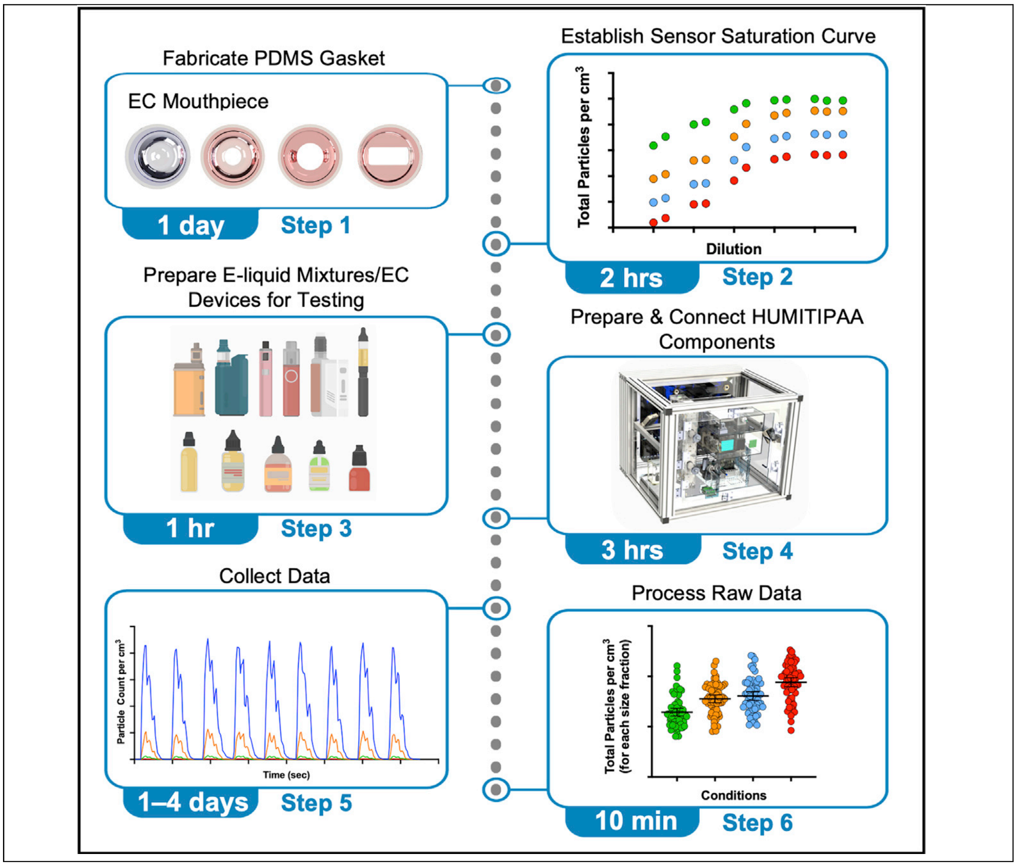
STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for the operation of a breathing and vaping biomimetic robot to delineate real-time inhaled particle profile of electronic cigarettes



We recently developed a robotic human vaping mimetic real-time particle analyzer (HUMITIPAA) to evaluate the impact of change in chemical constituents and breathing profiles of electronic cigarettes (ECs) on potential pulmonary toxicity. Here, we describe the fabrication procedure of EC mouthpiece(s), establishment of sensor saturation curve, and preparation of e-liquid and vaping device(s) for testing. We further detail steps for HUMITIPAA preparation and connection setup, followed by data collection and processing.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Rachel F. Bogdanoff, Alexander J. Kaiser, Kambez H. Benam

benamk@pitt.edu

Highlights

Fabrication procedure of EC mouthpiece(s) and establishment of sensor saturation curve

Preparation of e-liquid mixtures and electronic cigarette(s) for testing

Preparation and connection of HUMITIPAA components

Raw data collection with HUMITIPAA and its post-acquisition processing in MATLAB

Bogdanoff et al., STAR Protocols 3, 101885
December 16, 2022
https://doi.org/10.1016/j.xpro.2022.101885

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for the operation of a breathing and vaping biomimetic robot to delineate real-time inhaled particle profile of electronic cigarettes

Rachel F. Bogdanoff,[1] Alexander J. Kaiser,[2] and Kambez H. Benam[1,2,3,4,5,6,*]

[1]Division of Pulmonary, Allergy and Critical Care Medicine, Department of Medicine, University of Pittsburgh, Pittsburgh, PA 15213, USA

[2]Department of Bioengineering, University of Colorado Denver, Aurora, CO 80045, USA

[3]Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA 15219, USA

[4]Vascular Medicine Institute, University of Pittsburgh, Pittsburgh, PA 15213, USA

[5]Technical contact

[6]Lead contact

*Correspondence: benamk@pitt.edu

https://doi.org/10.1016/j.xpro.2022.101885

SUMMARY

We recently developed a robotic human vaping mimetic real-time particle analyzer (HUMITIPAA) to evaluate the impact of change in chemical constituents and breathing profiles of electronic cigarettes (ECs) on potential pulmonary toxicity. Here, we describe the fabrication procedure of EC mouthpiece(s), establishment of sensor saturation curve, and preparation of e-liquid and vaping device(s) for testing. We further detail steps for HUMITIPAA preparation and connection setup, followed by data collection and processing.

For complete details on the use and execution of this protocol, please refer to Kaiser et al. (2021).[1]

BEFORE YOU BEGIN

Background

The popularity of electronic cigarettes (ECs) has risen considerably in recent years. A growing concern has been EC use particularly among the youth population, with a 2019 study showing that 27.5% of US high school students reported using an EC in the last 30 days.[2] Despite gaining popularity in part due to public perception of ECs posing minimal health risks, many EC liquids contain a wide variety of compounds that have not been well-characterized for their toxicity when aerosolized and inhaled.[3,4] With EC technologies developing seemingly faster than the research on their safety, the development of systems to analyze the potential toxicological effects of ECs and their chemical constituents is paramount. In fact, there is a pressing need for better characterization of aerosols from ECs as they emerge.

We recently reported development of a bioinspired vaping robotic system (Human Vaping Mimetic Real-Time Particle Analyzer [HUMITIPAA]) to characterize quantity and distribution pattern of particulate matters (PM) at four size fractions (300 nm–1 μm, 1 μm–2.5 μm, 2.5 μm–4 μm, and 4 μm–10 μm) in EC aerosols in real-time, while emulating pathophysiological breathing and vaping topography.[1] The protocol below describes specific steps for analysis of vitamin E acetate (VEA) added to an e-liquid base (50:50 solution of propylene glycol – vegetable glycerin [PG:VG] in this case). However, the HUMITIPAA can be applied to study the impact of a variety of EC additives. Higher particle quantities would imply increased total particle inhaled into the lungs and may serve as a surrogate for enhanced possibility of pulmonary toxicity. Size distribution would allow one to partly deduce potential localization site(s) of particles.[5]

