig(os.path.join(fig_dir, figname), facecolor="None",
            bbox_inches="tight", dpi=300)
plt.close("all")

# Classifier ROC & PRC
sns.set(context="paper", style="ticks", font_scale=1.5)
roc_fig, roc_axs = plt.subplots(2, 3, figsize=(10, 6), sharex=True,
                                sharey=True)
roc_axs[1, 2].set_visible(False)
prc_fig, prc_axs = plt.subplots(2, 3, figsize=(10, 6), sharex=True,
                                sharey=True)
prc_axs[1, 2].set_visible(False)
for i in range(1, nK+1):
    rv = f"delta_k{new2old_map[i]}"
    rv_lab = f"$\Delta$k{i}"
    yvar = is2hs[rv].dropna()
    yp = final_ypred[rv]
    y_true = yvar > 0
    fpr, tpr, thresh = roc_curve(y_true, yp)
    roc_auc = roc_auc_score(y_true, yp)
    roc_ax0 = roc_axs.flatten()[i-1]
    roc_ax0.plot(fpr, tpr)
    roc_ax0.plot([0, 1], [0, 1], ls="--", color="k")
    roc_ax0.set_xlabel(f"FPR")
    roc_ax0.set_ylabel(f"TPR")
    roc_ax0.set_title(f"{rv_lab}")
    roc_ax0.text(0.5, 0.15, f"AUC = {roc_auc:3.2f}",
                 transform=roc_ax0.transAxes)
    prec, recall, thresh = precision_recall_curve(y_true, yp)
    prc_auc = auc(recall, prec)
    prc_ax0 = prc_axs.flatten()[i-1]
    prc_ax0.plot(recall, prec)
    prc_ax0.plot([0, 1], [0.5, 0.5], ls="--", color="k")
    prc_ax0.set_xlabel(f"Recall")
    prc_ax0.set_ylabel(f"Precision")
    prc_ax0.set_title(f"{rv_lab}")
    prc_ax0.text(0.2, 0.15, f"AUC = {prc_auc:3.2f}",
                 transform=prc_ax0.transAxes)
roc_fig.tight_layout(pad=0.4, w_pad=0.8, h_pad=1.05)
prc_fig.tight_layout(pad=0.4, w_pad=0.8, h_pad=1.05)
# plt.show()
figname = fprefix + "ROC_k.svg"
roc_fig.savefig(os.path.join(fig_dir, figname), facecolor="None",
                bbox_inches="tight", dpi=300)
figname = fprefix + "PR_k.svg"
prc_fig.savefig(os.path.join(fig_dir, figname), facecolor="None",
                bbox_inches="tight", dpi=300)
plt.close("all")

# Plots for non-clearable activity
fig, axs = plt.subplots(2, 3, figsize=(10, 6))
axs[1, 2].set_visible(False)
for i in range(1, nK+1):
    rv = f"delta_flr{new2old_map[i]}"
    rv_lab = rv_lab = f"$\Delta$N{i}"
    yvar = is2hs[rv].dropna()
    yp = final_ypred[rv]
    adj_mean = final_adjr[rv]
    ci_score = final_ci[rv]
    ax0 = axs.flatten()[i-1]
    ax0.plot(yvar, yp, ls="None", marker="o")
    ymin, ymax = ax0.get_ylim()
    ax0.plot([ymin, ymax], [ymin, ymax], ls="--", color="k")
    ax0.set_xlabel(f"{rv_lab} Actual")
    ax0.set_ylabel(f"{rv_lab} Predicted")
    ax0.text(0.25, 0.1,
             f"Adj. $R^{2}$ = {adj_mean:3.2f} ($\pm${ci_score:3.2f})",
             transform=ax0.transAxes, size="small",
             backgroundcolor=(1., 1., 1., 0.6), in_layout=False,
             linespacing=1.0, va="center")
    ax0.set_title(f"{final_nf[rv]} Features")
plt.tight_layout(pad=0.4, w_pad=0.8, h_pad=1.05)
# plt.show()
figname = fprefix + "Parity_Delta_flr.svg"
plt.savefig(os.path.join(fig_dir, figname), facecolor="None",
            bbox_inches="tight", dpi=300)
plt.close("all")

# Classifier ROC & PRC
sns.set(context="paper", style="ticks", font_scale=1.4)
roc_fig, roc_axs = plt.subplots(2, 3, figsize=(10, 6), sharex=True,
                                sharey=True)
roc_axs[1, 2].set_visible(False)
prc_fig, prc_axs = plt.subplots(2, 3, figsize=(10, 6), sharex=True,
                                sharey=True)
prc_axs[1, 2].set_visible(False)
for i in range(1, nK+1):
    rv = f"delta_flr{new2old_map[i]}"
    rv_lab = f"$\Delta$N{i}"
    yvar = is2hs[rv].dropna()
    yp = final_ypred[rv]
    y_true = yvar < 0
    fpr, tpr, thresh = roc_curve(y_true, -yp)
    roc_auc = roc_auc_score(y_true, -yp)
    roc_ax0 = roc_axs.flatten()[i-1]
    roc_ax0.plot(fpr, tpr)
    roc_ax0.plot([0, 1], [0, 1], ls="--", color="k")
    roc_ax0.set_xlabel(f"FPR")
    roc_ax0.set_ylabel(f"TPR")
    roc_ax0.set_title(f"{rv_lab}")
    roc_ax0.text(0.5, 0.15, f"AUC = {roc_auc:3.2f}",
                 transform=roc_ax0.transAxes)
    prec, recall, thresh = precision_recall_curve(y_true, -yp)
    prc_auc = auc(recall, prec)
    prc_ax0 = prc_axs.flatten()[i-1]
    prc_ax0.plot(recall, prec)
    prc_ax0.plot([0, 1], [0.5, 0.5], ls="--", color="k")
