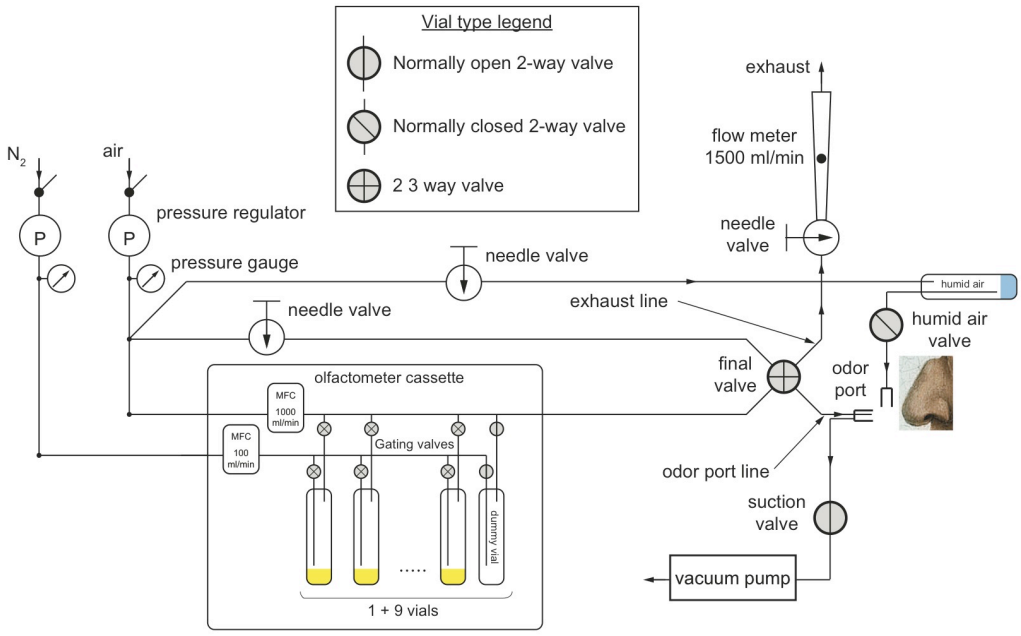
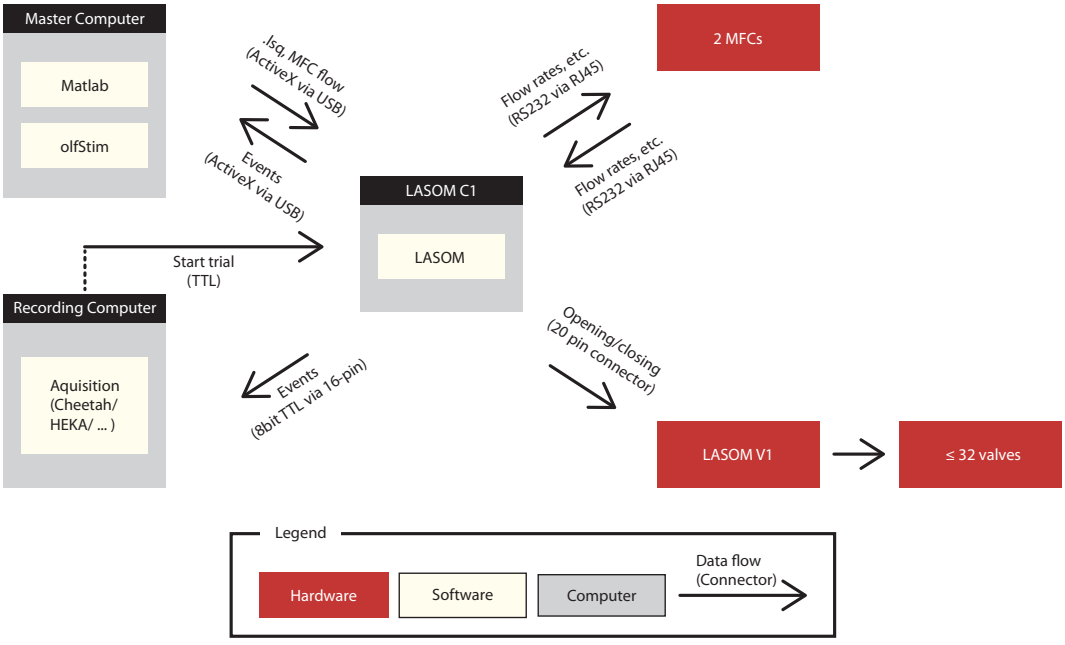
olfStim

Table of Contents





List of hardware

The hardware of the olfactometer has been developed by Dmitry Rinberg. This documentation describes the setup used in the Laurent lab, which was basically a copy of his design with some modifications.
Useful companies for valves, tubing, fittings, mass flow controllers, etc. are:

NResearch, Bronkhorst, Omnifit, Clippard, Aalborg

LASOM

In order to achieve millisecond precision in valve action, sending triggers, reacting to triggers, etc. it is necessary to use some kind of real time computer. This could be a computer with a patched real-time Linux kernel, or some microprocessor. In this design we're using a microprocessor specifically developed for olfactometer control. This control board is called LASOM and was developed by RPMetrix (http://rpmetrix.com/) in collaboration with Dmitry Rinberg.

The advantages of LASOM are, that valves and MFCs can be connected directly to the board and that the sequencer language was developed specifically for the purpose olfactometer control.

LASOM2 comprises two boards, the LASOM_C1 and LASOM_V1.

LASOM2_C1

Description from RP Metrix:
The LASOM2 Scalable Olfactometer Module is a plug mount module that connects to clean gas sources and delivers an odorized gas flow to an olfaction experiment. The output flow contains a controlled concentration of odorant selected electronically from one of fourteen user supplied odor source vials mounted in the module. The module can support an experiment requiring up to 14 odors as is, or can be plugged into one slot of one or more multi-slot module bins available from your supplier. Up to fifteen modules may be combined to construct a 210-odor system. A single USB connection is used to connect a single or multi-module system to a host computer for automated management of the odor selections.
Two integrated independent mass flow controllers permit precise control of purge and odorized carrier gas flows. Clean gas is supplied to both flow controllers. The non-odorized purge gas flow making up the bulk of the flow during a trial is delivered to the subject between trials. The odorized flow is created by routing clean carrier gas through its flow controller and through a selected odor vial during a trial. This combines with clean purge gas which makes up the bulk of the flow. The net olfactometer output flow rate matches an externally managed background flow.
During a trial, external valves swap the balanced background and odorized flows near the subject.
Between trials, the external valves and odor select valves are deactivated. The clean purge gas clears the olfactometer and flows externally to exhaust. The clean background gas flow is directed to the subject.
The on/off valves used to select odors are easily replaceable by the user if necessary. Wetted components can easily be removed, cleaned with solvent, baked, and reinstalled. The valves are powered by an external DC power supply, which permits one module to support one set of 24 VDC valves.
The controller has a USB 2.0 Type B jack for use with the user supplied host computer. A 5-meter USB 2.0 Type A to Type B cable is supplied with the LASOM. A CDROM supplied with the LASOM contains software for installation in a user supplied PC running Microsoft Windows ® 2000, Windows ® XP, or a compatible operating system. USB driver and user application programs are included.
Additional slave LASOM2 modules may be attached to the master (USB connected) module via the integrated CAN BUS connectors and standard CAT5-type 8-wire RJ-

45 terminated cables. The cable carries power and data to the other modules, so that typically only one 24VDC power module is needed. An external power supply connected to J13, and enabled via jumper JP1, is diode mixed to the CAN 24V bus circuit. Each control board routes power from this circuit through its on-board in rush current limiter to power the rest of the module.