The HUMITIPAA consists of five units, four of which are interconnected subsystems – namely, (i) Vaping Robot, (ii) Dilution Robot, (iii) Breathing-Emulator, and (iv) PM Sensor, housed within (v) Inhalation Exposure Chamber; they are individually and collectively controlled to operate as a single system (Figure 1). The Inhalation Exposure Chamber (Figure 1A) uses a set of pressure-controlling pneumatics to contain all exposures and safely exhaust unneeded aerosols out of the system. It also senses and controls environmental temperature, humidity, and carbon dioxide (CO_2). The Vaping Robot (Figure 1B) is responsible for generating fresh EC vapor and emulating vaping topography parameters selected by user (e.g., puff duration, puff volume, inter-puff interval). It is designed to be a modular assembly of two subunits: rear and front. The rear subunit houses the electronics (including a 5-volt power supply and an Arduino UNO microcontroller with an 8-channel shield relay to program the direction of power to the various components of the system), pneumatics, 4 × pinch valves, a vacuum pump and a fan. The pinch valve assembly is arranged so that fluidic paths between the components of the Vaping Robot, Dilution Robot, and Breathing-Emulator can be programmed and fully automated. The front subunit houses EC polydimethylsiloxane (PDMS) sealing gaskets and acrylic EC stands. The EC is placed into the gasket assembly (in front subunit) which is connected to the pneumatics (in rear subunit) of the Vaping Robot. When triggered by the program, the vacuum pump activates the EC by creating a negative pressure in the gasket assembly around the mouthpiece and draws freshly generated aerosols into reservoir contained within the Dilution Robot. The Vaping Robot is also simultaneously able to control the fluidic connections into and out of the Breathing-Emulator. That is, the Breathing-Emulator can draw in through either filtered air or EC aerosols, and exhale through a separate air filter. The Dilution Robot (Figure 1C) contains a programmable dual-syringe actuation assembly which connects to the aerosol reservoir. Like the Vaping Robot, the electrical compartment of the Dilution Robot uses an Arduino UNO microcontroller with an 8-channel shield relay to direct power to its components (7 × pinch valves and a vacuum pump), utilizing the Vaping Robot's power supply. The microcontroller also controls the linear actuator affixed to the syringe assembly. The syringe assembly contains a sample syringe and mixing syringe, and it is arranged such that one retracts while the other compresses to allow for aerosols to mix between the syringes. The vacuum pump is used to purge the reservoir and tubing dead volume of aerosols in the pinch valve assembly during the dilution program in the syringe assembly. The pinch valves are used to direct the fluidic connections between the Dilution Robot components, determining the state of the syringe assembly (sampling, mixing, and diluting with filtered air), purging the reservoir of aerosols, and connecting the reservoir to the Vaping Robot during EC activation and aerosol sampling. After dilution, the aerosols are sampled from the reservoir via the Breathing-Emulator (Figure 1D). The Breathing-Emulator operates using two air-tight syringes, where the plungers are actuated to mimic physiologically-relevant breathing patterns through their controlled displacement. The Breathing-Emulator is constructed of a rigid acrylic frame that contains supports for a center carriage to move linearly within the frame. The syringe bodies are affixed to the rigid frame while the plungers are fixed to the center carriage. The movement of the center carriage is controlled by a stepper motor and lead screw assembly. The lead screw translates the stepper motor's rotation into linear motion, and its nut is affixed to the center carriage. The assembly also contains two end-stop switches used to trigger the start of the breathing cycle when the syringes compress, and to reset the Breathing-Emulator if the center carriage drifts. The Breathing-Emulator is controlled using an Arduino UNO microcontroller with a printed circuit board (PCB) shield. The PCB contains the components to operate the stepper motor and allows for connections to the electrical components of the Breathing-Emulator (the stepper motor and end-stops), an external 12-volt power supply, and communication lines to the Vaping Robot. Breathing patterns (normal, obstructive, and restrictive) are programmed in the Breathing-Emulator firmware. The PM Sensor (Figure 1E) is a light-scattering-based sensor, sealed in a PDMS gasket and placed in-line between the aerosol reservoir (in Dilution Robot) and Breathing-Emulator so that the real-time particle profile can be detected as the aerosols are pulled through it. The PM Sensor contains

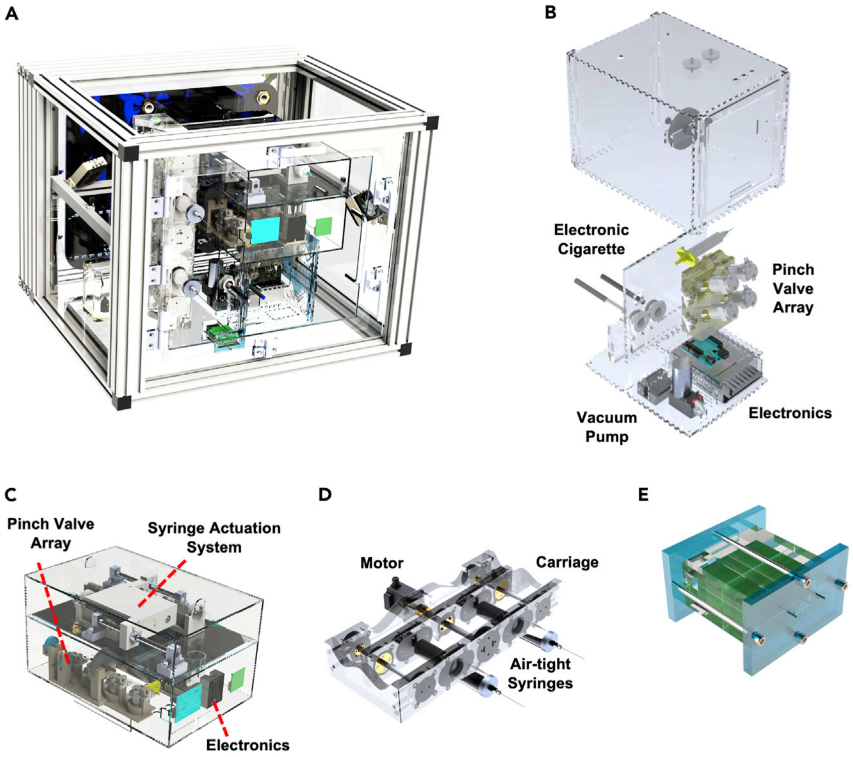


Figure 1. HUMITIPAA system assembly and elements

(A) Human Vaping Mimetic Real-Time Particle Analyzer (HUMITIPAA) assembly within the Inhalation Exposure Chamber.

(B) The Vaping Robot (exploded view) is used to activate the electronic cigarettes (ECs) for a programmable puff duration, drawing aerosols into a reservoir where they can be diluted and sampled. It utilizes an array of pinch valves to direct the fluidic connections between the EC, vacuum pump, aerosol reservoir, PM sensor, and Breathing-Emulator. The electronics contain an on-board power supply and a microcontroller that controls an 8-channel relay and communicates with other elements in the HUMITIPAA system.

(C) The Dilution Robot is used to dilute the aerosols drawn from the Vaping Robot with filtered air. The Dilution Robot uses a syringe assembly affixed to a linear actuator to mix and withdraw a calculated volume of aerosols in the reservoir. Once the aerosols have been withdrawn into the syringe actuation system, a series of dilutions is performed in a sample syringe while the aerosol reservoir is simultaneously purged by a vacuum pump on either the Dilution Robot or Vaping Robot.

(D) The Breathing-Emulator acts as the biomimetic "lungs" of the system, inhaling and exhaling air to imitate the flow dynamics of various human breathing patterns. The Breathing-Emulator is made of a rigid acrylic frame that holds a stepper motor and the barrels of two air-tight syringes. A center carriage holds the syringe plungers and is secured within the rigid frame so that is free to move in the direction of the syringe actuation. The center carriage moves via a lead screw and nut system. A lead screw is affixed to the stepper motor and the nut is secured to the center carriage, allowing the rotation of the stepper motor to be translated into linear movement of the center carriage and, thus, actuation of the syringes. Retraction of the syringes simulates inhalation while compression simulates exhalation.

(E) The PM sensor provides real-time particle count and mass concentration readings for the four particle size fractions (300 nm–1 μm, 1 μm–2.5 μm, 2.5 μm–4 μm, and 4 μm–10 μm). The sensor is pneumatically connected between the aerosol reservoir (connected to the sensor inlet) and the Breathing-Emulator (connected to the sensor outlet), enclosed in a gasket assembly to direct the flow of aerosols across the sensor.

Reproduced with permission from Kaiser et al.[1]

an internal channel between the inlet and outlet that has a laser directed through it. The light from this laser is scattered when a particle passes by it. A photosensitive electrical component is then used to detect this scattered light and output a signal that is eventually translated to a reading of the particle size. To better control the PM reading, prevent interference with ambient PM, and allow the sensor to interface with the system's closed pneumatics, the sensor is placed in an air-tight gasket assembly. The PM sensor interfaces with an Arduino MEGA 2560 microcontroller, and I^2C

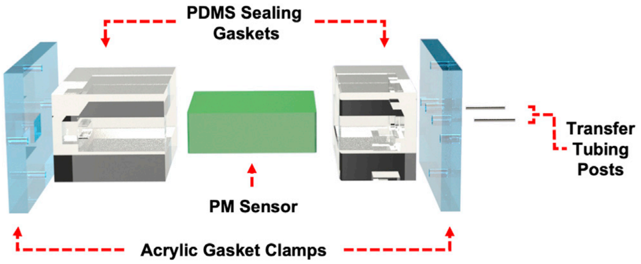


Figure 2. Diagram of PM sensor gasket and assembly
The gasket assembly is used to create a seal around the particulate matter (PM) sensor so that it can be placed in-line between the Vaping Robot and Breathing-Emulator. The acrylic clamps are used to compress the two gasket pieces and provide a better seal; the transfer tubing posts are placed at the sensor's inlet and outlet. Reproduced with permission from Kaiser et al.[1]

communication is used to operate it in this protocol. The raw data generated using HUMITIPAA is processed using a custom-developed program run in MATLAB. More information on the functionality and firmware of these elements is detailed under materials and equipment.