    prc_ax0.set_xlabel(f"Recall")
    prc_ax0.set_ylabel(f"Precision")
    prc_ax0.set_title(f"{rv_lab}")
    prc_ax0.text(0.2, 0.15, f"AUC = {prc_auc:3.2f}",
                 transform=prc_ax0.transAxes)
roc_fig.tight_layout(pad=0.4, w_pad=0.8, h_pad=1.05)
prc_fig.tight_layout(pad=0.4, w_pad=0.8, h_pad=1.05)
# plt.show()
figname = fprefix + "ROC_flr.svg"
roc_fig.savefig(os.path.join(fig_dir, figname), facecolor="None",
                bbox_inches="tight", dpi=300)
figname = fprefix + "PR_flr.svg"
prc_fig.savefig(os.path.join(fig_dir, figname), facecolor="None",
                bbox_inches="tight", dpi=300)
plt.close("all")

fig, axs = plt.subplots(2, 3, figsize=(10, 6))
axs[1, 2].set_visible(False)
for i in range(1, nK+1):
    rv = f"delta_C{new2old_map[i]}"
    rv_lab = rv_lab = f"$\Delta$C{i}"
    yvar = is2hs[rv].dropna()
    yp = final_ypred[rv]
    adj_mean = final_adjr[rv]
    ci_score = final_ci[rv]
    ax0 = axs.flatten()[i-1]
    ax0.plot(yvar, yp, ls="None", marker="o")
    ymin, ymax = ax0.get_ylim()
    ax0.plot([ymin, ymax], [ymin, ymax], ls="--", color="k")
    ax0.set_xlabel(f"{rv_lab} Actual")
    ax0.set_ylabel(f"{rv_lab} Predicted")
    ax0.text(0.25, 0.1,
             f"Adj. $R^{2}$ = {adj_mean:3.2f}($\pm${ci_score:3.2f})",
             transform=ax0.transAxes, size="small",
             backgroundcolor=(1., 1., 1., 0.6), in_layout=False,
             linespacing=1.0, va="center")
    ax0.set_title(f"{final_nf[rv]} Features")
plt.tight_layout(pad=0.4, w_pad=0.8, h_pad=1.05)
# plt.show()
figname = fprefix + "Parity_Delta_C.svg"
plt.savefig(os.path.join(fig_dir, figname), facecolor="None",
            bbox_inches="tight", dpi=300)
plt.close("all")

# Plot feature importance
all_rv = list(final_feat_in.keys())
all_feat = []
```

```
for rv in all_rv:
    feat_in = list(final_feat_in[rv])
    all_feat = all_feat + feat_in
all_feat = set(all_feat)
feat_imp = pd.DataFrame(index=all_feat, columns=all_rv)
raw_model = pd.DataFrame(index=all_feat.union({"Intercept"}), columns=all_rv)
for rv in all_rv:
    for f, feat in enumerate(final_feat_in[rv]):
        feat_imp.loc[feat, rv] = abs(final_lm[rv].coef_[f])
        raw_model.loc[feat, rv] = final_bhat[rv][f]
    raw_model.loc["Intercept", rv] = final_mhat[rv]
col_reorder = [3, 4, 5, 1, 2, 8, 9, 10, 6, 7, 13, 14, 15, 11, 12]
new_cols = [col for (i, col) in sorted(zip(col_reorder,
                                           feat_imp.columns.tolist()))]
feat_imp = feat_imp[new_cols]
raw_model = raw_model[new_cols]
# raw_model.to_json(fprefix + "raw_model.json")

dk = feat_imp[new_cols[0:5]].dropna(axis=0, how="all")
dflr = feat_imp[new_cols[5:10]].dropna(axis=0, how="all")
dC = feat_imp[new_cols[10:]].dropna(axis=0, how="all")
dk_tot_imp = dk.sum(axis=1).sort_values(ascending=False)
dflr_tot_imp = dflr.sum(axis=1).sort_values(ascending=False)
dC_tot_imp = dC.sum(axis=1).sort_values(ascending=False)
dk = dk.loc[dk_tot_imp.index].fillna(0)
dflr = dflr.loc[dflr_tot_imp.index].fillna(0)
dC = dC.loc[dC_tot_imp.index].fillna(0)
dk_cum = dk.cumsum(axis=1)
dflr_cum = dflr.cumsum(axis=1)
dC_cum = dC.cumsum(axis=1)
dk_names = [r"$\Delta$"+f"k{i}" for i in range(1, 6)]
dflr_names = [r"$\Delta$"+f"N{i}" for i in range(1, 6)]
dC_names = [r"$\Delta$"+f"C{i}" for i in range(1, 6)]

sns.set(context="paper", style="ticks", font_scale=1.5)
bar_colors = plt.colormaps["Blues_r"](np.linspace(0.2, 0.8, 5))
# Delta ks
fig, ax = plt.subplots(figsize=(8, 8))
drep = [n.replace("delta_", r"$\Delta$") for n in dk.index]
rownames = [n.replace(" (mean)", "") for n in drep]
ax.invert_yaxis()
ax.set_xticks([])
ax.set_xlabel("Relative Importance")
ax.set_xlim(0, dk_tot_imp.max())
for i, (name, color) in enumerate(zip(dk_names, bar_colors)):
    widths = dk.iloc[:, i]
    starts = dk_cum.iloc[:, i] - widths
    rects = ax.barh(rownames, widths, left=starts, height=0.75, label=name,
                    color=color)
ax.legend(ncol=5, bbox_to_anchor=(0.5, 1), loc=8)
plt.tight_layout()
# plt.show()
figname = fprefix + "RelImport_dk.svg"
plt.savefig(os.path.join(fig_dir, figname), facecolor="None",
            bbox_inches="tight", dpi=300)