LASOM2_V1

Description from RP Metrix
The LASOM2.V1 Valve Driver Module converts logic level control signals into valve excitation signals for up to 32 valves using 24 VDC drive coils. A LASOM2.Cx Controller Module prepares the control signals and directs the valve driver module to energize the valves.
The core logic components on the Valve Driver Module comprise a daisy chain of four 74HCT594 8-bit shift registers. The Controller Module shifts a 32-bit word of excitation signals into the register chain, and then copies the result into the 74HCT594 output registers.
Each of the 32 register output logic signals drives a monitor LED (D401..D432) and energizes a high-side 24VDC valve driver. The drivers are internally protected from short circuits and thermal overload. The per-valve current capacity of 500 mA is far above the typical 10 to 20 mA requirement of odor selection valve coils.
The LASOM2.V1 Driver Module is intended to support an olfactometer servicing 10 to 14 odor vials, with 2 valve circuits used per vial. One extra vial is typically operated with normally open valves to support a background gas flow comparable to the flow occurring when an actual odor is selected. This implies that up to 15 vials and 30 valve circuits are committed to odor selection, with 2 valve circuits unassigned. Controller Module firmware manages these conventions for valve circuit activation – the Valve Driver itself merely carries out instructions to energize an arbitrary combination of 32 valve circuits.
The previous 32-bit control word is normally looped back to the Controller via resistor R40. The J40 expansion port may be used to extend the valve driver to additional valves. (Appropriate controller firmware modifications are also required.)
The on board 1-Wire Identifier chip electronically serializes the Driver Module so that the set of odor vials coupled to the board can be tracked in managed between experiments.

External supplies

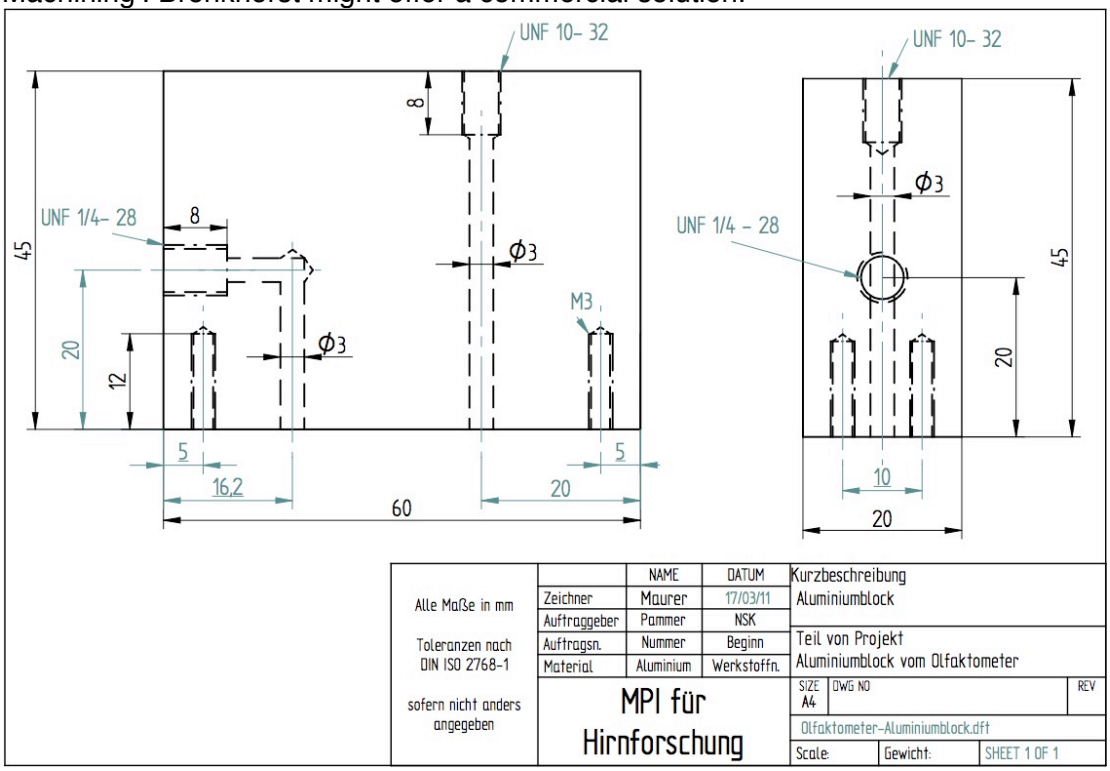
The olfactometer must be supplied with two gas sources. One pressurized air and the other pressurized Nitrogen. Both supplies should deliver at least 2 bar and will be connected to the mass flow controllers.

Mass Flow Controllers

Mass flow controllers are used to regulate the flow rate of the gas, which will carry the odor after passing through the olfactometer. olfStim currently only supports mass flow controllers (MFC), which can be directly controlled by LASOM control board. These include Bronkhorst-type mass flow controllers (MFC), using the digital RS232 interfaces. Tested were the Bronkhorst MFCs IQ-Flow. See parts list for details.
Per olfactometer cassette you need two mass flow controllers, one for regulating the flow of the air stream, the other one for the N2 stream. This means with 2 MFCs you can present up to 14 odors. If you want to extend your olfactometer to be able to

present more odorants, and mix them, you need to add new olfactometer cassettes (and as said 2 MFCs per cassette).
The mass flow controllers are calibrated for a defined gas pressure as input. We suggest to have the MFCs calibrated to accept 2 bar as input pressure.

The Bronkhorst IQ-Flow MFCs don't have fittings to connect tubes. We therefore machined one aluminium block per MFC, which allows tube fittings to be connected. For the design, see the pdf in 'olfStim/Documentation/Hardware Parts/Custom Machining'. Bronkhorst might offer a commercial solution.



Valves

In order to avoid cross contamination of odors presented subsequently, it is important that all wetted parts of the valves are made from inert material such as Teflon. All the valves we use come from NResearch (www.nresearch.com), because of high quality and inertness. LASOM supports valves which are controlled with a 24V signal.

Tubes

As with the valves the tubes need to be from inert material. We use Teflon tubes. Another parameter that helps to minimize cross contamination is the diameter of the tubes. The thinner the tubes the faster the gas flow at a given flow rate. This means with thin tubes the cleaning between trials is a lot more efficient. We therefore suggest using Teflon tubing with an outer diameter of 1/16" and inner diameter of 1/32" for all tubing which comes in contact with odor-carrying air (ie all tubing behind the olfactometer).

Order List

Valves, MFCs

Description	Name	Part	Vendor	Quantity	Unit

		number			price
Mass flow controller for Air	MFC (1.5l/min Air, 2 bar, 20ºC N.C. Control Valve)	IQFD-200C-AAD-00-V-A	Bronkhorst	1	1000 €
Mass flow controller for Nitrogen	MFC (100 ml/min Nitrogen, 2bar, 20ºC N.C. Control Valve)	IQFD-200C-AAD-00-V-A	Bronkhorst	1	1000 €
Valves gating the odorant vials and the dummy vial	Valve manifold (5NC valves), 24V	161T102HH2	NResearch	2 or 4 (depending on whether you want to load 9 or 14 odorants into your olfactometer)	323 $
Valves gating the odorant vials	Valve manifold (4NC valves, 1NO valve), 24V	161T102HH	NResearch	2	323 $
Final valve, gating the odor delivery to the subject	Dual 3-way isolation valve, 4 ports, 24V	SH360T042	NResearch	1	162,69$
Optional. For the “humid air valve” or similar.	Standard 2-way normally closed valve, 2mm orifice, teflon-teflon, 24V	360T012	NResearch	1	112,53$
Optional. For the “suction valve”.	Standard 2-way Normally Open, 3mm orifice, teflon-teflon, 24V	360T022	NResearch1	1	110,93$
Needle valve to control pressure	Aalborg Needle Valve PTFE 1/8NPT HighRes	W728AC	Aalborg (via Novodirect)	2	164,00$
Flow meter with needle valve to control flow rate and pressure jumps	Aalborg Flow Meter tube Air 23l/min 15cm	W73344	Aalborg (via Novodirect)	1	95,00