The protocol discussed here specifically details the preparations required to run experiments, and the steps to connect and operate its elements, collect experimental data, and process and analyze this data. For more information regarding the design and construction of each element of the HUMITIPAA system, please refer to Kaiser et al. (2021).[1] We would like to also highlight the ability of our platform for integration with Lung-on-a-Chip systems to evaluation of biological impact of EC aerosols.[6–11]

Preparations prior to operation

Create PDMS gaskets for PM sensor and EC mouthpiece

Timing: 1 day

The PM sensor needs to be secured within a sealed gasket to contain and direct the flow of aerosols through it. A gasket also needs to be developed to create a seal around the EC mouthpiece so that a negative pressure can be applied to activate the EC once secured to the Vaping Robot.

1. 3D print and cure the gasket molds for the PM sensor and EC mouthpiece.
   a. There will be a minimum of 5 pieces total: the two-part assembly for the front half of the sensor gasket, the two-part assembly for the back of the sensor gasket, and at least one mold for the EC mouthpiece.

   Note: This was done using FormLabs' Form 3B+ SLA printer (model number: PKG-F3B-WSVC-MSP-BASIC) with their Black Photopolymer Resin (model number: RS-F2-GPBK-04). If using different types of sensors or ECs, the molds will need to be custom-designed in a 3D modeling software. SOLIDWORKS Standard 2020 was used to design the molds in this protocol.

   Note: STEP and STL files are available upon request.

2. Create the PDMS gasket pieces.
   a. Prepare PDMS using a base elastomer to curing agent ratio of 5:1.
   b. Pour PDMS into molds and cure at 60°C for 3 h.
3. Laser cut and prepare the acrylic gasket clamps for the sensor assembly (Figure 2).
   a. Use 3 mm thick acrylic for the front pieces that contain the inlet and outlet ports, and 6 mm thick clear acrylic for the back plate that contains the cable connection port.
   b. Use an M3 tap to thread the 4 screw holes on the back plate.

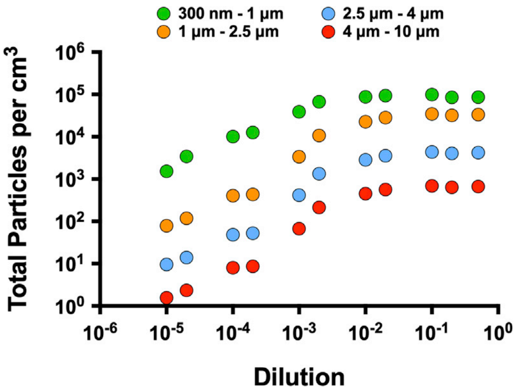


Figure 3. Saturation curve for the Sensirion SPS30 PM sensor used in this experiment

Experimental data collected by sampling EC aerosols at 11 separate dilutions and plotting the peak values of each sample. Each dilution ratio was created and sampled four times. An approximately linear trend can be observed between 10^{-5} and 10^{-3} dilutions, while a saturation plateau can be seen around 10^{-2}.

Note: STEP and DXF files are available upon request.

4. Assemble the PM sensor gasket.
   a. Remove the PDMS gasket from the molds and place around the sensor.
   b. Attach inlet and outlet tubing to the posts.
      i. Place 1/32″ inner-diameter silicon tubing around 18-gauge posts.
      ii. Insert the posts into the front mold and clamp the ends using the acrylic gasket clamps and 4 screws.
   c. Insert the cable and seal all openings using partially cured PDMS.
      i. Seal the cable inlet, the edges where the front and the back gaskets join, and around the transfer tubing posts.

⚠ CRITICAL: When sealing the openings with PDMS, ensure it is viscous enough so that it will not leak into the sensor and damage it.

⚠ CRITICAL: Ensure *all* openings are fully sealed, as leaks in the gasket can greatly reduce the PM count reading.

   d. Cure the full gasket assembly for an additional 2 h at 60°C.

⚠ CRITICAL: The maximum storage temperature for the Sensirion SPS30 PM sensor used in this experiment is 70°C. Ensure this temperature is not exceeded when curing the assembly. If using a different sensor, check its datasheet to verify it will not be damaged at the selected curing temperature.

Test PM sensor to find its saturation curve

Timing: 2 h

The PM sensors that are accessible for the purpose of this experiment have a PM count threshold that is much lower than the PM count of undiluted EC aerosols. Reducing the PM count to below this threshold would require significant dilution which can compromise the consistency of the data through variability in the dilution process as well as add additional time to the sampling series. To avoid this issue, the saturation curve for the PM sensor was determined by sampling and plotting data of various dilutions of EC aerosols (Figure 3). This provides insight into the particle count ranges that the sensor can accurately capture and, thus, appropriate dilution parameters for the experiment.

5. Ensure the PM sensor is running and ready to collect data.
   a. Connect the corresponding pins of the sensor's cable to its 12-volt power supply and microcontroller.
   b. Open the microcontroller's communication port.
      i. Open the Arduino integrated development environment (IDE) and upload the sensor firmware to the connected microcontroller.
      ii. Open the serial port and follow its automatically generated directions to begin logging the PM reading when ready.
6. Sample aerosols from the EC and dilute.
   a. Place the EC into the inlet of its gasket assembly and connect a syringe to the outlet.
   b. Draw vapor from the EC by retracting the syringe.

   Note: The ideal flow rate for drawing EC aerosols with a syringe will result in the EC activating and deactivating instantaneous with the starting and stopping of the syringe retraction. This value typically falls within the range of 15 mL/s to 40 mL/s. If the flow rate is too slow, the EC will not activate; if the flow rate is too fast, there will be an inconsistent delay between the syringe retraction and the EC activation. Additionally, a minimum volume of 15 mL is recommended for the initial draw of vapor.

   c. Disconnect the syringe from the EC gasket assembly,
      i. Attach a syringe filter (0.2 μm or finer) to the output.
      ii. Dilute the EC aerosols with filtered air.

   Note: For the saturation curve shown in Figure 3, 15 mL of vapor was drawn from the EC then depressed to 6 mL. The syringe was then retracted to 60 mL with filtered air to achieve a dilution factor of 10. Each repetition of this exponentially increased this dilution factor.

   ⚠ CRITICAL: To reduce the margin of error, dilutions should be done using a minimum volume that is no less than 10% of the total syringe volume. For example, a dilution factor of 10 could be achieved by mixing 5 mL of vapor with 45 mL of filtered air. However, a dilution factor of 100 should not be achieved by mixing 0.5 mL of vapor with 49.5 mL of filtered air, and instead be done using two iterations of the 10-fold dilution.

7. Record the PM count of the diluted aerosols.
   a. Attach a syringe filter to the PM sensor outlet.
   b. Using an 18-gauge blunt needle with the syringe, insert the needle into the transfer tubing for the PM sensor inlet and inject the aerosols directly into the PM sensor using a consistent flow rate that approximately matches that of the Breathing-Emulator.

   Note: Since the volume displacement of the Breathing-Emulator syringes ranges from 23 mL to 38 mL depending on the selected breathing profile, the recommended minimum final volume of diluted aerosols to pass through the sensor is 30 mL.

   c. After injecting the aerosols, replace the syringe filter onto the syringe outlet, and continue injecting filtered air into the sensor until the PM reading returns to approximately 0 #/cm^3.
8. Repeat **steps 6 and 7** at least 3 times for each dilution factor, then repeat for various dilution factors until a clear saturation curve can be seen.
   a. Once the desired number of repetitions for one dilution factor is completed, stop the PM sensor reading using the serial port.
      i. Save the serial port reading for each dilution factor by copying it to a .txt file.

Note: Processing the data requires the total readings for each dilution factor to be saved separately. For this experiment, there were 11 total .txt files, with each file containing the data for 4 separate aerosol samples.

9. Process the collected data for visualization and analysis.
   a. Refer to "process raw data" in the step-by-step method details for an explanation of the data processing software.
   b. Run the "*Dilution Validation*" MATLAB script.

Note: This script will output two Excel files that summarize the peak particle count and concentration values for each sample ("ParticleDilutionDataCountTable" and "ParticleDilutionDataConcTable"). This data can be transferred to GraphPad Prism for further processing.