plt.close("all")

# Delta flr
fig, ax = plt.subplots(figsize=(8, 8))
drep = [n.replace("delta_", r"$\Delta$") for n in dflr.index]
rownames = [n.replace(" (mean)", "") for n in drep]
ax.invert_yaxis()
ax.set_xticks([])
ax.set_xlabel("Relative Importance")
ax.set_xlim(0, dflr_tot_imp.max())
for i, (name, color) in enumerate(zip(dflr_names, bar_colors)):
    widths = dflr.iloc[:, i]
    starts = dflr_cum.iloc[:, i] - widths
    rects = ax.barh(rownames, widths, left=starts, height=0.75, label=name,
                    color=color)
ax.legend(ncol=5, bbox_to_anchor=(0.5, 1), loc=8)
plt.tight_layout()
# plt.show()
figname = fprefix + "RelImport_dflr.svg"
plt.savefig(os.path.join(fig_dir, figname), facecolor="None",
            bbox_inches="tight", dpi=300)
plt.close("all")

# Delta C
fig, ax = plt.subplots(figsize=(8, 8))
drep = [n.replace("delta_", r"$\Delta$") for n in dC.index]
rownames = [n.replace(" (mean)", "") for n in drep]
ax.invert_yaxis()
ax.set_xticks([])
ax.set_xlabel("Relative Importance")
ax.set_xlim(0, dC_tot_imp.max())
for i, (name, color) in enumerate(zip(dC_names, bar_colors)):
    widths = dC.iloc[:, i]
    starts = dC_cum.iloc[:, i] - widths
    rects = ax.barh(rownames, widths, left=starts, height=0.75, label=name,
                    color=color)
ax.legend(ncol=5, bbox_to_anchor=(0.5, 1), loc=8)
plt.tight_layout()
# plt.show()
```

```
figname = fprefix + "RelImport_dC.svg"
plt.savefig(os.path.join(fig_dir, figname), facecolor="None",
            bbox_inches="tight", dpi=300)
plt.close("all")
```

Appendix C Statistical Submodels

C.1 All Variables Included

Multivariable linear models to estimate the change in MCC rate coefficients and non-clearable activity from IS to HS days for CF individuals are presented in Chapter 4, Tables 9 and 10. A multivariable linear model to estimate the change in Tc-SC concentration at 10 min is presented in Table 12 below.

Table 12: Coefficients and intercept for multivariable linear regression models to estimate change in Tc-SC concentration at 10 min from IS to HS days using all variables.

		Response Variable (% Total Activity)				
		ΔC_1	ΔC_2	ΔC_3	ΔC_4	ΔC_5
Feature	**Units**	**Coefficient**				
Sweat Cl	μM	-0.0772	-0.0709	0.0464	-0.171	0.0426
Cell ABS	% cleared/day	-0.163	-0.240	0.225	-0.160	—
Norm. V_{ap}	%/day	-8.34	—	—	23.2	6.90
LCI	—	0.537	—	—	1.48	-0.540
P_{wcb} ($\times 10^{-4}$)	μm/min	-6.17	-1.89	—	2.65	—
P_{bkca}	μm/min	—	—	—	-3.91	0.955
FEV_1	% of predicted	—	—	—	-0.0653	-0.117
FRAP	τ/τ_{saline}	—	—	2.67	-2.59	-1.27
Age	years	—	—	—	-0.445	—
P_{pcl}	μm/min	—	—	—	1.63	-1.15
P_{pk}	μm/min	4.55	—	—	—	—
BMI	kg/m^2	—	0.350	—	—	-0.136
P_{acc}	μm/min	—	—	—	21.5	—
J_{nkcc}	mmol/(min·m^2)	-0.220	—	—	—	—
P_{bcl}	μm/min	—	—	1.32	—	—
$\Delta FEF_{25\text{-}75}$	% of predicted	0.222	—	—	—	—
ΔFEV_1	% of predicted	-0.286	—	—	—	—
P_{wab}	μm/min	—	—	—	9.00×10^{-5}	—

C.2 Clinical and *In Vitro* Variables Included

Multivariable linear models to estimate the change in MCC rate coefficients, non-clearable activity, and Tc-SC concentration at 10 min from IS to HS days using only clinical and *in vitro* variables as inputs are presented in Tables 13, 14, and 15, respectively.

Table 13: Coefficients and intercept for multivariable linear regression models to estimate change MCC rate coefficients from IS to HS days using clinical and *in vitro* data.

		Response Variable (min^{-1})				
		Δk_1	Δk_2	Δk_3	Δk_4	Δk_5
Feature	**Units**	**Coefficient**				
Sweat Cl	μM	—	0.0225	—	—	—
Cell ABS	% cleared/day	—	—	0.0110	—	—
Norm. V_{ap}	%/day	—	3.98	0.631	—	—
LCI	—	-2.58	-0.589	—	—	-0.0411
FEV_1	% of predicted	-0.621	-0.139	-8.55 $\times 10^{-3}$	—	—
FRAP	τ/τ_{saline}	10.6	2.60	—	0.298	—
Age	years	—	—	—	—	0.0110
BMI	kg/m^2	—	0.116	-0.0412	0.0506	—
$\Delta FEF_{25\text{-}75}$	% of predicted	-0.589	-0.153	—	—	—
ΔFEV_1	% of predicted	—	0.645	—	0.0334	0.0119
Δ(Cell ABS)	% cleared/day	—	—	—	-0.0390	-0.0157
Intercept	min^{-1}	37.0	2.03	0.989	-2.36	0.0196

Table 14: Coefficients and intercept for multivariable linear regression models to estimate change in non-clearable activity from IS to HS days using clinical and *in vitro* data.