Electrical connectors

Name	Part number	Vendor	Quantity	Unit price

Piping, tubing accessories

Description	Name	Part number	Vendor	Quantity	Unit price
Gas purifying column	Drierite Gas Purifier	27068 or 09-206 (Fisher)	Fisher Scientific	2	245,70 €
Column clamp for 250/300 ml flask	Universal Platform Clamps for 250/300 ml flask	11-676-014	Fisher Scientific	2	37,80 €
Brown glass vials with Teflon lid for odorants	Thermo Scientific Vials with PTFE Septa Amber Borosilicate	12-100-108	Fisher Scientific	1 box (100 pieces)	180 €
Breadboard as base plate			Thorlabs	1	
Teflon tubing large diameter (non odorant wetted)	Teflon Tubing OD 0.105, ID 1/16	TBGM108	NResearch	20 ft	29,80 $
Teflon tubing small diameter (non odorant wetted)	Teflon Tubing OD 1/16, ID 1/32	TBGM109	NResearch	30 ft	44,70 $
Silicone tubing	Silicone Tubing, OD 1/4, ID 1/8	TBGM103	NResearch	10 ft	31,00 $
Teflon fittings for OD 1/16 tubing	Two Piece Fitting Sets 1/4-28	252P109-50	NResearch	1 pack (50 pieces)	161,68 $
Teflon fittings for OD 0.105 tubing	Three Piece Fitting Sets 1/4-28	253P108-50	NResearch	1 pack (50 pieces)	202,24$
Plug for ¼-28 port	Plug ¼-28	FITM128	NResearch	5 pieces	1,56 $
Straight connector with ¼-28 ports	1/4-28 Connectors Straight Teflon	FITM131	NResearch	5 pieces	10,88$
Elbow connector with ¼-28 ports	1/4-28 Connectors Elbow Teflon	FITM132	NResearch	3 pieces	10,88$
Three way T-connector with ¼-28 ports	1/4-28 Connectors Tee Teflon	FITM133	NResearch	5 pieces	13,50$

Barbed fitting	Barb Fitting Zero Dead Volume (1/4-28, 1/16)	FITM331	NResearch	5 pieces	2,42$
Versatile Teflon connectors	Omnifit small 2-way straight connector 1.5 mm	24600	Omnifit (vendor: Novodirect)	2	53,00€
Versatile Teflon connectors	Omnifit small 2-way straight connector 0.8 mm	24400	Omnifit (vendor: Novodirect)	3	60,8€
Versatile Teflon connectors	Omnifit small 2-way L-connector 1.5 mm	24750	Omnifit (vendor: Novodirect)	1	60,50€
Versatile 3 way Teflon T-connectors	Omnifit small 3-way T connector 1.5 mm	24606	Omnifit (vendor: Novodirect)	3	57,70 €
Versatile Teflon connectors	Omnifit large to small connector 1.5 mm diam	24607	Omnifit (vendor: Novodirect)	1	73,10€
Versatile Teflon connectors	Omnifit large 2 way connector 4.5 mm diam	24608	Omnifit (vendor: Novodirect)	1	86,20€
Gasket	Omnifit O-Ring 0.5-4 mm for small OD	24672	Omnifit (vendor: Novodirect)	2	6,30 €
Gasket	Omnifit O-Ring 0.5-4 mm for large OD	24673	Omnifit (vendor: Novodirect)	2	6,30 €
Gasket	Omnifit PTFE cone 3 mm	24641	Omnifit (vendor: Novodirect)	1	14,50€
Gasket	Omnifit	24639	Omnifit	1	14,50€

	PTFE cone 2 mm		(vendor: Novodirect)		
Gasket	Omnifit PTFE cone1/16"	24632	Omnifit (vendor: Novodirect)	1	14,50€

Setting up the hardware

Connecting the valves

Two of the valve manifolds have one valve, which is open in the resting position ("normally open" – NO). LASOM makes assumptions about which valves are connected to which connectors on the LASOM2_V1 board (the board with the connectors for the valves). Therefore always connect the NO valves gating the empty dummy vial to V1 and V2. From V3 to V20 connect the other odor vial gating valves of the manifold. Connect the gating valves from the N2 stream to the even-numbered connectors, and the gating valves from the air stream to the odd-numbered connectors. The two valves gating one vial have to be connected to two consecutive connectors. Using 4 valve manifolds you can present up to 9 odorants, using 6 manifolds you can use 14 different odorants.
Example: the valve gating the air stream for the first vial has to be connected to V3, and the valve gating the N2 stream for the first vial, has to be connected to V4. For the second vial, the gating valve for the air stream goes into V5 and the N2 equivalent into V6.

The other valves (final valve or custom valves you want to control) go into valve connectors V21 to V32. You have to define the mapping of valves handled by the olfStim software to the connectors on the LASOM2_V1 board. For the default valves in olfStim the mapping is as follows:

Valve	**Connector**
Final valve	V31
Sniffing valve (Laurent lab only)	V32
Suction valve	V28
Humidity valve	V27

These definitions are just part of the sequencer building blocks for every action.

For powering the final valve looks something like this:

Gate, slaveIndex, 31;

If you want to change the mapping for the default valves just change the numbers in the parameter definition. 31 in this example means that finalValve is connected to connector 31.
If you want to add new valves and control them through olfStim, read on in the section for developers.

Connecting the Mass Flow Controllers

There are two mass flow controllers, one for regulating the flow of the air stream, the other one for the N2 stream. olfStim currently only supports mass flow controllers (MFC), which can be directly controlled by LASOM control board. These currently include Bronkhorst-type mass flow controllers (MFC), using the digital RS232 interfaces. Tested were the Bronkhorst MFCs IQ-Flow. See parts list for details.

There are two RS232 connectors on the LASOM_C1 board: J21 and J22. The MFC #1 goes into J21 the MFC #2 goes into J22. OlfStim expects MFC #1 to be the air regulating MFC, and MFC #2 to be the N2 regulating MFC.

MFC #	**MFC type**	**Connector**
1	Air	J21
2	N2	J22

Olfactometer - computer connection

Get the latest LASOM package, which includes the driver by connecting to RPMetrix' ftp server (using an ftp client like FileZilla):

ftp.rpmetrix.com
user: support4rpm
password: U32x44rawvb9

Download the LASOM2 zip file with the latest timestamp. Unzip the folder. Open the command line, cd to the "RP Metrix" folder in the downloaded LASOM2 folder and execute the following statement:

regsvr32 LASOMX.ocx

You should receive a success pop-up message.

Note: You should also download a LASOM1 folder from the RPMetrix ftp server as the documentation for the sequencer codes are only contained in this package.

Required testing

Before starting to use the olfactometer with olfStim some initial testing has to be done to make sure, everything works as expected. For this use the LASOM control

software “LoadBoard” which can be downloaded from the RPMetrix ftp server. The olfStim package includes the load board executable in

Go to:

Understanding the sequencer code

The microprocessor of the LASOM_C1 board understands a certain set of commands. For a description of this sequencer language see the “LASOMSequencerOperation.pdf” in the Documentation folder.