Prepare the e-liquid mixtures

Timing: 1 h

Note: This outlines the process for preparing the 50:50 PG:VG base e-liquid and solutions of 1.25%, 2.5%, and 5% (v/v) VEA concentrations. However, VEA may be substituted with other additives, as long as the appropriate measures are taken to thoroughly incorporate the additives into the mixture.

10. Heat the glycerol (VG) to reduce its viscosity and allow for easier measuring and mixing.
    a. For this experiment, this was done by placing the glycerol in the Inhalation Exposure Chamber, heated to 37°C, for 15 min.
11. Measure out equal parts by volume of glycerol and propylene glycol (PG) into a mixing container.

Note: For this experiment, 25 mL of each component were measured using a serological pipette.

Optional: Alternatively, these components may be measured out by weight but the different densities (1.036 g/cm^3 for PG and 1.25 g/cm^3 for VG) must be considered.

12. Place the container into a mixer and mix for 5 min, followed by 2 min of deaeration. A non-vacuum planetary centrifugal mixer (Thinky, model ARE-310) was used for this experiment.

⚠ CRITICAL: Ensure the mixture is properly deaerated prior to measuring it out for the VEA solutions. If using another method to mix the solution, heating the solution may help with deaeration as the air bubbles will dissipate quicker.

13. Measure out 19 mL of the base e-liquid to a separate container and add 1.0 mL of VEA to create a concentration of 5%.
14. Thoroughly mix the solution.
15. Serially dilute the 5% VEA mixture to create concentrations of 2.5% and 1.25% VEA.
    a. Remove 10 mL from the 5% VEA e-liquid solution and dispense into a separate container.
    b. Add 10 mL of the base e-liquid (50:50 PG:VG) to the container and thoroughly mix to create a concentration of 2.5% VEA.
    c. Repeat this method, beginning with the 2.5% VEA solution, to create a concentration of 1.25% VEA.
16. Using a separate syringe and 18-gauge needle for each concentration, dispense ~1 mL of solution from each concentration into separate EC tanks and label accordingly.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Chemicals, peptides, and recombinant proteins		
(+/-)-A-Tocopherol acetate	Sigma-Aldrich	T3376-100G
Propylene glycol	Sigma-Aldrich	W294004-1KG-K
Glycerol	Sigma-Aldrich	W252506-1KG-K
Polydimethylsiloxane (Sylgard 184)	Sigma-Aldrich	761036
Software and algorithms		
MATLAB 2020b	MathWorks	RRID: SCR_001622
Arduino IDE 1.8.19	Arduino	https://www.arduino.cc/en/software
Prism 9	GraphPad	RRID: SCR_002798
Custom-designed data parsing software		Available from lead contact upon request
Other		
Sensirion SPS30 Particulate Matter Sensor for Air Quality Monitoring and Control	sparkfun.com	Cat#: SEN-15103
Mig Vapor Super Battery - automatic	VaporFi.com	N/A
Mig Vapor Clear Fusion Tanks (1.8 Ω)	VaporFi.com	N/A
Acrylic sheet, 3 mm thickness	McMaster-Carr	8560K199
Acrylic sheet, 6 mm thickness	McMaster-Carr	8560K358
Syringe filters, cellulose acetate membrane (surfactant-free), diam. 28 mm, pore size 0.2 μm	Sigma-Aldrich	CLS431219
Transfer tubing, platinum-cured silicone, 1/32″ ID × 3/32″ OD	Cole-Parmer	EW-95802-01
Luer 18ga. blunt needle	VWA	89134-014
Syringe, 3 mL	Thomas Scientific	1219C23
Syringe, 60 mL	Thomas Scientific	1190A94

MATERIALS AND EQUIPMENT

E-liquid solution – 5% VEA

Reagent	Final concentration [%]	Amount
Propylene Glycol	47.5	14.5 mL
Glycerol (Vegetable Glycerin)	47.5	14.5 mL
(+/-)-A-Tocopherol-Acetate	5.0	1.0 mL
Total	N/A	**20 mL**

The e-liquid solution should be stored protected from light and between 2°C and 8°C, with a maximum storage time of four months.

> ***Note:*** Serial dilutions of the 5% VEA e-liquid solution are performed with the base PG:VG mixture to achieve the 2.5% and 1.25% VEA concentrations.

- The Arduino code for the SPS30 PM sensor uses I^2C communication to interface with the sensor. The microcontroller used for this is an Arduino MEGA 2560. The only user-defined parameter is the sampling frequency. Otherwise, the sensor is controlled through the firmware’s interface with the serial port, where user input begins and stops the reading, and can trigger various other functions.
- The Vaping Robot’s Arduino code controls all electrical components of the robot (pinch valves and vacuum pump) using an 8-channel relay that interfaces with its Arduino UNO microcontroller. The main parameters set in the code are the puff duration and breathing profile. The Vaping Robot has four states: puff the EC, inhale filtered air through the sensor, purge the reservoir, and sample the reservoir contents through the PM sensor. It steps sequentially through an array of these states with each inhale of the Breathing-Emulator. The sequence and timing of when each step is

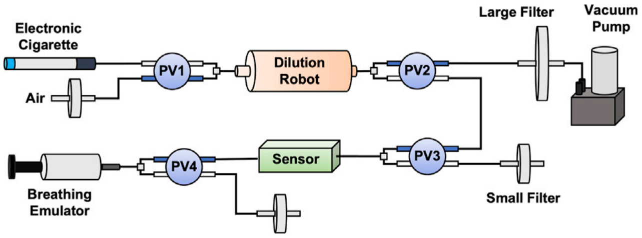


Figure 4. Pneumatic diagram of Vaping Robot and its connections to other elements of the HUMITIPAA

Schematic of the pneumatic components and connections in the Vaping Robot. The labeled blue circles indicate three-way pinch valves, where only one select path is open at any time. The vacuum pump is used to draw vapor from the EC or purge the reservoir with filtered air. The pinch valve arrangement allows for the Breathing-Emulator to either sample vapor or inhale filtered air through the sensor and bypass the sensor when exhaling to allow for unidirectional flow through the sensor.

executed can be programmatically varied. While the vapor is diluting in the Dilution Robot, the Vaping Robot receives a signal from the Dilution Robot to purge the reservoir. The pneumatic layout of the Vaping Robot is detailed in Figure 4.

- The Dilution Robot’s Arduino code calculates the system’s dilution parameters given a user-defined dilution factor, and controls all electrical components (pinch valves, vacuum pump, and the linear actuator that controls the syringe assembly). Similar to the Vaping Robot, the Dilution Robot uses a relay that interfaces with an Arduino UNO. The layout of the Dilution Robot is shown in Figure 5. Once vapor is drawn into the reservoir by the Vaping Robot, it is mixed between the syringes (sample syringe empties its contents, mixing syringe is filled). The sample syringe then fills with the diluted vapor for further dilution (through the filters at pinch valve 1). While the vapor is diluting in the sample syringe, the reservoir and tubing dead volume are being purged. This is done using the purge states of both the Dilution Robot and Vaping Robot. The diluted vapor is then mixed back into the reservoir to be sampled using the Breathing-Emulator.
  - ○ This code also has a ‘‘tube cleaning’’ parameter. This uses the syringes and vacuum to push air through the tubing and connects to the Vaping Robot which can be manually controlled to clear its tubing of residual EC condensate and/or suspended vapor.

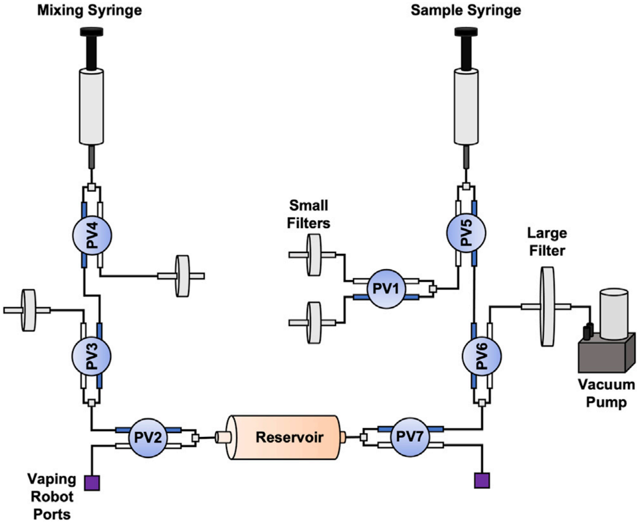


Figure 5. Pneumatic diagram of the Dilution Robot

Schematic of the pneumatic components and connections in the Dilution Robot. The Vaping Robot ports allow for the aerosols drawn from the EC to collect in the Dilution Robot’s reservoir. This layout also allows for the Dilution Robot to mix the reservoir contents between the sample and mixing syringes, sample the reservoir contents, and dilute the contents using the sample syringe while the vacuum pump clears the dead volume of vapor remnants.