		Response Variable (% Total Activity)				
		ΔNC_1	ΔNC_2	ΔNC_3	ΔNC_4	ΔNC_5
Feature	**Units**	**Coefficient**				
Cell ABS	% cleared/day	—	-0.118	0.217	—	—
Norm. $\mathrm{V_{ap}}$	%/day	—	—	3.99	—	4.03
LCI	—	0.816	0.747	—	—	-0.381
$\mathrm{FEV_1}$	% of predicted	—	0.0719	—	—	-0.0275
FRAP	τ/τ_{saline}	—	-1.44	—	-4.46	-1.53
Age	years	-0.337	-0.137	—	—	0.0628
BMI	$\mathrm{kg/m^2}$	0.467	0.276	—	—	-0.236
$\Delta\mathrm{FEF_{25\text{-}75}}$	% of predicted	—	—	0.135	—	—
$\Delta\mathrm{FEV_1}$	% of predicted	—	—	-0.366	—	—
Δ(Cell ABS)	% cleared/day	0.132	—	0.213	0.187	—
Intercept	% Total Activity	-10.6	-8.20	-10.0	15.5	12.7

Table 15: Coefficients and intercept for multivariable linear regression models to estimate change in Tc-SC concentration at 10 min from IS to HS days using clinical and *in vitro* data.

		Response Variable (% Total Activity)				
		ΔC_1	ΔC_2	ΔC_3	ΔC_4	ΔC_5
Feature	**Units**	**Coefficient**				
Cell ABS	% cleared/day	—	-0.168	0.242	—	—
Norm. V_{ap}	%/day	—	—	—	9.71	8.10
LCI	—	—	0.761	—	—	-0.667
Sweat Cl	μM	-0.0696	-0.0794	0.0564	—	—
FEV_1	% of predicted	—	—	—	-0.0952	-0.0953
FRAP	τ/τ_{saline}	—	—	1.47	—	-1.25
Age	years	—	-0.154	0.0935	-0.184	—
BMI	kg/m^2	—	0.454	-0.0957	—	-0.194
$\Delta FEF_{25\text{-}75}$	% of predicted	—	—	0.147	—	—
ΔFEV_1	% of predicted	—	—	-0.333	—	—
Δ(Cell ABS)	% cleared/day	—	—	0.193	—	0.0430
Intercept	% Total Activity	6.29	0.851	-19.1	8.57	20.2

C.3 Clinical Variables Only

Multivariable linear models to estimate the change in MCC rate coefficients, non-clearable activity, and Tc-SC concentration at 10 min from IS to HS days using only clinical variables as inputs are presented in Tables 16, 17, and 18, respectively.

Table 16: Coefficients and intercept for multivariable linear regression models to estimate change MCC rate coefficients from IS to HS days using only clinical data.

		Response Variable (min^{-1})				
		Δk_1	Δk_2	Δk_3	Δk_4	Δk_5
Feature	**Units**	**Coefficient**				
Sweat Cl	μM	—	2.32×10^{-3}	—	—	—
LCI	—	-1.14	-0.598	—	—	-0.0523
FEV$_1$	% of predicted	-0.398	-0.0966	-3.07×10^{-3}	-5.60×10^{-3}	—
Age	years	—	0.0497	9.79×10^{-3}	—	0.0147
BMI	kg/m^2	—	—	-0.0269	0.0563	—
ΔFEF$_{25\text{-}75}$	% of predicted	-0.298	-0.104	2.65×10^{-3}	—	—
ΔFEV$_1$	% of predicted	-0.389	—	4.14×10^{-3}	—	—
Intercept	min^{-1}	45.2	13.7	0.563	-0.389	0.280

Table 17: Coefficients and intercept for multivariable linear regression models to estimate change in non-clearable activity from IS to HS days using only clinical data.

		Response Variable (% Total Activity)				
		ΔNC_1	ΔNC_2	ΔNC_3	ΔNC_4	ΔNC_5
Feature	**Units**	**Coefficient**				
LCI	—	0.701	—	—	—	—
FEV_1	% of predicted	—	—	—	—	-0.0180
Age	years	-0.303	-0.0820	—	-0.132	—
BMI	kg/m^2	0.351	—	—	—	—
ΔFEV_1	% of predicted	—	—	-0.166	0.315	—
Intercept	% Total Activity	-9.83	-1.43	-1.61	0.232	0.637

Table 18: Coefficients and intercept for multivariable linear regression models to estimate change in Tc-SC concentration at 10 min from IS to HS days using only clinical data.

		Response Variable (% Total Activity)				
		ΔC_1	ΔC_2	ΔC_3	ΔC_4	ΔC_5
Feature	**Units**	**Coefficient**				
LCI	—	—	1.08	—	0.954	-0.639
Sweat Cl	μM	-0.0618	-0.0828	—	-0.0434	0.0408
FEV_1	% of predicted	—	—	—	-0.0461	-0.0450
Age	years	—	-0.188	—	-0.333	0.0882
BMI	kg/m^2	0.178	0.371	—	0.444	—
$\Delta\text{FEF}_{25\text{-}75}$	% of predicted	—	—	—	—	0.0615
ΔFEV_1	% of predicted	—	—	-0.153	—	—
Intercept	% Total Activity	1.35	-7.04	0.183	-2.32	3.87

Bibliography

[1] G. S. Sawicki, D. E. Sellers, and W. M. Robinson. “High treatment burden in adults with cystic fibrosis: Challenges to disease self-management”. *Journal of Cystic Fibrosis* 8.2 (Mar. 2009), pp. 91–96. DOI: 10.1016/j.jcf.2008.09.007.