Here is a further description of some commands used by olfStim:

Odor,slaveIndex,odorValveIndex	Command will open the two valves gating the vial defined by odorValveIndex and will automatically close the two valves gating the dummy vial in the slave defined by slaveIndex. ATTENTION: As the two valves gating the vial #2 are connected to V3 and V4, the odorValveIndex of the first vial is 2. The reason is because the dummy valves are connected to connectors V1 and V2. Therefore all valve indices are shifted by 1.
Gate,slaveIndex,valveIndex	Command will open the valve connected to the connector defined by valveIndex on the slave defined by slaveIndex. Gate,1,3 will open the valve connected to connect V3 on slave 1.
StartTimer, 2	Starts an internal Timer at timer index 2 (timer indices 2-7 are generally usable, timer indices 0 and 1 are reserved for loops.
WhileAlways SinceTimer, 2, 3 CheckLapse, 3, someTime IfLapse, 3, @lapsedOut EndWhile	Enter an infinite while loop, for every iteration, compute the relative time since the timer was started for timer index 2, write that relative time into timer index 3. Check whether the time in index 3 exceeds the time in ms defined in someTime, store the result. IfLapse checks whether timer index 3 did exceed the defined time value, if so the sequencer jumps out of the while loop to label @lapsedOut.

The commands have to be provided by the master computer running olfStim to the LASOM_C1 board in the form of sequencer files. Sequencer files, also called osq (olfactometer sequencer) files, are txt files with the .osq extension. They typically include the setting of parameters and action commands to execute.

Sequencer Files in olfStim

One of olfStim's core functions is buildTrialOsq.m. It uses the trial information present in smell to construct a sequencer file, which can then be sent to the LASOM_C1 for execution.

There are two important keywords in sequencer files (also osq files) which buildTrial.m uses for parsing the text.

`eof`

This marks the end of the sequencer code and is contained at the end of the core.osq file. buildTrialOsq.m adds the sequencer building blocks at the location of 'eof'.

`bof`

bof marks the step in the sequencer code where time is set to zero. This is where the clock of the sequencer starts to count. Relative to this clock all actions are triggered. The sequencer file sent to LASOM (trial.osq) always contains some parameter definitions at the beginning, and depending on the use sometimes for instance an endless loop which checks for a

For users

Installation

Download or checkout olfStim from the github repository:
https://github.com/lorenzpammer/olfStim

Add the olfStim folder to the Matlab path. Only select the parent folder, as all necessary subfolders will be added to the path once olfStim is being called.

Necessary customization

It was a design goal to make olfStim work out of the box, but there are several customization options, which allow users to adapt olfStim for their particular needs without having to alter code.

Odorant library

All odorants you want to present have to be entered into the odorLibrary, which is just a text file with definitions for each odorant. The purpose is, that all information, which characterizes an odorant is entered once for each odor in use.

You can find the file under the following path: olfStim/odorLibrary/odorLibraryGenerator.m

There are a number of default odorants in the odor library. When you start using olfStim yourself you will either have to update them, for instance because you use odor X with a different purity, or just delete all entries and start from scratch.

Let's say we want to add Cineole to our library of odorants. We'll first add the odor name to the list of odor names:

```
odorNames = {'Benzyl alcohol','Butanone','Camphor','Citral', ...
    'Eugenol', 'Geranyl acetate', 'Hexanal', '1-Hexanol', '2-
Hexanone', 'Isoamyl acetate', 'Vanillin', 'Turtle food', 'Paraffin
oil', 'Mineral oil', 'Ethanol', 'Water', 'Empty', 'Cineole'}; % Cell
array of strings with names of odorants in Library
```

Then we'll add an entry to the list of odors. To do this just copy the template, which you can find at the top of paragraph C in the odorLibraryGenerator.m file, paste it to the end of the file (above the last 'end') and update the input to each field for the odorant you are using.

```
odor = 'Cineole';
for i = 1 : length(odorLibrary)
    if strcmp(odorLibrary(i).odorName, odor) % at the respective odor
write the following properties to the structure:
        odorLibrary(i).iupacName = '1,3,3-trimethyl-2-
oxabicyclo[2,2,2]octane'; % the name of the molecule following IUPAC
convention as a string
        odorLibrary(i).CASNumber = '470-82-6'; % the CAS registry
number identifying the molecule as a string
        odorLibrary(i).producingCompany = 'Sigma'; % the name of the
company that produced the odorant: 'Sigma', 'Roth', etc.
        odorLibrary(i).odorantPurity = 0.99; % purity of the molecule
given as a fraction, usually above 0.95: 0-1
        odorLibrary(i).state = 'liquid'; % string describing whether
the state of the molecule is liquid or solid: 'liquid' or 'solid'
        odorLibrary(i).odorantDilution = 0.1; % volume fraction
(v(odorant)/v(dilutive solution)) of odorant in the vial after
diluting it with water or oil: 0-1
        odorLibrary(i).dilutedIn = 'Paraffin oil'; % in which
solution the odor was diluted: 'Water' 'Paraffin oil' 'Mineral oil'
or [] if no dilution
        odorLibrary(i).concentrationAtPresentation = 0.005; %
concentration as volume fraction (v/v) of saturated headspace, which
is presented to the animal: 0-1
        odorLibrary(i).inflectionPointResponseCurve = []; %
concentration (v/v, of saturated head space) of presented odorant at
which the response curve as measured in olfactory nerve has its
inflection point.
        odorLibrary(i).vaporPressure = [253.3 25]; % vapor pressure
in Pascal @ 25?C. For mixtures use Raoult's law. To convert mmHg into
Pascal multiply by 133.322
```

```
    end
end
```

Configuring valves

Regarding valves several separate properties can be configured.

1. Valve actions
2. Valve connector index
3. Action properties

Valve actions

Actions are loosely defined. Basically an action is just a block of sequencer code, which will then be used by buildTrialOsq.m to construct the sequencer file for the current trial. This can be very simple such as opening or closing one specific valve, but it can just as well be a complex sequence of commands for the sequencer. If you want to add a new valve action create a text file in the olfStim/osq directory. Change the filename (no spaces!) to the name of the new action and add the .osq extension. The file must contain the sequencer code for that action.

Valve connector index

Valves are connected to the connectors of the LASOM2_V1 board. If you add a new valve you have to tell the sequencer to which connector the new valve is connected. This information has to be provided in the block of sequencer code for the specific valve action (see the above paragraph).

Say you want to add a new action, which opens the 'xyz' valve connected to the connector V30 on the first slave LASOM2_V1 board. You have to add a text file with a name like openXyz.osq to the olfStim/osq/ directory. This text file should contain something like:

```
Gate, 1, 31; Opens the xyz valve
```

The Gate command tells the sequencer to power the valve connected to the valve connector #31 (second argument) on slave #1 (first argument). See the section Understanding sequencer code.

Action properties

All other customization and configuration for valve actions can be set in the main configuration file (olfStim/configuration/olfStimConfiguration.m) under the 'Valves' section. Every action you want to be able to access, alter and use from the gui, has to be defined by one entry in each of the various configuration variables (actionNames, values, used, etc.) and the order of entries in each variable has to be the same. The purpose of each variable is described in detail in the configuration file.

One can rename, add or delete valve actions, by altering, adding or deleting the corresponding entries in the variable. The list of action names in the 'actionNames' variable have to match the filename (without the .osq) of the valve action, as described above.

As an example not all users will need a "sniffing valve". We would therefore delete 'openSniffingValve' and 'closeSniffingValve' from the actionNames variable. Also in

all other variables of the Valves section in the configuration file, one would have to remove the entry

Configuring I/O actions

The configuration and customization of I/O actions follows the same logic as configuring valve actions.
Add snippets of sequencer code (osq files) for each I/O action to the olfStim/osq/ioCodes/ folder in the same way you add valve actions.
The default parameters for the actions can be set in the olfStimConfiguration.m file under the "I/O" section.