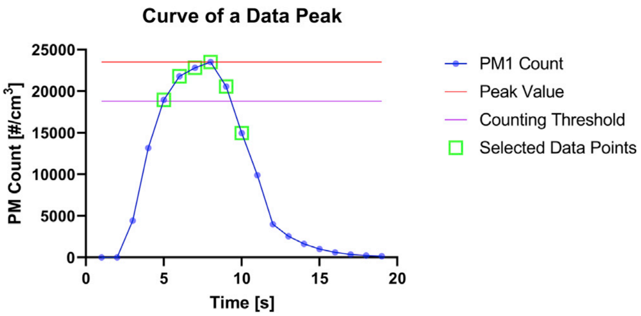


Figure 6. Graph of PM 300 nm–1 μm data points collected over one peak and the criteria for determining which points to select and tabulate

Visual of the MATLAB data processing script's criteria for selecting data points to span an aerosol sample over one breathing interval. Graph shows example data collected over one peak at a sampling rate of approximately 1 Hz. Six points are selected sequentially, beginning with the first data point that passes counting threshold.

- The Breathing-Emulator's Arduino code controls the element's stepper motor, which drives the syringes' volume displacement; it also outputs a signal to the Vaping Robot on each inhale, triggering the Vaping Robot to step through its states. The breathing profile is set in this firmware (normal, obstructive, or restrictive), and must match the breathing profile set in the Vaping Robot firmware, mainly for timing constraints. It also has two end-stops: one to trigger the Breathing-Emulator to begin once the syringes are fully compressed, and the other to reset the Breathing-Emulator if the stepper motor stalls and the center carriage drifts.
- The custom-designed data parsing software converts the text files of the PM sensor's raw data into Excel tables, separating the particle count and mass concentration. Tabulating the data allows for further processing in MATLAB.
- The MATLAB script is designed to search through the tabulated data files and find the location of the peaks (where the aerosols are sampled). It then gathers 6 data points spanning each peak to show the real-time particle count as they are being inhaled and exhaled. This data span begins when the PM count reaches 80% of the peak's PM count as shown in Figure 6.

STEP-BY-STEP METHOD DETAILS

Prepare and connect HUMITIPAA components

Timing: 3 h

Build-up of e-liquid in the tubing can obstruct the pneumatics of the system and ultimately affect the PM reading. Failure to properly connect the components will result in poor data collection and interruptions to the experiment. This step ensures that all systems are clean and functioning properly with each other. This will provide more accurate results and prevent interruptions.

1. Prepare the Inhalation Exposure Chamber.
   a. Open the Inhalation Exposure Chamber's code in the Arduino IDE.
   b. Set the temperature, humidity, and CO_2 parameters in the code and upload it to the chamber's microcontroller.
      i. Set the temperature to 37°C, humidity to 70% (RH%), and CO_2 levels to 5%.

Note: It may take up to 3 h for the Inhalation Exposure Chamber to equilibrate upon start up. Upon reaching this point, the temperature, humidity, and CO_2 levels significantly drop when the chamber is opened. Allow a minimum of 20 min for the chamber to equilibrate after it has been opened.

2. If the system was used previously, replace all pneumatic components of the setup that come into contact with EC aerosols.

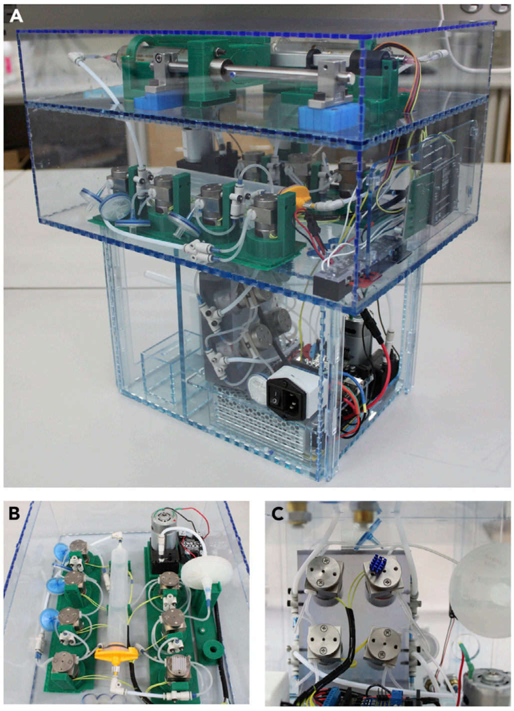


Figure 7. Vaping Robot and Dilution Robot assembly and pneumatic components

(A) The two units of the Dilution Robot assemble such that the syringe assembly sits atop the unit that contains the pneumatic and electrical assemblies. The Dilution Robot sits atop the Vaping Robot with ports to enable the pneumatic and electrical connections between the two robots.

(B) Dilution Robot pinch valve and pneumatic assembly. The two Y-connectors with empty ports connect to the syringe assembly. The reservoir and all connectors and tubing must be cleared of excess e-liquid or replaced between trials.

(C) Vaping Robot pinch valve assembly. The two tubes seen exiting from the top of the robot connect to the Dilution Robot to access the aerosol reservoir. All connectors and tubing need to be purged of excess e-liquid or replaced between trials.

a. Replace the tubing for all systems, as well as the filters, reservoir, and tube connectors in the Vaping Robot and Dilution Robot.

Note: The pneumatic components that must be replaced for the Dilution Robot and Vaping Robot are pictured alongside their final assembly in Figure 7. The pneumatic diagrams for the Vaping Robot and Dilution Robot are seen in Figures 4 and 5.

⚠ CRITICAL: Always make sure systems are powered off and unplugged before handling beyond normal experimental operation.

3. Place and connect robots inside the Inhalation Exposure Chamber.
   a. Connect communication lines between the robots' microcontrollers (Figure 8).
      i. Connect the PM sensor (Figure 9A) and Breathing-Emulator (Figure 9B) to their corresponding microcontrollers.
   b. Connect pneumatics from the Vaping Robot to the PM sensor and Breathing-Emulator and insert the EC in the system (Figure 10).

Note: The image of the full setup in the Inhalation Exposure Chamber is shown in Figure 11.

4. Define the system parameters for the PM sensor, Breathing-Emulator, Dilution Robot, and Vaping Robot.
   a. Define the sampling rate in the PM sensor firmware and test.
      i. Run the sensor for 5 min to ensure that the sampling interval is not too short.

Note: The sampling rate was set to was set to 1 Hz for this experiment. This is the sensor's limit, so setting this value any higher may result in a poor reading. When the sampling rate is set too fast, an overflow (''ovf'') warning will appear in the mass concentration reading.

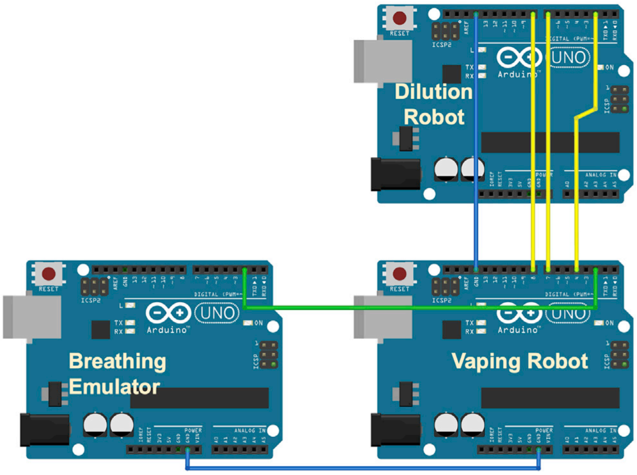


Figure 8. Wiring diagram for communication between microcontrollers

The microcontrollers communicate using binary (high/low) signals. All electrical grounds must be connected within the system to ensure proper communication between microcontrollers (blue lines). The Breathing-Emulator signals the Vaping Robot to step through its states (green line). The Vaping Robot triggers the Dilution Robot to either dilute the EC aerosols or to clear the system of all aerosols. The Dilution Robot sends a signal to the Vaping Robot to utilize its aerosol purging function (yellow lines).

b. Select breathing and dilution profiles in the Vaping Robot firmware.

c. Set the dilution ratio, linear actuator limits, and sample syringe volume in the Dilution Robot's firmware.

   i. Use the "*UNIT_TEST_DILUTION*" Arduino file to find the linear actuator limits.