[2] G. Davies, N. J. Rowbotham, S. Smith, et al. “Characterising burden of treatment in cystic fibrosis to identify priority areas for clinical trials”. *Journal of Cystic Fibrosis* 19.3 (2020), pp. 499–502. DOI: 10.1016/j.jcf.2019.10.025.

[3] M. N. Eakin and K. A. Riekert. “The impact of medication adherence on lung health outcomes in cystic fibrosis”. *Current Opinion in Pulmonary Medicine* 19.6 (Nov. 2013), pp. 687–691. DOI: 10.1097/MCP.0b013e3283659f45.

[4] R. A. Cameron, D. Office, J. Matthews, et al. “Treatment Preference Among People With Cystic Fibrosis: The Importance of Reducing Treatment Burden”. *Chest* 162.6 (2022), pp. 1241–1254. DOI: 10.1016/j.chest.2022.07.008.

[5] N. J. Rowbotham, S. Smith, P. A. Leighton, et al. “The top 10 research priorities in cystic fibrosis developed by a partnership between people with CF and healthcare providers”. *Thorax* 73.4 (2018), pp. 388–390. DOI: 10.1136/thoraxjnl-2017-210473.

[6] C. F. Foundation. *Key Research Priorities for Applicants*. 2019. URL: https://www.cff.org/key-research-priorities-applicants.

[7] J. C. Nawroth, A. M. Van Der Does, A. Ryan, et al. “Multiscale mechanics of mucociliary clearance in the lung”. *Philosophical Transactions of the Royal Society B: Biological Sciences* 375.1792 (2020), pp. 1–8. DOI: 10.1098/rstb.2019.0160.

[8] D. B. Hill, B. Button, M. Rubinstein, et al. “Physiology and Pathophysiology of Human Airway Mucus”. *Physiological Reviews* 102.4 (2022), pp. 1757–1836. DOI: 10.1152/physrev.00004.2021.

[9] B. Button, L.-H. Cai, C. Ehre, et al. “A Periciliary Brush Promotes the Lung Health by Separating the Mucus Layer from Airway Epithelia”. *Science* 337.6097 (Aug. 2012), pp. 937–941. DOI: 10.1126/science.1223012. eprint: NIHMS150003.

[10] T. E. Corcoran, A. S. Huber, S. L. Hill, et al. "Mucociliary Clearance Differs in Mild Asthma by Levels of Type 2 Inflammation". *Chest* 160.5 (Nov. 2021), pp. 1604–1613. DOI: 10.1016/j.chest.2021.05.013.

[11] J. C. Hogg, P. D. Paré, and T. L. Hackett. "The contribution of small airway obstruction to the pathogenesis of chronic obstructive pulmonary disease". *Physiological Reviews* 97.2 (2017), pp. 529–552. DOI: 10.1152/physrev.00025.2015.

[12] R. Robinot, M. Hubert, G. D. de Melo, et al. "SARS-CoV-2 infection induces the dedifferentiation of multiciliated cells and impairs mucociliary clearance". *Nature Communications* 12.1 (July 2021), p. 4354. DOI: 10.1038/s41467-021-24521-x.

[13] A. G. Henderson, C. Ehre, B. Button, et al. "Cystic fibrosis airway secretions exhibit mucin hyperconcentration and increased osmotic pressure". *Journal of Clinical Investigation* 124.7 (2014), pp. 3047–3060. DOI: 10.1172/JCI73469.

[14] E. Weibel and D. Gomez. "Architecture of the human lung". *Science* 137.3530 (1962), pp. 577–585.

[15] E. Toskala, S. M. Smiley-Jewell, V. J. Wong, et al. "Temporal and spatial distribution of ciliogenesis in the tracheobronchial airways of mice". *American Journal of Physiology - Lung Cellular and Molecular Physiology* 289.3 33-3 (2005), pp. 454–459. DOI: 10.1152/ajplung.00036.2005.

[16] K. De Boeck, A. Zolin, H. Cuppens, et al. "The relative frequency of CFTR mutation classes in European patients with cystic fibrosis". *Journal of Cystic Fibrosis* 13.4 (2014), pp. 403–409. DOI: 10.1016/j.jcf.2013.12.003.

[17] US CF Foundation, Johns Hopkins University, The Hospital for Sick Children. *The Clinical and Functional TRanslation of CFTR (CFTR2).* 2011. URL: http://cftr2.org (visited on 12/28/2020).

[18] *Cystic Fibrosis Foundation Patient Registry 2021 Annual Data Report.* Tech. rep. Bethesda, MD: Cystic Fibrosis Foundation, 2022.

[19] L. Medina, S. Sabo, and J. Vespa. "Living Longer: Historical and Projected Life Expectancy in the United States, 1960 to 2060". *U.S. Census Bureau* (2020), pp. 1–27.

[20] N. A. O'Leary, M. W. Wright, J. R. Brister, et al. "Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation". *Nucleic Acids Research* 44.D1 (Jan. 2016), pp. D733–D745. DOI: 10.1093/nar/gkv1189.

[21] T. C. Hwang and K. L. Kirk. "The CFTR Ion channel: Gating, regulation, and anion permeation". *Cold Spring Harbor Perspectives in Medicine* 3.1 (2013), pp. 1–15. DOI: 10.1101/cshperspect.a009498.

[22] K. Kunzelmann, R. Schreiber, and H. B. Hadorn. "Bicarbonate in cystic fibrosis". *Journal of Cystic Fibrosis* 16.6 (2017), pp. 653–662. DOI: 10.1016/j.jcf.2017.06.005.

[23] F. Serrano Castillo, C. A. Bertrand, M. M. Myerburg, et al. "A physiologically-motivated model of cystic fibrosis liquid and solute transport dynamics across primary human nasal epithelia". *Journal of Pharmacokinetics and Pharmacodynamics* 0 (2019). DOI: 10.1007/s10928-019-09649-0.