Configuring the look of the odorSelectionGui

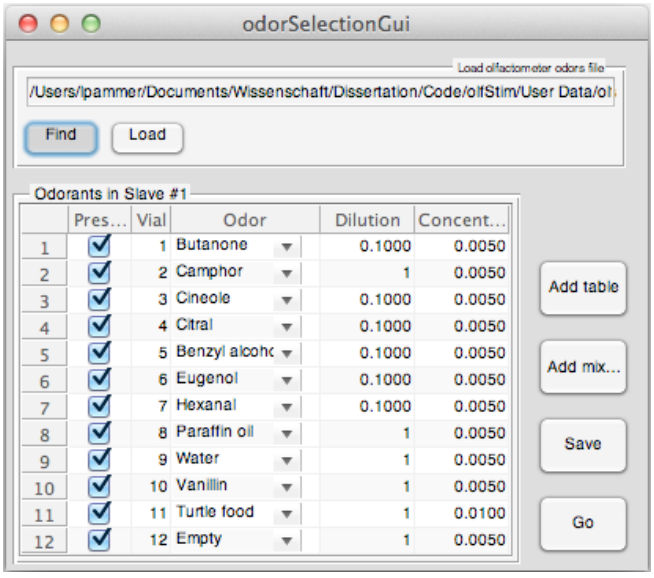
In the olfStimConfiguration.m file under the rubric "odorSelectionGui" some default parameters for the odorant vials setup can be entered. You'll find a detailed description of all parameters in the configuration file. For instance the number of vials displayed in the odorantSelectionGui can be set. This might be useful if you want to load more than the current default of 9 odor vials into the olfactometer.

Running olfStim from the gui

In the Matlab command line type

```
initOlfStim
```

This will result in the odorSelectionGui to open.



For every slave, choose the odorant in every connected vial. In the first column check whether you want to use this odor in the session. In the third column choose the odorant contained in the vial from the dropdown menu – the second column specifies the vial number.

Make sure that the dilution of the odorant in the vial (eg you diluted Cineole in Paraffin oil) is specified correctly in the 4th column. For the standardStim protocol it is necessary to set the concentration at presentation (v/v) in the 5th column.

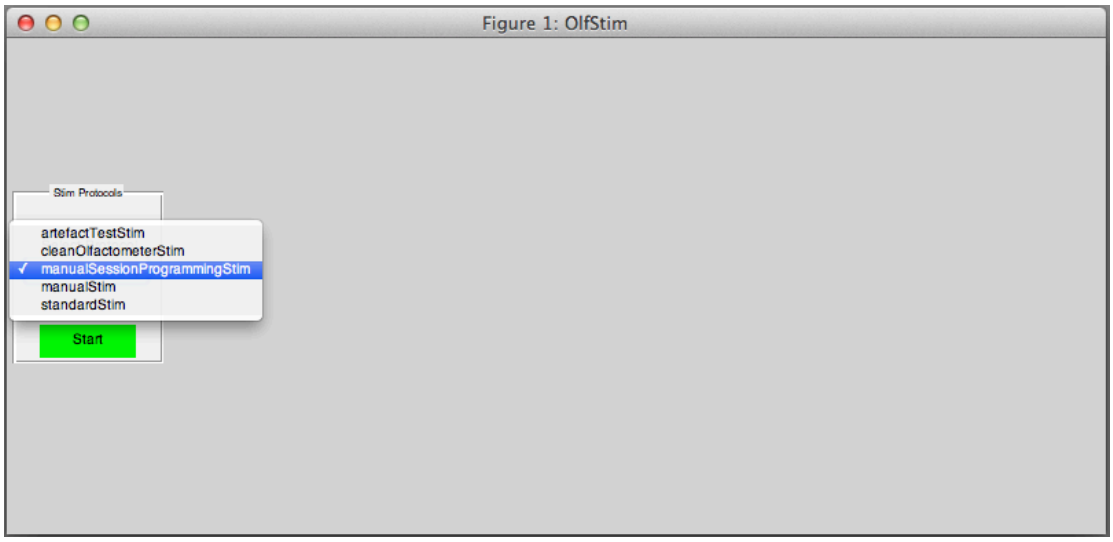
In order to avoid having to enter the loaded odorants every time, you can save the settings to a file by clicking the “Save” button. Previously saved settings can be loaded by specifying the location of the file with the “Find” button and loading it with the “Load” button.

NOTE: odor mixtures cannot yet be used.

Once the Gui contains the information which odor is contained in which vial click the “Go” button.

You can change the default look of the odorSelectionGui by changing parameters in the olfStimConfiguration.m file section “Odorants”

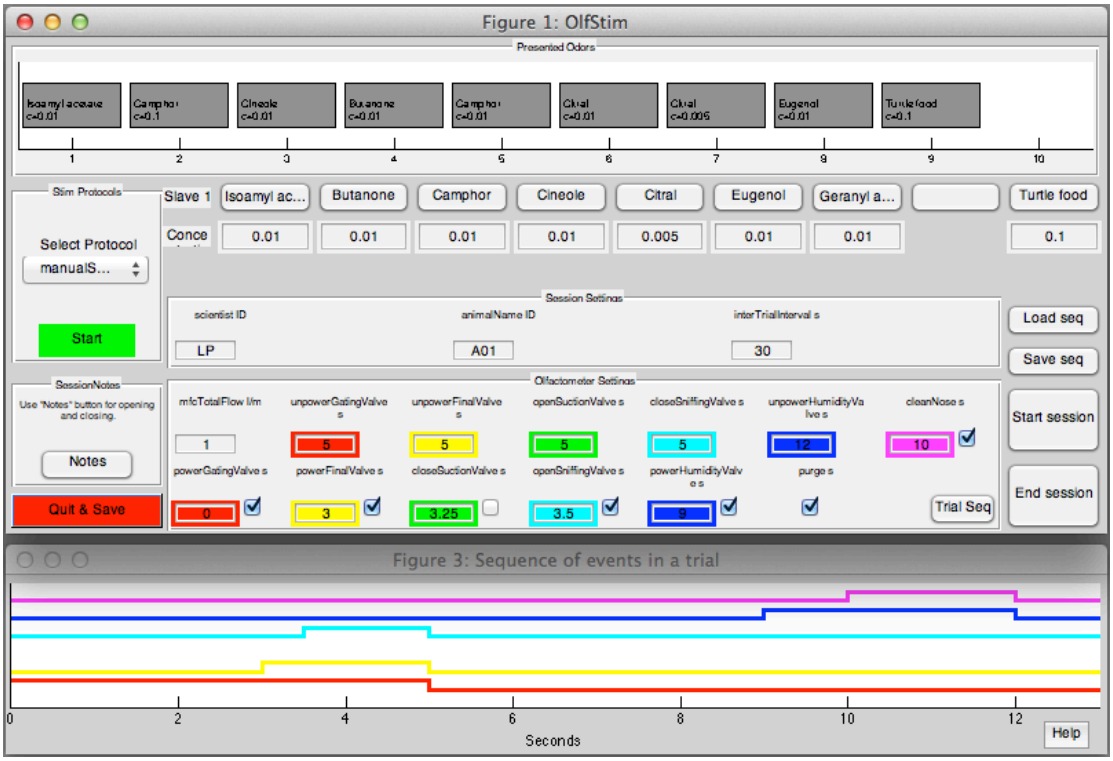
The main gui will now open and you have to choose which stimulation protocol you want to use. See details about the protocols below. Then click the “Start” button.