Note: This file's code allows the user to manually input the retraction length then see how it translates to the syringe assembly. The minimum and maximum actuator limits are the pulse values for which the sample syringe is retracted to 0 mL and 30 mL.

d. Select breathing profile in the Breathing-Emulator software, ensuring it matches the profile selected in the Vaping Robot's firmware.

5. Verify that the parameters and system setup provide an acceptable PM reading using a control 50:50 PG:VG mixture. Refer to the "collect data" step detailed below for the process for running the HUMITIPAA system to collect this PM data.

⚠ CRITICAL: Between samples, the PM sensor particle count and mass concentration readings should drop to 0 #/cm^3 and 0 μg/m^3. Since the pneumatics of the HUMITIPAA system are contained with all access to air being filtered, the PM reading should not be influenced by ambient PM. (Troubleshooting 1).

a. Ensure that the PM reading during the sampling of aerosols is sufficiently below the sensor's saturation threshold that was previously determined. If not, adjust the dilution ratio accordingly (troubleshooting 2).

Note: For this protocol, a dilution factor of 10,000 was used to collect the 300 nm–1 μm data, and a dilution factor of 900 was used to collect the 1 μm–10 μm data.

Note: Testing the dilution parameters using a consistent control PG:VG mixture prior to each experiment validates the Dilution Robot's efficacy and accuracy over time.

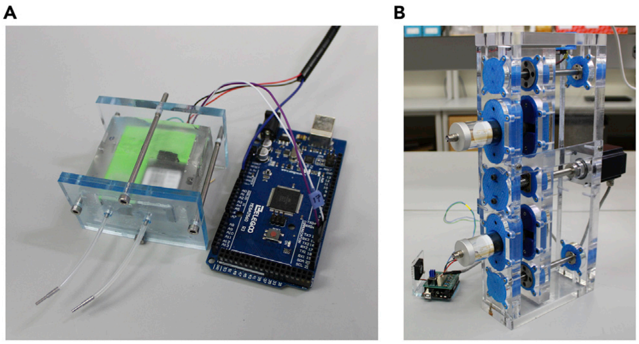


Figure 9. Assembled particulate matter (PM) sensor and Breathing-Emulator with connections to their microcontrollers

(A) The PM Sensor contained within its gasket assembly is connected to its microcontroller and an external power supply (not pictured). Two serial communication wires can be seen connecting to the microcontroller. A ground wire from the power supply can also be seen leading to the microcontroller as it must share a common electrical ground with the sensor to properly communicate. The tubes connected to the sensor inlet and outlet will connect with the Vaping Robot.

(B) The Breathing-Emulator's stepper motor and two end-stop switches are connected to the PCB. The PCB also contains ports for connections to its 12-volt external power supply (not pictured) and for communication wires to the Vaping Robot.

Collect data

⏱ Timing: 4 days

This step outlines the process for collecting data to ensure consistency from the system and prevent cross-contamination between trials.

6. Load the Arduino firmware for each robot and monitor the serial ports.
   a. Open each Arduino IDE with code associated with each robot.
   b. Open the serial ports so that the data on each serial port can be monitored or recorded simultaneously.
7. The Breathing-Emulator drives the rest of the setup, power this on to begin the experiment.
   a. Leave setup running until the desired number of puffs is recorded (troubleshooting 3).
8. After the data is recorded, stop the setup, pause the PM sensor, and charge the EC.
9. Copy the PM sensor serial monitor data into a .txt file. For easier data processing, save with a descriptive file name, followed by the trial number formatted as "T#", and ensure all subsequent files follow the same format (e.g., "0_Percent_VEA_T1.txt", "1.25_Percent_VEA_T1.txt" …) (troubleshooting 4).

⚠ CRITICAL: As the dilution factor increases, the disparity between the submicron particle count (300 nm–1 μm) and the microparticle counts (1 μm–10 μm) greatly increases, resulting in low microparticle counts. This can result in high variability in the microparticle measurements, while the submicron particle count is fairly consistent. This means that two trials at separate dilutions will likely be required to record accurate data for all particle size ranges.

10. Run the tube cleaning cycle for the Dilution Robot and Vaping Robot twice.
    a. Set the "tube cleaning" parameter to 1 in the Dilution Robot code's run settings, as shown in the following snippet of code:

```
>#define TUBE_CLEANING  1
```

b. Set the "debug" parameter to 1 in the Vaping Robot code's run settings, as shown in the following snippet of code:

```
>#define DEBUG    1
```

c. While the Dilution Robot is connected to the Vaping Robot during the first cycle, manually enter the "Smoke" function through the Vaping Robot's serial port communication interface.
d. During the second cycle, manually enter the "Purge" function.
e. Once both cycles are complete, run the "Purge" function again for at least 10 s, or until there is no visible e-liquid passing through the transfer tubing.

11. Replace the EC with the next concentration to be studied and repeat experiment until the data for all desired concentrations has been collected.
12. After all data has been collected, examine it to ensure there are no inconsistencies/missed puffs. If there are, repeat the data collection for that concentration before cleaning the system.

⚠ CRITICAL: Replacing the pneumatic components can slightly affect the PM reading. So, it is imperative that all concentrations are studied within the same trial, before replacing any components. When replacing tubing, make a best effort to align the tubing in the same fashion as it was previously.

13. Once the trial is complete, remove the setup from the Inhalation Exposure Chamber for cleaning.
    a. Clear all pneumatic connectors and tubing of excess e-liquid or replace with new components.

Note: To avoid the introduction of foreign materials to the system, the tubing and connectors can be cleared of e-liquid accumulation using filtered compressed air or a filtered vacuum. If these components dirty from e-liquid build-up over multiple uses, they must be replaced.

Note: Due to accumulation of the e-liquid in the tubing and connectors, it is not advised to run more than ~50 puffs without replacing the pneumatic connectors. So, for an experiment measuring 4 concentrations, it is advised to have 9 puffs per concentration (to give room to repeat a concentration if needed), then manually replace and clean the pneumatic components between trials.

14. Fully charge the EC battery between trials.
15. Reassemble the setup in the Inhalation Exposure Chamber.
16. Wait for the chamber to equilibrate.
17. Repeat this data collection process for as many trials as desired.

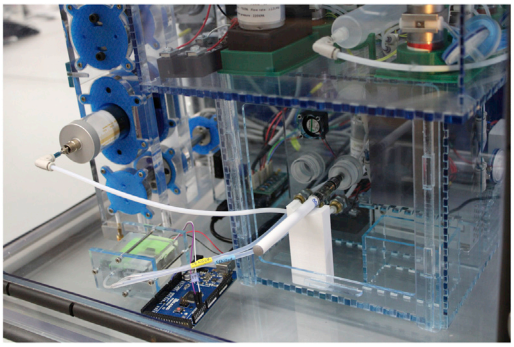


Figure 10. Image of Vaping Robot's pneumatic connections to other elements of the system and electronic cigarette placed in gasket assembly
Pneumatic ports in the Vaping Robot are used to connect to the Breathing-Emulator and PM sensor inlet and outlet. The EC gasket assembly is connected to the Vaping Robot's pneumatic assembly to draw aerosols from the EC.

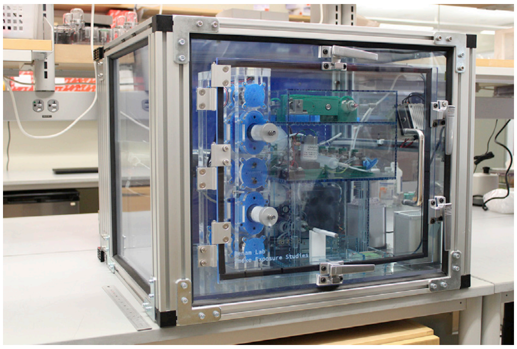


Figure 11. Fully assembled HUMITIPAA setup in Inhalation Exposure Chamber

The final setup of the HUMITIPAA. All pneumatic and electrical connections between and within each component have been established. The USB connections from each components' microcontroller lead to an external computer (not pictured) which is used to control the system. A 12″ ruler is placed next to the Chamber.