[24] D. L. O'Donoghue, V. Dua, G. W. J. Moss, et al. "Increased apical Na + permeability in cystic fibrosis is supported by a quantitative model of epithelial ion transport". *The Journal of Physiology* 591.15 (Aug. 2013), pp. 3681–3692. DOI: 10.1113/jphysiol.2013.253955.

[25] F. Serrano Castillo, C. A. Bertrand, T. E. Corcoran, et al. "A Dynamic Model of Cystic Fibrosis Airway Epithelium Electrophysiology". *IFAC-PapersOnLine* 51.19 (Jan. 2018), pp. 94–97. DOI: 10.1016/J.IFACOL.2018.09.027.

[26] C. A. Bertrand, S. Mitra, S. K. Mishra, et al. "The CFTR trafficking mutation F508del inhibits the constitutive activity of SLC26A9". *American Journal of Physiology - Lung Cellular and Molecular Physiology* 312.6 (2017), pp. L12–L925. DOI: 10.1152/ajplung.00178.2016.

[27] P. Jourdain, F. Becq, S. Lengacher, et al. "The human CFTR protein expressed in CHO cells activates an aquaporin 3 in a cAMP dependent pathway: study by Digital Holographic Microscopy". *Journal of Cell Science* (Jan. 2013). DOI: 10.1242/jcs.133629.

[28] M. O. Henke, G. John, M. Germann, et al. "MUC5AC and MUC5B mucins increase in cystic fibrosis airway secretions during pulmonary exacerbation". *American Journal of Respiratory and Critical Care Medicine* 175.8 (2007), pp. 816–821. DOI: 10.1164/rccm.200607-1011OC.

[29] R. Suri. "The Use of Human Deoxyribonuclease (rhDNase) in the Management of Cystic Fibrosis". *BioDrugs* 19.3 (2005), pp. 135–144. DOI: 10.2165/00063030-200519030-00001.

[30] W. D. Bennett, B. L. Laube, T. Corcoran, et al. "Multisite Comparison of Mucociliary and Cough Clearance Measures Using Standardized Methods". *Journal of Aerosol Medicine and Pulmonary Drug Delivery* 26.3 (2013). DOI: 10.1089/jamp.2011.0909.

[31] A. T. Trimble, A. Whitney Brown, B. L. Laube, et al. "Hypertonic saline has a prolonged effect on mucociliary clearance in adults with cystic fibrosis". *Journal of Cystic Fibrosis* 17.5 (2018), pp. 650–656. DOI: 10.1016/j.jcf.2018.01.001.

[32] W. D. Bennett, J. Wu, F. Fuller, et al. "Duration of action of hypertonic saline on mucociliary clearance in the normal lung". *Journal of Applied Physiology* 118.12 (2015), pp. 1443–1561. DOI: 10.1152/japplphysiol.00404.2014.

[33] S. H. Donaldson, W. D. Bennett, K. L. Zeman, et al. "Mucus clearance and lung function in cystic fibrosis with hypertonic saline." *The New England journal of medicine* 354.3 (Jan. 2006), pp. 241–50. DOI: 10.1056/NEJMoa043891.

[34] T. E. Corcoran, C. A. Bertrand, M. M. Myerburg, et al. "Nasal epithelial cell culture FRAP predicts cystic fibrosis therapeutic response". *ERJ Open Research* (2022), pp. 00382–2022. DOI: 10.1183/23120541.00382-2022.

[35] M. R. Markovetz, T. E. Corcoran, L. W. Locke, et al. "A physiologically-motivated compartment-based model of the effect of inhaled hypertonic saline on mucociliary clearance and liquid transport in cystic fibrosis". *PLoS ONE* 9.11 (2014), pp. 1–13. DOI: 10.1371/journal.pone.0111972.

[36] W. D. Bennett, M. Xie, K. Zeman, et al. "Heterogeneity of particle deposition by pixel analysis of 2D gamma scintigraphy images". *Journal of Aerosol Medicine and Pulmonary Drug Delivery* 28.3 (2015), pp. 211–218. DOI: 10.1089/jamp.2013.1095.

[37] T. E. Corcoran, K. M. Thomas, M. M. Myerburg, et al. “Absorptive clearance of DTPA as an aerosol-based biomarker in the cystic fibrosis airway”. *European Respiratory Journal* 35.4 (2010), pp. 781–786. DOI: 10.1183/09031936.00059009.

[38] S. Newman, W. D. Bennett, M. Biddiscombe, et al. “Standardization of techniques for using planar (2D) imaging for aerosol deposition assessment of orally inhaled products”. *Journal of Aerosol Medicine and Pulmonary Drug Delivery* 25.SUPPL.1 (2012), pp. 10–28. DOI: 10.1089/jamp.2012.1Su4.

[39] M. T. Emmerich and A. H. Deutz. “A tutorial on multiobjective optimization: fundamentals and evolutionary methods”. *Natural Computing* 17.3 (2018), pp. 585–609. DOI: 10.1007/s11047-018-9685-y.

[40] H. T. Kung, F. Luccio, and F. P. Preparata. “On Finding the Maxima of a Set of Vectors”. *Journal of the ACM (JACM)* 22.4 (1975), pp. 469–476. DOI: 10.1145/321906.321910.

[41] B. Li, J. Li, K. Tang, et al. “Many-objective evolutionary algorithms: A survey”. *ACM Computing Surveys* 48.1 (2015). DOI: 10.1145/2792984.

[42] J. Handl, D. B. Kell, and J. Knowles. “Multiobjective optimization in bioinformatics and computational biology”. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 4.2 (2007), pp. 279–291. DOI: 10.1109/TCBB.2007.070203.