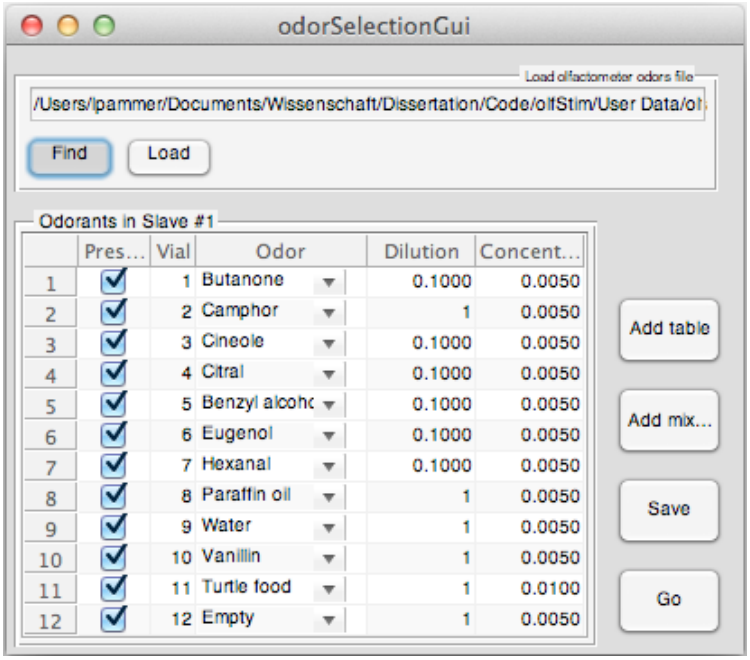
This results in the gui being populated by all features of the chosen protocol. You are now ready to start presenting odors! For the explanation of the protocols’ features read the protocols section in this document.

Features

In this section olfStim’s general features are documented.



Odorant Selection



The gui for odor selection is opened automatically, when starting olfStim in Gui mode with initOlfStim. However it can also be opened by itself, by calling

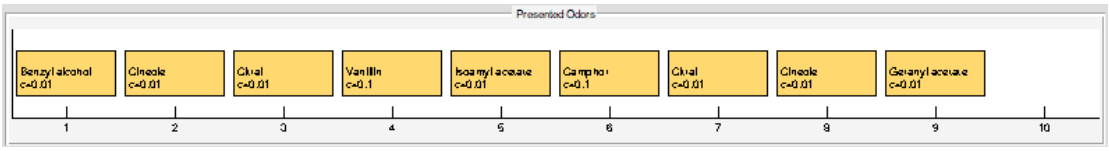
```
odorSelectionGui
```

For every slave, choose the odorant in every connected vial. In the first column check whether you want to use this odor in the session. In the third column choose the

odorant contained in the vial from the dropdown menu – the second column specifies the vial number.
See that the dilution of the odorant in the vial (eg you diluted Cineole in Paraffin oil) is specified correctly in the 4th column. For the standardStim protocol it is necessary to set the concentration at presentation (v/v) in the 5th column.

In order to avoid entering the loaded odorants every time, you can save the settings to a file by clicking the “Save” button. Previously saved settings can be loaded by specifying the location of the file with the “Find” button and loading it with the “Load” button.

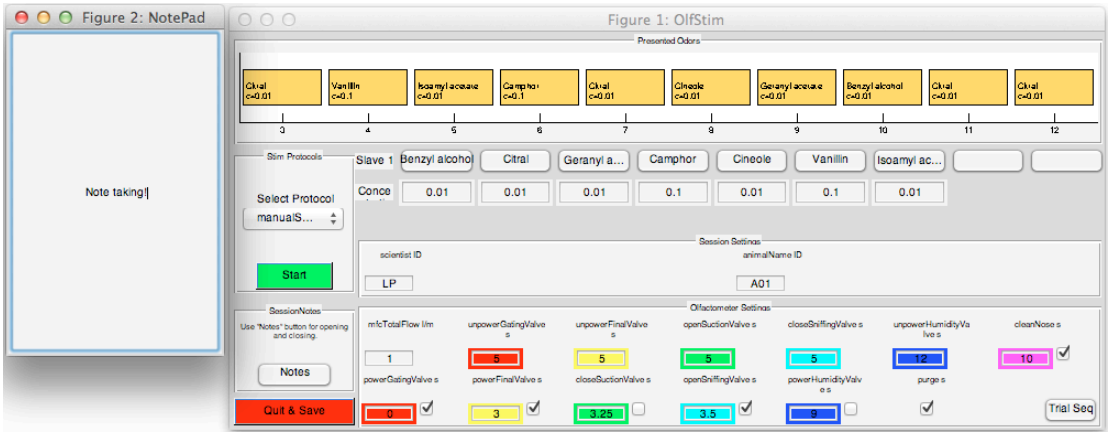
Progress panel



The progress panel is at the top of the GUI and shows the history of presented odors, identifying each trial with the odor name and the concentration.

Notes

Clicking on the “Notes” button on the left lower part of the GUI will open a new figure – the note pad - for note taking. In this figure you can write your notes for the current trial. When the trial is triggered the notes are extracted and associated with the trial in the output data structure. At the same time the figure will be cleaned of the old notes. Use the “Notes” button both to open and close the note pad.

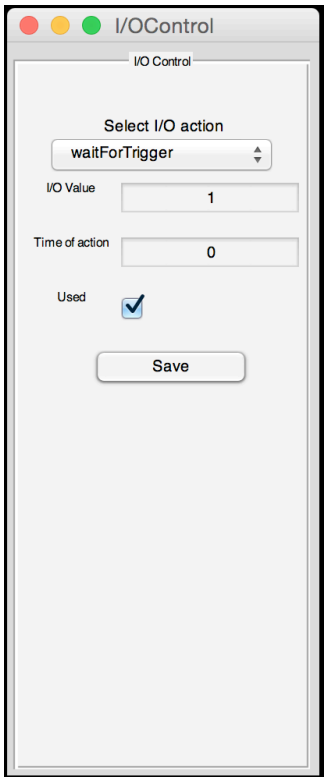


Olfactometer Settings

In the olfactometerSettings panel the times of valves opening & closing can be entered manually for each trial. Also multiple opening and closing times are allowed. This can be done by entering two values in square brackets (matlab vector notation). For instance entering [3 6] in the final valve and [5 8] in the close final valve field will result in the final valve switching on at 3 s, switching off at 5s, switching on again at 6 s and off again at 8 s.

Note however that at the end of the trial the resting state has to be reestablished. Which means you have to close/unpower every valve you open/power.

I/O and execution flow



For many applications it is necessary to trigger or record precise time stamps of olfactometer actions with external devices. olfStim allows a simple form of digital input/output triggers. To add trigger actions, one has to do two things. Save a block of sequencer code (language for the olfactometer controller) to the ioCodes folder and create an entry in olfstim's configuration file under the header I/O. Once this procedure has been followed, the actions will be available in the I/O gui and for scripting and can be placed at one or multiple time points within the sequence of actions.

smell Structure

The smell structure serves two purposes. First the smell structure contains all relevant information for every trial of olfactory stimulus presentation. These instructions are provided by the user through the gui interface and are then used by olfStim's functions for the execution of each trial. Additionally during the execution of each trial some trial-related data is stored in smell such as the actual flow rate of the mass flow controllers, user's notes, etc. Thus after finishing one session of odor presentation, the smell structure contains all relevant data for every trial and will typically be saved as a mat file by the user. Additionally it is saved automatically after every trial to ensure against data loss at crashes. The final smell structure is useful for later analyses of the experiment.

Data contained in smell are: Information associated with the presented odorant in the odor library (see below), time of presentation, target odor concentration, target flow rates of the mass flow controllers, actual mass flow controller flow rates, user's notes, valves opening and closing times, inter-trial intervals, names of experimenter and animal, command file for sequencer, digital triggers.