Optional: If the microparticle counts within the 1 μm–10 μm size fractions are too low, repeat the experiment using a lower dilution factor. Note that this will likely mean the submicron particle count (300 nm–1 μm) will be beyond the sensor's linearity region, so the data for these particle size ranges would need to be processed separately.

Process raw data

Timing: 10 min (for step 18)

This step converts the raw data into table form for further processing and analysis. It also shows a graph of the collected data over time to verify that there were no immediate issues with the experiment.

18. Use the ''*Raw Sensor Output Parser*'' (RSOP) software to convert the PM sensor's raw text files into Excel files that can be further processed in MATLAB.
    a. Group the raw files according to concentration and process each group separately.
    b. Copy the group of raw files into the folders: ''*…/RawSensorOutputParser/ProcessDilutionLinearity/ActiveDataFiles*'' and ''*…/RawSensorOutputParser/Release/ActiveDataFiles*''.
    c. Run the *ProcessDilutionLinearity* executable file located in the folder: ''*…/RawSensorOutputParser/Release*''.
    d. Locate the output files in the folder: ''*…/RawSensorOutputParser/Release/ActiveDataFiles*'' (Table 1).

Note: This software will output two Excel files for each raw text file. The Excel filenames will be that of the text file followed by 'N' for the particle count data, and 'M' for the mass concentration data. It will also output a ''ProcessedFiles'' text file that lists the names of these Excel files. This text file is necessary to run the MATLAB code in the next step.

    e. Arrange the file names listed in *ProcessedFiles.txt* in ascending order according to trial number.
        i. Based on Table 1, the only text contained within *ProcessedFiles.txt* should be:

```
AllFileNames,
0_Percent_VEA_T1.txtM.csv,
0_Percent_VEA_T1.txtN.csv,
0_Percent_VEA_T2.txtM.csv,
0_Percent_VEA_T2.txtN.csv,
0_Percent_VEA_T3.txtM.csv,
0_Percent_VEA_T3.txtN.csv.
```

Table 1. Example of output files from RSOP for three trials of one concentration

File name	File type
0_Percent_VEA_T1.txt	Text Document
0_Percent_VEA_T1.txtM	Microsoft Excel Comma Separated Values File
0_Percent_VEA_T1.txtN	Microsoft Excel Comma Separated Values File
0_Percent_VEA_T2.txt	Text Document
0_Percent_VEA_T2.txtM	Microsoft Excel Comma Separated Values File
0_Percent_VEA_T2.txtN	Microsoft Excel Comma Separated Values File
0_Percent_VEA_T3.txt	Text Document
0_Percent_VEA_T3.txtM	Microsoft Excel Comma Separated Values File
0_Percent_VEA_T3.txtN	Microsoft Excel Comma Separated Values File
ProcessedFiles	Text Document

The custom-designed data processing software outputs separate Excel files tabulating the particle count and mass concentration data for each raw data file directly copied from the PM sensor readings. An additional text file lists all files processed in the software. These files are manually copied into the MATLAB program's path for further processing and analysis.

19. Run the MATLAB script (*ECDataProcessing.m*) with the RSOP's output files to process the data into tables that show the span of data points along each peak (troubleshooting 5).
    a. Copy all output files from the RSOP to the folder containing the MATLAB code.
    b. In the MATLAB code, enter the name of the files to be processed, the trials to be processed, and the number of peaks per trial (the number of puffs).
        i. Based on Table 1, this code would be:

```
>RawDataFileToBeProcessed = "0_Percent_VEA_";

>TrialsToBeProcessed = ["T1" ``T2'' ``T3''];

>NUMBER_OF_PEAKS = 9;
```

    c. Run the program to process the data.

Note: The processed data will be saved into Excel files named "SpanDataCount" for the PM count data and "SpanDataConc" for the PM concentration data. Graphs of the real-time collected data will appear for each file (Figure 12).

    d. Review the tables of processed data (Table 2) and save under a different file name, or in a separate location, as the Excel files will be written over every time the MATLAB script is run.

⚠ CRITICAL: The data is tabulated based on the order that the files are processed. If the order of the file names in *ProcessedFiles.txt* is incorrect, the trial numbers in *SpanDataCount.xlsx* will be mislabeled.

20. Repeat this process for each concentration.

EXPECTED OUTCOMES

For a 50:50 PG:VG mixture with no additives at a dilution factor of 10,000, we get a submicron particle count (300 nm–1 μm) that generally peaks between 15,000 #/cm^3 and 30,000 #/cm^3. At a dilution factor of 900, we get a microparticle count that peaks between 2,000 #/cm^3 and 6,000 #/cm3 for the 1 μm–2.5 μm size fraction, 350 #/cm^3 and 800 #/cm^3 for the 2.5 μm–4 μm fraction, and 40 #/cm^3 and 120 #/cm^3 for the 4 μm–10 μm fraction. These counts are expected to increase with the VEA additive.

QUANTIFICATION AND STATISTICAL ANALYSIS

The peak span data output from the MATLAB script is processed in GraphPad Prism. It is first tested for normal distribution among the data sets using a Shapiro-Wilk test (α=0.05). If the distributions of

Table 2. Example of the *SpanDataCount.xlsx* output file created by the MATLAB script for three trials and two peaks

	1	1	1	1	2	2	2	2	3	3	3	3
	PM1	PM25	PM4	PM10	PM1	PM25	PM4	PM10	PM1	PM25	PM4	PM10
1	22850.46	113.95	8.49	2.08	25616.08	233.89	23.71	4.45	20944.1	373.63	41.17	7.3
1	22676.78	120.74	8.27	2.23	27192.61	247.28	28.65	4.74	24130.22	438.1	48.4	8.58
1	18267.29	108.44	9.85	2.01	26810.17	282.17	21.86	5.45	22288.89	400.69	44.2	7.84
1	19241.24	123.25	7.52	2.3	25524.08	214.81	14.7	4.14	19278.17	342.04	37.65	6.69
1	15194.32	95.24	4.98	1.77	23890.89	144.43	9.25	2.79	16640.75	290.49	31.89	5.68
1	10453.97	64.07	3.07	1.2	18034.61	90.94	3.43	1.76	14260.78	246.88	27.07	4.83
2	28329.94	415.16	44.4	8.08	25788.04	299.25	37.17	5.75	21205.29	109.06	7.15	1.99
2	30697.94	216.19	23.25	4.21	28572.82	364.34	33.44	7.04	22231.27	98.98	6.71	1.81
2	30751.19	136.89	14.74	2.66	29217.14	323.47	23.43	6.27	19436.64	95.21	6.73	1.76
2	29977.49	84.1	9.05	1.94	30264.24	227.56	14.92	4.41	17767.34	75.16	5.29	1.38
2	29289.27	50.85	5.46	0.99	25385.38	145.57	9.12	2.82	14107	52.14	3.63	0.96
2	27694.8	30.19	3.23	0.59	18175.19	89.25	5.46	1.73	9939.81	33.23	2.28	0.61

The MATLAB script tabulates that data such that the first column labels which peak the data points belong to, and the first and second rows label the trial number and particle fraction of the data below it. This table shows data collected for the submicron particle fraction.

the data sets are shown to be not normal, which is expected, a nonparametric one-way ANOVA is performed (Kruskal-Wallace test, α=0.05) to determine if there is a significant difference among the medians. If a significant difference among the medians is found with the Kruskal-Wallace test, a post-hoc analysis is performed using Dunn's Multiple Comparisons Test, where a significant difference is indicated when $p < 0.05$. If the distributions among the data sets are normal, Brown-Forsythe and Welch ANOVA tests are performed (α=0.05) since it is not assumed that the standard deviations are equal. If a significant difference is determined among the means, a post-hoc analysis is conducted using Dunnett's T3 multiple comparisons test.

LIMITATIONS

Most particulate sensors that can operate with this system have a relatively low saturation point. This means the system requires significant diluting of the vapor to make sure that the sensor is operating within its linearity region. If the values increase as the study continues such that the count begins falling outside of the linearity region, the results cannot be trusted and the experimental setup would need to be restructured to compensate for this.

TROUBLESHOOTING

Problem 1

PM count not returning to zero between samples (step 5).