[43] L. Alcoforado, A. Dornelas de Andrade, J. L. Herraiz, et al. “Anatomically Based Analysis of Radioaerosol Distribution in Pulmonary Scintigraphy: A Feasibility Study in Asthmatics”. *Journal of Aerosol Medicine and Pulmonary Drug Delivery* 31.5 (Oct. 2018), pp. 298–310. DOI: 10.1089/jamp.2017.1403.

[44] E. E. Greenblatt, T. Winkler, R. S. Harris, et al. “Analysis of three-dimensional aerosol deposition in pharmacologically relevant terms: Beyond black or white rois”. *Journal of Aerosol Medicine and Pulmonary Drug Delivery* 28.2 (Apr. 2015), pp. 116–129. DOI: 10.1089/jamp.2013.1120.

[45] C. Chen. *Linear System Theory and Design.* The Oxford Series in Electrical and Computer Engineering. Oxford University Press, 2014.

[46] A. Raue, C. Kreutz, T. Maiwald, et al. "Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood". *Bioinformatics* 25.15 (2009), pp. 1923–1929. DOI: `10.1093/bioinformatics/btp358`.

[47] S. S. Wilks. "The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses". *The Annals of Mathematical Statistics* 9.1 (1938), pp. 60–62. DOI: `10.1214/aoms/1177732360`.

[48] A. Raue, B. Steiert, M. Schelker, et al. "Data2Dynamics: a modeling environment tailored to parameter estimation in dynamical systems: Fig. 1." *Bioinformatics* 31.21 (Nov. 2015), pp. 3558–3560. DOI: `10.1093/bioinformatics/btv405`.

[49] P. Royston. "Profile likelihood for estimation and confidence intervals". *Stata Journal* 7.3 (2007), pp. 376–387. DOI: `10.1177/1536867x0700700305`.

[50] L. Zhang, W. Dong, D. Zhang, et al. "Two-stage image denoising by principal component analysis with local pixel grouping". *Pattern Recognition* 43.4 (2010), pp. 1531–1549. DOI: `10.1016/j.patcog.2009.09.023`.

[51] L. Fan, F. Zhang, H. Fan, et al. "Brief review of image denoising techniques". *Visual Computing for Industry, Biomedicine, and Art* 2.1 (Dec. 2019), p. 7. DOI: `10.1186/s42492-019-0016-7`.

[52] M. Turk and A. Pentland. "Eigenfaces for Recognition". *Journal of Cognitive Neuroscience* 3.1 (Jan. 1991), pp. 71–86. DOI: `10.1162/jocn.1991.3.1.71`.

[53] V. Siless, K. Chang, B. Fischl, et al. "AnatomiCuts: Hierarchical clustering of tractography streamlines based on anatomical similarity". *NeuroImage* 166.March 2017 (2018), pp. 32–45. DOI: `10.1016/j.neuroimage.2017.10.058`.

[54] R. Socher, A. Maas, and C. D. Manning. "Spectral Chinese restaurant processes: Nonparametric clustering based on similarities". *Journal of Machine Learning Research* 15 (2011), pp. 698–706.

[55] U. Von Luxburg. "A tutorial on spectral clustering". *Statistics and Computing* 17.4 (2007), pp. 395–416. DOI: `10.1007/s11222-007-9033-z`. arXiv: `0711.0189`.

[56] B. Nicholson, J. D. Siirola, J.-P. Watson, et al. "pyomo.dae: a modeling and automatic discretization framework for optimization with differential and algebraic equations".

Mathematical Programming Computation 10.2 (June 2018), pp. 187–223. DOI: `10.1007/s12532-017-0127-0`.

[57] C. Kreutz, A. Raue, and J. Timmer. “Likelihood based observability analysis and confidence intervals for predictions of dynamic models”. *BMC Systems Biology* 6 (Sept. 2012). DOI: `10.1186/1752-0509-6-120`. arXiv: `1107.0013`.

[58] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. “Scikit-learn: Machine Learning in Python”. *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[59] F. Murtagh and P. Legendre. “Ward’s Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward’s Criterion?” *Journal of Classification* 31.3 (Oct. 2014), pp. 274–295. DOI: `10.1007/s00357-014-9161-z`.

[60] L. W. Locke, M. M. Myerburg, M. R. Markovetz, et al. “Quantitative imaging of airway liquid absorption in cystic fibrosis”. *European Respiratory Journal* 44.3 (2014), pp. 675–684. DOI: `10.1183/09031936.00220513`.

[61] P. Wark and V. M. Mcdonald. “Nebulised hypertonic saline for cystic fibrosis”. *Cochrane Database of Systematic Reviews* 2018.9 (2018). DOI: `10.1002/14651858.CD001506.pub4`.

[62] F. Serrano Castillo, C. A. Bertrand, M. M. Myerburg, et al. “A physiologically-motivated model of cystic fibrosis liquid and solute transport dynamics across primary human nasal epithelia”. *Journal of Pharmacokinetics and Pharmacodynamics* 0 (2019). DOI: `10.1007/s10928-019-09649-0`.

[63] L. A. Zhang, A. Urbano, G. Clermont, et al. “APT-MCMC, a C++/Python implementation of Markov Chain Monte Carlo for parameter identification”. *Computers and Chemical Engineering* 110 (2018), pp. 1–12. DOI: `10.1016/j.compchemeng.2017.11.011`.

[64] J. Lever, M. Krzywinski, and N. Altman. “Points of Significance: Regularization”. *Nature Methods* 13.10 (2016), pp. 803–804. DOI: `10.1038/nmeth.4014`.

[65] I. Guyon, J. Weston, S. Barnhill, et al. “Gene Selection for Cancer Classification using Support Vector Machines”. *Machine Learning* 46.1 (2002), pp. 389–422. DOI: `10.1023/A:1012487302797`.