Protocols

Cleaning

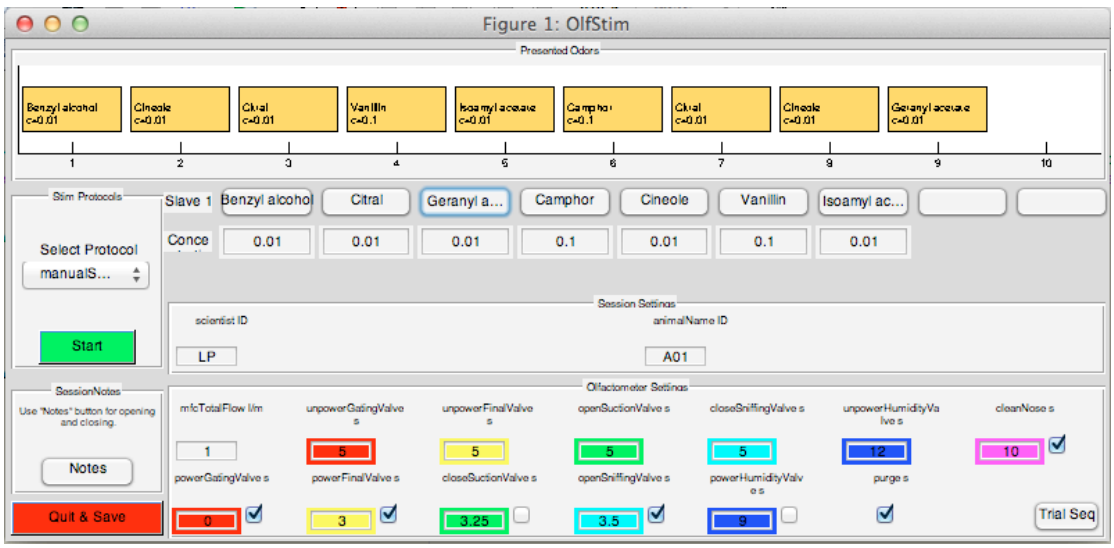
Start by filling all vials with Ethanol. Then start the cleaning protocol and for one vial after the other olfStim will open the gating valve thus rinsing all lines with odor-contact with ethanol and drying with air flow.

manualStim

The manualStim stimulation protocol is simple and a good starting point to test the olfactometer but also for exploratory experiments. It allows you to set the concentration of the odor presentation, control the total flow rate and the opening and closing times of all relevant valves. Trials are triggered by manually clicking a button in the gui.

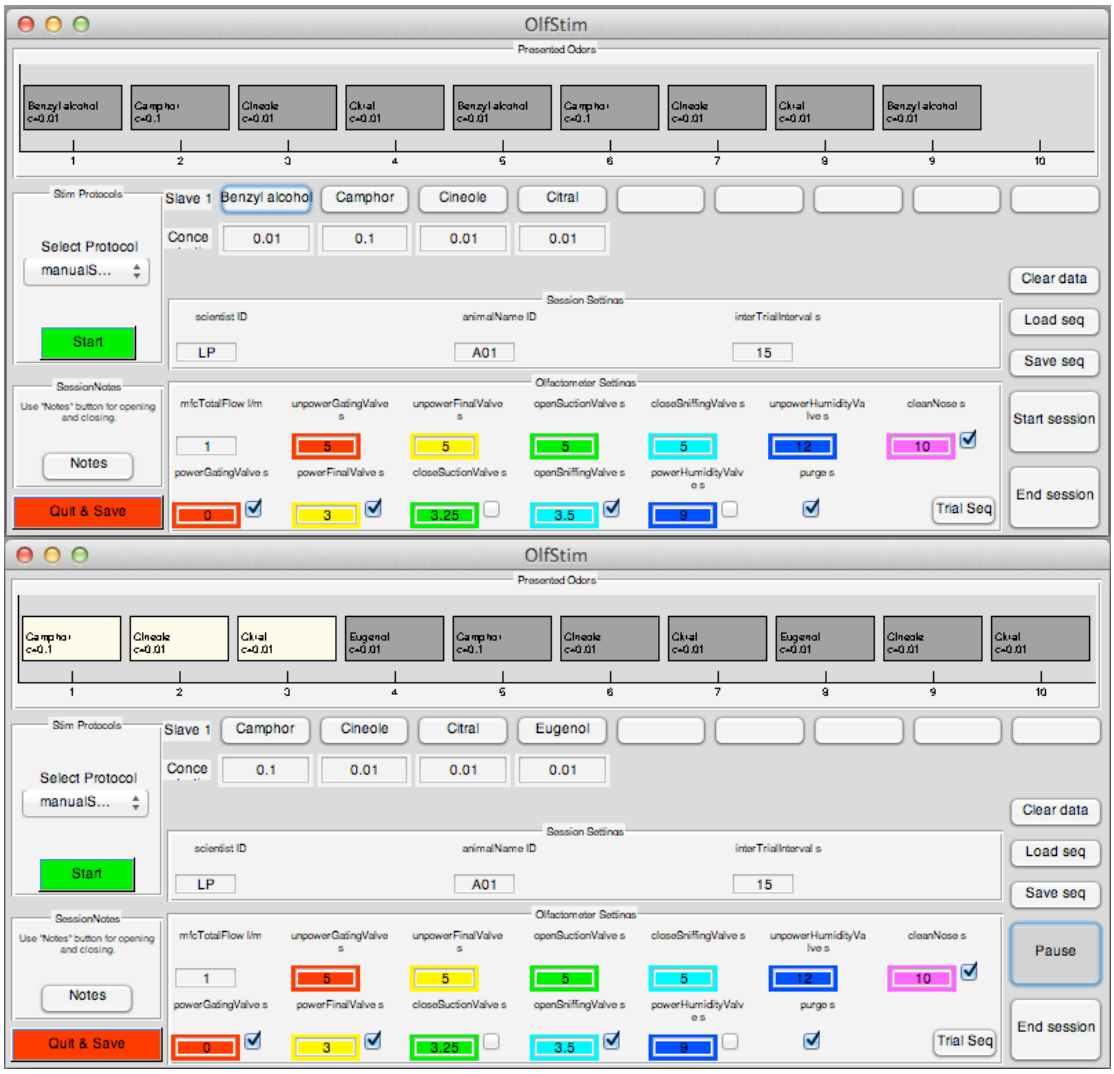
Every odor you have defined in the odorSelectionGui will show up in form of a button in the first row underneath the progress panel. Vial numbers ascend from left to right. Beneath every odor button a concentration field is present. In the concentration field you can set the concentration (= fraction of saturated headspace) at which the odor will be presented. If you have set the dilution of the odorant in the vial correctly in the Odorant Library or the odorSelectionGui, olfStim will automatically adjust the flow rates accordingly.
If you enter concentration values, which cannot possibly be reached given the specification of the MFCs and your chosen total flow you will get a warning via pop up. For example, if you want to present an undiluted odor at a flow rate of 1 liter per minute and at a concentration of 0.3 you will get a warning, because the Nitrogen MFC has a maximum flow rate of 100 ml per minute. Another second error check will warn you if you enter a concentration, which would result in MFC flow rates so low that they cannot be reliably achieved.

Once you have set the concentration and the other relevant parameters for the trial you can press the odor button and the trial will start. Executed trials are shown in the progress panel.