Potential solution

Check that there are no leaks in the tubing, connectors, or PM sensor gasket assembly that may be introducing unfiltered air into the system. This may also result from the vacuum pump not sufficiently purging the system of EC aerosols between samples. This can occur when there are leaks or obstructions in the system. It can also occur if the vacuum pump or its filter need to be replaced. The pump can be tested individually with a pressure gauge to ensure it is creating a substantial negative pressure (-60 kPa or less).

Problem 2

Dilution Robot not properly diluting (step 5).

Potential solution

First, check that there are no leaks or obstructions in the pneumatics and that all pinch valves are working properly. Other than this, this issue may be due to the sample syringe not fully emptying prior to intaking the aerosols for dilution or mixing, or not completely filling with aerosols during

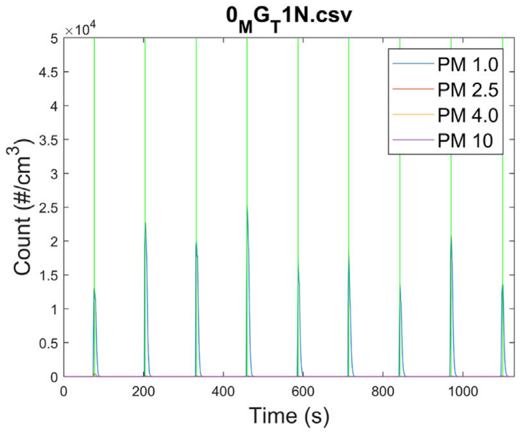


Figure 12. MATLAB graph of all PM data for nine puffs of the control 50:50 PG:VG e-liquid

Plot that is automatically generated by the MATLAB script showing the particle count data with respect to time, where PM1 labels the particles from 300 nm–1 μm, PM2.5 from 1 μm–2.5 μm, PM4 from 2.5 μm–4 μm, and PM10 from 4 μm–10 μm. Note that the readings for PM2.5 – PM10 may not be visible due to the large difference relative to PM1 and scaling. The vertical lines are located at the maximum values for each peak.

these processes. For this, the speed of the linear actuator can be increased on its control board, or the time delay set for the actuation may need to be increased in the Dilution Robot firmware. Also, check the alignment of the syringe actuation assembly, as misalignment can introduce resistance to the actuator movement and slow it down.

Problem 3

HUMITIPAA components desynchronizing when running the system.

Potential solution

First, ensure that all electrical connections are secured between the microcontrollers and that they share a common electrical ground. Loose or missing connections will result in the components receiving or sending faulty signals, causing timing issues or skipped steps by the components. After this, if the issue is occurring in the Dilution Robot specifically (i.e., the dilution cycle does not finish before sampling), the issue can be resolved in the firmware. The time delay set for the linear actuator in the Dilution Robot code can be decreased if the linear actuator speed is sufficient to compensate for it. If not, the Vaping Robot firmware can be modified by adding more states between the EC activation and aerosol sampling states to give additional time for the dilution cycle to finish.

Problem 4

Unexpectedly high or low PM readings (step 7).

Potential solution

Check that there are no clogs in the tubing or connectors. Airflow obstruction and/or leaks in any path will affect the reading. If the reading is too low, there may be an obstruction and/or leak in the path to inhale the vapor, the path to the sensor, or the path from the mixing syringe. If the reading is too high, the vapor may not be diluting properly or the reservoir may not be fully evacuating during dilution. This can be due to an airflow obstruction along the Vaping Robot's purge path or the Dilution Robot's evacuation path, it can also be due to mechanical issues with the actuation of syringes. Also verify that the pinch valves are all working by manually checking each one.

Problem 5

MATLAB script is not running (step 19).

Potential solution

The error message often explains the issue but is sometimes unclear. One common cause for this is the sampling rate for the PM sensor being set too high. If an overflow warning occurs, it will not be caught in the data parsing software and will be recorded as ''ovf'' when tabulated. The MATLAB code will not be able to read this and give a warning saying that the matrix dimensions of the files do not agree. Another cause for this warning is if the PM recording was interrupted before completing a sample. This will result in the mass concentration reading not being fully documented after the particle count reading and, therefore, the length of the two readings not matching.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Kambez H. Benam (benamk@pitt.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The data that support the findings of this study, the code for analysis and any additional information are available from the corresponding author upon reasonable request.

ACKNOWLEDGMENTS

This work was supported by the Division of Pulmonary, Allergy and Critical Care Medicine at the University of Pittsburgh, United States, and the Gordon and Betty Moore Foundation.

AUTHOR CONTRIBUTIONS

K.H.B. and R.F.B. designed the experiments and manuscript layout; R.F.B. prepared the manuscript; K.H.B. and A.J.K. critically revised the manuscript.

DECLARATION OF INTERESTS

K.H.B. is the founder of and holds equity in Pneumax, LLC. K.H.B. and A.J.K. are co-inventors on patent application(s) related to technologies presented here.

REFERENCES

1. Kaiser, A.J., Salem, C., Alvarenga, B.J., Pagliaro, A., Smith, K.P., Valerio, L.G., Jr., and Benam, K.H. (2021). A robotic system for real-time analysis of inhaled submicron and microparticles. iScience *24*, 103091. https://doi.org/10.1016/j.isci.2021.103091.

2. Cullen, K.A., Gentzke, A.S., Sawdey, M.D., Chang, J.T., Anic, G.M., Wang, T.W., Creamer, M.R., Jamal, A., Ambrose, B.K., and King, B.A. (2019). E-cigarette use among youth in the United States, 2019. JAMA *322*, 2095–2103. https://doi.org/10.1001/jama.2019.18387.

3. Romijnders, K.A.G.J., van Osch, L., de Vries, H., and Talhout, R. (2018). Perceptions and reasons regarding E-cigarette use among users and non-users: a narrative literature review. Int. J. Environ. Res. Public Health *15*, E1190. https://doi.org/10.3390/ijerph15061190.

4. Marques, P., Piqueras, L., and Sanz, M.J. (2021). An updated overview of E-cigarette impact on human health. Respir. Res. *22*, 151. https://doi.org/10.1186/s12931-021-01737-5.

5. Valavanidis, A., Fiotakis, K., and Vlachogianni, T. (2008). Airborne particulate matter and human health: toxicological assessment and importance of size and composition of particles for oxidative damage and carcinogenic mechanisms. J. Environ. Sci. Health C Environ. Carcinog. Ecotoxicol. Rev. *26*, 339–362. https://doi.org/10.1080/10590500802494538.

6. Benam, K.H., Dauth, S., Hassell, B., Herland, A., Jain, A., Jang, K.J., Karalis, K., Kim, H.J., MacQueen, L., Mahmoodian, R., et al. (2015). Engineered in vitro disease models. Annu. Rev. Pathol. *10*, 195–262. https://doi.org/10.1146/annurev-pathol-012414-040418.

7. Benam, K.H., Novak, R., Nawroth, J., Hirano-Kobayashi, M., Ferrante, T.C., Choe, Y., Prantil-Baun, R., Weaver, J.C., Bahinski, A., Parker, K.K., and Ingber, D.E. (2016). Matched-comparative modeling of normal and diseased human airway responses using a microengineered breathing lung chip. Cell Syst. *3*, 456–466.e4. https://doi.org/10.1016/j.cels.2016.10.003.

8. Benam, K.H., Villenave, R., Lucchesi, C., Varone, A., Hubeau, C., Lee, H.H., Alves, S.E., Salmon, M., Ferrante, T.C., Weaver, J.C., et al. (2016). Small airway-on-a-chip enables analysis of human lung inflammation and drug responses in vitro. Nat. Methods *13*, 151–157. https://doi.org/10.1038/nmeth.3697.

9. Benam, K.H., Mazur, M., Choe, Y., Ferrante, T.C., Novak, R., and Ingber, D.E. (2017). Human lung small airway-on-a-chip protocol. Methods Mol. Biol. *1612*, 345–365. https://doi.org/10.1007/978-1-4939-7021-6_25.

10. Benam, K.H., Königshoff, M., and Eickelberg, O. (2018). Breaking the in vitro barrier in respiratory medicine. Engineered microphysiological systems for chronic obstructive pulmonary disease and beyond. Am. J. Respir. Crit. Care Med. *197*, 869–875. https://doi.org/10.1164/rccm.201709-1795PP.

11. Niemeyer, B.F., Zhao, P., Tuder, R.M., and Benam, K.H. (2018). Advanced microengineered lung models for translational drug discovery. SLAS Discov. *23*, 777–789. https://doi.org/10.1177/2472555218760217.