[66] I. Guyon and A. Elisseeff. “An Introduction to Variable and Feature Selection”. *J. Mach. Learn. Res.* 3 (Mar. 2003), pp. 1157–1182. DOI: 10.5555/944919.944968.

[67] S. Malhotra, D. Hayes, and D. J. Wozniak. “Mucoid Pseudomonas aeruginosa and regional inflammation in the cystic fibrosis lung”. *Journal of Cystic Fibrosis* 18.6 (2019), pp. 796–803. DOI: 10.1016/j.jcf.2019.04.009.

[68] Z. Li, D. B. Sanders, M. J. Rock, et al. “Regional differences in the evolution of lung disease in children with cystic fibrosis”. *Pediatric Pulmonology* 47.7 (July 2012), pp. 635–640. DOI: 10.1002/ppul.21604.

[69] S. D. Davis, L. A. Fordham, A. S. Brody, et al. “Computed tomography reflects lower airway inflammation and tracks changes in early cystic fibrosis”. *American Journal of Respiratory and Critical Care Medicine* 175.9 (2007), pp. 943–950. DOI: 10.1164/rccm.200603-343OC.

[70] T. M. Martínez, C. J. Llapur, T. H. Williams, et al. “High-resolution computed tomography imaging of airway disease in infants with cystic fibrosis”. *American Journal of Respiratory and Critical Care Medicine* 172.9 (2005), pp. 1133–1138. DOI: 10.1164/rccm.200412-1665OC.

[71] A. S. Brody, J. S. Klein, P. L. Molina, et al. “High-resolution computed tomography in young patients with cystic fibrosis: Distribution of abnormalities and correlation with pulmonary function tests”. *Journal of Pediatrics* 145.1 (2004), pp. 32–38. DOI: 10.1016/j.jpeds.2004.02.038.

[72] M. R. Elkins, M. Robinson, B. R. Rose, et al. “A Controlled Trial of Long-Term Inhaled Hypertonic Saline in Patients with Cystic Fibrosis”. *New England Journal of Medicine* 354.3 (Jan. 2006), pp. 229–240. DOI: 10.1056/NEJMoa043900.

[73] N. Sood, W. D. Bennett, K. Zeman, et al. “Increasing Concentration of Inhaled Saline with or without Amiloride”. *American Journal of Respiratory and Critical Care Medicine* 167.2 (Jan. 2003), pp. 158–163. DOI: 10.1164/rccm.200204-293OC.

[74] E. Daviskas, S. D. Anderson, I. Gonda, et al. “Inhalation of hypertonic saline aerosol enhances mucociliary clearance in asthmatic and healthy subjects”. *European Respiratory Journal* 9.4 (1996), pp. 725–732. DOI: 10.1183/09031936.96.09040725.

[75] N. Jakeways, T. McKeever, S. A. Lewis, et al. “Relationship between FEV1 reduction and respiratory symptoms in the general population”. *European Respiratory Journal* 21.4 (2003), pp. 658–663. DOI: 10.1183/09031936.03.00069603.

[76] J. Davies, H. Sheridan, N. Bell, et al. “Assessment of clinical response to ivacaftor with lung clearance index in cystic fibrosis patients with a G551D- CFTR mutation and preserved spirometry: a randomised controlled trial”. *The Lancet Respiratory Medicine* 1.8 (Oct. 2013), pp. 630–638. DOI: 10.1016/S2213-2600(13)70182-6.

[77] S. Stanojevic, S. D. Davis, G. Retsch-Bogart, et al. “Progression of lung disease in preschool patients with cystic fibrosis”. *American Journal of Respiratory and Critical Care Medicine* 195.9 (2017), pp. 1216–1225. DOI: 10.1164/rccm.201610-2158OC.

[78] A. T. Lennox, S. L. Coburn, J. A. Leech, et al. “ATP12A promotes mucus dysfunction during Type 2 airway inflammation”. *Scientific Reports* 8.1 (2018), pp. 1–13. DOI: 10.1038/s41598-018-20444-8.

[79] D. B. Hill, R. F. Long, W. J. Kissner, et al. “Pathological mucus and impaired mucus clearance in cystic fibrosis patients result from increased concentration, not altered pH”. *European Respiratory Journal* 52.6 (Dec. 2018), p. 1801297. DOI: 10.1183/13993003.01297-2018.

[80] W. D. Bennett, A. Burbank, M. Almond, et al. “Acute and durable effect of inhaled hypertonic saline on mucociliary clearance in adult asthma”. *ERJ Open Research* 7.2 (2021), pp. 00062–2021. DOI: 10.1183/23120541.00062-2021.

[81] W. Mi, C. Zhang, H. Wang, et al. “Measurement and analysis of the tracheobronchial tree in Chinese population using computed tomography”. *PLoS ONE* 10.4 (2015), pp. 1–14. DOI: 10.1371/journal.pone.0123177.

[82] J. S. Fleming, J. Conway, M. J. Bennett, et al. “Quantitative Assessment of Mucociliary Clearance in Smokers with Mild-to-Moderate Chronic Obstructive Pulmonary Disease and Chronic Bronchitis from Planar Radionuclide Imaging Using the Change in Penetration Index”. *https://home.liebertpub.com/jamp* 32.4 (Aug. 2019), pp. 175–188. DOI: 10.1089/JAMP.2017.1441.

[83] A. L. Young, F. J. Bragman, B. Rangelov, et al. “Disease progression modeling in chronic obstructive pulmonary disease”. *American Journal of Respiratory and Critical Care Medicine* 201.3 (2020), pp. 294–302. DOI: 10.1164/rccm.201908-1600OC.

etdbib.bib