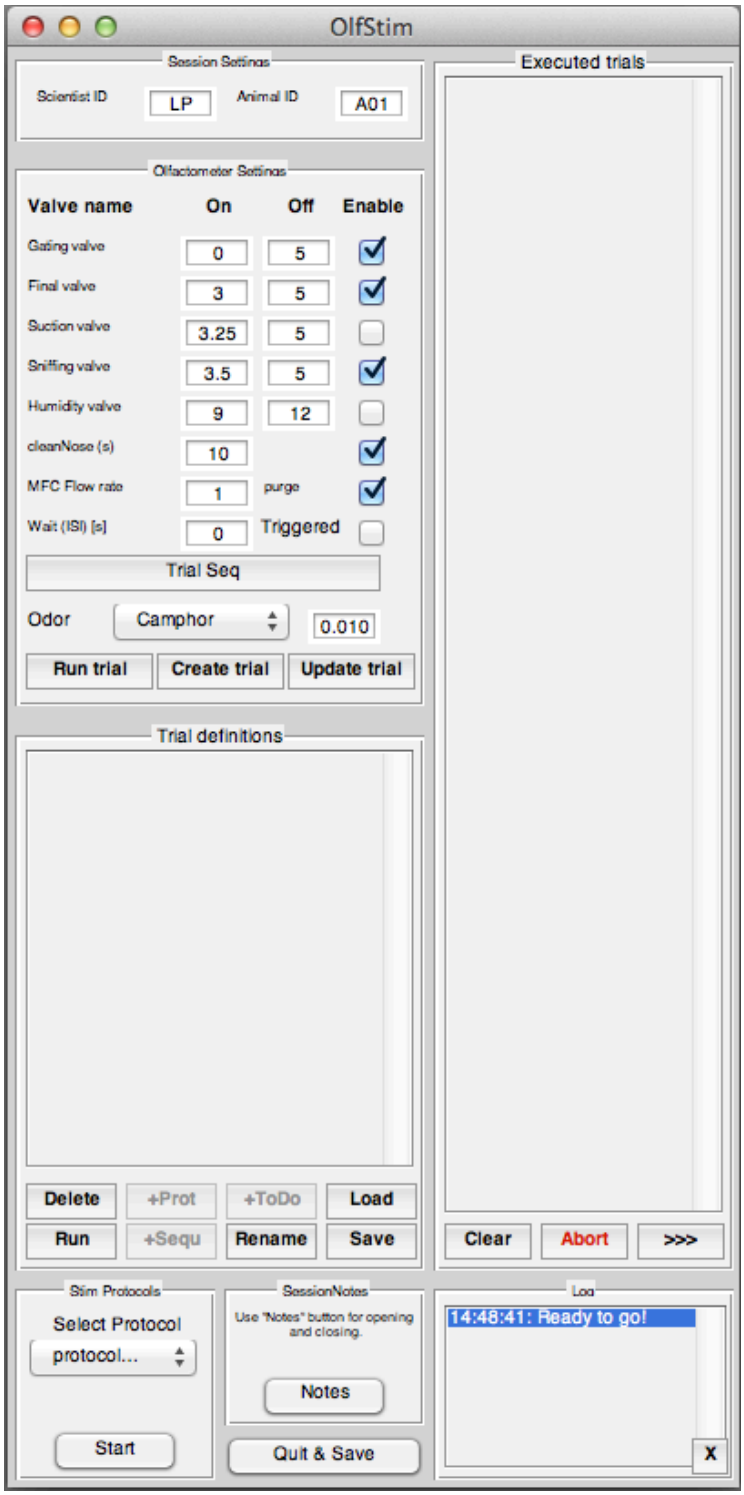


manualProgrammingStim

All the functionality described in manualStim are also available in manualProgrammingStim, with the additional feature of creating sequences of odor presentations with defineable inter trial intervals. It is also possible to load and save a smell structure containing the necessary information as a template for a sequence of odor presentation trials.



protocolsSJ_Stim



Running olfStim in test mode

In the Matlab commandline type

```
initOlfStim('testing')
```

or

```
initOlfStim('test')
```

This will start the test version of olfStim, which does not require the computer to be connected to an olfactometer. Most features not directly linked to olfStim-olfactometer interaction are active and can be tested. The test mode is useful if you want to start playing with the software but don't have an olfactometer connected, or if you're coding/hacking olfStim and want to test your code on your development machine. Keep in mind however that the test mode does not emulate the olfactometer's microprocessor, therefore not all of your code's potential bugs are caught.

Scripting olfStim

olfStim can also be scripted. This means if you don't want to use the GUI to control the olfactometer you can also do so by calling the functions of olfStim. This can be practical in several use cases, for instance if you have stereotypical stimulation sessions or if you want to access olfStim functionality from another program.

Initiation script

First open olfStim/scripting/olfStimScripting.m, the entry script into olfStim's scripting mode. It sets all necessary variables and paths and will then trigger the execution of the scripting protocol, which defines the progression of odor presentations.
Go through olfStimScripting.m and change all the marked lines to configure the scripting of olfStim for your computer and experiments.

Start the odor presentations by executing the initiation script and providing some arguments:

```
olfStimScripting(protocol,#OfTrials,scientist,animalID,interTrialInterval)
```

protocol – which scripting protocol should be used, #ofTrials – how many trials should be presented, scientist – your name, animalID – identification of the experimental animal, interTrialInterval – time between individual trials in seconds.

The resulting information of the olfactory stimulation session (the smell structure) will be saved by default in olfStim/UserData/data/Today'sDate_smell.mat. You can change this default by altering the corresponding lines in the initiation script in the "Save the resulting smell structure" section.

Scripting protocol

Decide which scripting protocol you want to use and provide its name as an argument to the initiation script. The scripting protocols handle the definition of events (which odor to present at what concentration, when to open which valve, etc.) for every trial and trigger the trials.
Currently there are two scripting protocols provided in olfStim: 'randomOdorPresentation' and 'presentPredefinedOdorList' – randomOdorPresentation protocol will randomly draw one of the odorants listed in the loaded odorant library file (structure called olfactometerOdors) and present it at the defined concentration, while presentPredefinedOdorList will load a predefined smell structure and then execute the trials as defined in the smell structure.

You can add your own scripting protocols by writing a Matlab function. The procedure is simple:

1. Add a folder with the name of your protocol prefixed with a plus sign into olfStim/scripting/+scriptingProtocols. For example choose the folder name '+myScriptingProtocol'
2. Add a Matlab m-file into this folder and name it identically, only omitting the plus sign. For the example above this file would be called 'myScriptingProtocol.m'
3. When you now call the initiation script from the command line, provide 'myScriptingProtocol' as an argument and your new function will be called after the initiation script has set up the necessary variables and paths.

For Developers

If using LASOM it is necessary to set variable 1 ($Var1) to 0 before the trial begins:

```
SetVar, $Var1, 0
```

After the trial begins the variable has to be set to 1 and this updated status has to be emitted to the host computer:

```
SetVar, $Var1, 1
EmitStatus
```

At the end of the trial set the variable back to 0 and emit the new value:

```
SetVar, $Var1, 0
EmitStatus
```

This is necessary for the startTrial.m file to know in which state olfactometer is. EmitStatus causes the sequencer to emit the updated status to the host computer.

Creating a new protocol

The most common type of customization to be done in laboratories using olfStim is to create new protocols, in case the general default protocols cannot deliver a certain function necessary for the lab's experiments. Components of protocols are programmed in a modular way, so that they can be combined easily.
Try to use these existing components, which are functions in +protocolUtilities as much as possible. You have access to all gui components through the olfStimH

handles struc. This means you can rearrange and change everything, but the functions will still be able to access the information without writing new code.
Components are for example the “Olfactometer settings” which consists of a panel with gui components to edit valve opening and closing times, total flow rates, etc.. But also it employs consistency checks in the background, extracts information automatically from the configuration files to populate its fields with default numbers, has functions to extract the values in the gui fields, and so on. This entire functionality can be added to ones protocol with a few calls to one function.
Other such components include a button for every odor loaded to the olfactometer, session settings, I/O features, notes, progress panel, log panel, etc..

Adding an olfactometer controller other than LASOM

In that case all functions in olfStim/olfactometerUtilities/+olfactometerAccess have to be replicated for the new olfactometer controller. If some functionality is not available in the new olfactometer controller, a dummy function still has to be present.