ANDREY ZIYATDINOV

BIOMIMETIC SET UP FOR CHEMOSENSOR-BASED MACHINE OLFACTION

DOCTORAL THESIS IN BIOENGINEERING

BIOMIMETIC SET UP
FOR CHEMOSENSOR-BASED MACHINE OLFACTION

ASPIRANT: ANDREY ZIYATDINOV

ADVISER: ALEXANDRE PERERA LLUNA

Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAII)
Universitat Politècnica de Catalunya, UPC

http://variani.github.io/thesis
September 2014

Andrey Ziyatdinov: *Biomimetic set up for chemosensor-based machine olfaction*
http://variani.github.io/thesis

Dedicated to my mother

Посвящается моей маме

ABSTRACT

The thesis falls into the field of bioengineering and more particularly into experimental set up for chemical gas sensing. Perhaps more than any other sensory modality, chemical sensing faces with major technical and conceptual challenges: low specificity, slow response time, long term instability, power consumption, portability, coding capacity and robustness.

There is an important trend of the last decade pushing artificial olfaction to mimic the biological olfaction system of insects and mammalians. The designers of machine olfaction devices take inspiration from the biological olfactory system, because animals effortlessly accomplish some of the unsolved scenarios in machine olfaction. In a remarkable example of an olfactory guided behavior, male moths navigate over large distances in order to locate calling females by detecting pheromone signals both rapidly and robustly.

The biomimetic chemical sensing aims to identify the key blocks in the olfactory pathways at all levels from the olfactory receptors to the central nervous system, and simulate to some extent the operation of these blocks, that would allow to approach the sensing performance known in biological olfactory system of animals. New technical requirements arise to the hardware and software equipment used in such machine olfaction experiments.

This work explores the bioinspired approach to machine olfaction in depth on the technological side. At the hardware level, an embedded computer is assembled, being the core part of the experimental set up. The embedded computer is interfaced with two main biomimetic modules designed by the collaborators: a large-scale sensor array for emulation of the population of the olfactory receptors, and a mobile robotic platform for autonomous experiments for guiding olfactory behavior. At the software level, the software development kit is designed to host the neuromorphic models of the collaborators for processing the sensory inputs as in the olfactory pathway.

Virtualization of the set up was one of the key engineering solutions in its development. Being a device, the set up is transformed to a virtual system for running data simulations, where the software environment is essentially the same, and the real sensors are replaced by the virtual sensors coming from especially designed data simulation tool. The proposed abstraction of the set up results in an ecosystem containing both the virtual array for data generation and the models of the olfactory system for data processing. This ecosystem can be loaded from the developed system image on any personal computer.

The scientific results have been published in three journal articles, two book chapters and conference proceedings. The main results on validation of the set up under the scenario of robotic odor localization are reported in the book chapters. The series of three journal articles covers the work on the data simulation tool for machine olfaction: the novel model of drift, the models to simulate the sensor array data based on the reference data set, and the parametrized simulated data and benchmarks proposed for the first time in machine olfaction.

This thesis ends up with a solid foundation for the research in biomimetic simulations and algorithms on machine olfaction. The results achieved in the thesis are expected to give rise to new bioinspired applications in machine olfaction, which could have a significant impact in the biomedical engineering research area.

ACKNOWLEDGMENTS

In this complex project, many people provided advice, assistance and support during the past six years. I remember gratefully:

For guidance in accomplishing this work, my advisor Alexandre Perera Lluna. I appreciate a lot your support, guidance and openness to share your knowledge with me. It has meant so much to me over the years, both professionally and personally.

For opportunity to do a part of my work in their laboratories, Santiago Marco and laboratory of the Intelligent Signal Processing (ISP) group at University of Barcelona, Paul Verschure and laboratory of the Synthetic Perceptive, Emotive and Cognitive Systems (SPECS) group at Pompeu Fabra University, and Romeo Beccherelli and laboratory of Consiglio Nazionale delle Ricerche (CNR) at Institute for Microelectronics and Microsystems. For giving access to the unique UNIMAN data set, Professor at the University of Manchester Krishna Persaud.

For working in close co-operation in the Neurochem project, Jordi Fonollosa, Luis Fernández, Miguel Lechón, Sergi Bermúdez Badia, Lucas López, Vasiliki Vouloutsi, Alex Chimeno Escudero and Emiliano Zampetti.

For the department-level support, director of the Biomedical Engineering Research Centre (CREB) Pere Caminal, and director of the PhD Program in Bioengineering Raimon Jané. For providing painless maintenance of the IT infrastructure, Guillem Riera Galmés. For coping with the paper work, Virginia Cabello and Marta Clarós. For studying and working together in the ESAII department, Helena Brunel, Juan Maynou, Raimon Massanet, Joan Josep Gallardo, Francesc Fernández, Jorge Jaramillo, Joan Francesc Alonso, Mónica Rojas, Umberto Melia and Leidy Serna.

For investing time and effort in revision of this manuscript, Oleg Osychenko. For translating the abstract of this manuscript from English to Spanish, Helena Brunel. For sharing with me their thesis manuscripts as a reference of the format of a collection of published articles, Yorgos Athanasiadis, Joan Francesc Alonso and Narges Dialami.

For funding my PhD study, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216916: Biologically inspired computation for chemical sensing (Neurochem). For giving me the opportunity to work at the Institute of Biomedical Research of Hospital de la Santa Creu i Sant Pau for the last two years of my study, head of the Unit of Genomics of Complex Diseases José Manuel Soria, and head of the Unit of Thrombosis and Haemostasis Juan Carlos Souto.

For open science and open-source software projects related to my research, the Debian developers, the Comedi project developers, the R core development team, many creators of R packages, the maintainers of The University of California at Irvine (UCI) Machine Learning Repository, and the classicthesis package for LaTeX.

For being on the other side of my PhD study, friends I met in Barcelona Maxim, Tanya, Eduardo, Martin, Israel, Mariella, Giovanna, Oleg, Lourdes, Gregory, Eva, Katya, Marina, Pavel and Narges.

For absolute confidence in me and care, my mother Lubov, my sister Lubava and my grandmother Raisa. For being together with me, Nina.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

PR	ordered projection
ORN	olfactory receptor neurons
OSN	olfactory sensory neurons
GL	glomeruli
AL	antennal lobe
MB	mushroom body
OB	olfactory bulb
OC	olfactory cortex
CP	conducting polymer
MOX	metal oxide semiconductor
FSS	feature subset selection
FFT	fast Fourier transform
PCA	principal component analysis

LDA	linear discriminant analysis
ICA	independent component analysis
MSE	mean squared error
EL	extended Langmuir isotherm
OP	orthogonal projection
CC	component correction
SDK	software development kit

1 INTRODUCTION

1.1 MACHINE OLFACTION

1.1.1 *Historical perspective*

In 1991, Linda Buck and Richard Axel – 2004 Nobel laureates in Medicine and Physiology – discovered the family of proteins (seven trans-membrane proteins) that mediate the transduction of chemical information into electrical signals through olfactory receptor neurons (ORN) [Buck and Axel 10]. Later in 1994, Vassar and colleagues discovered the ordered projection (PR) of ORN onto the olfactory bulb based on the protein expressed by each ORN (Figure 1.1) [Vassar et al. 93]. These and other recent findings in biology have considerably improved our understanding of the olfactory system.

Back 1982, Krishna Persaud and George Dodd first introduced the use of arrays of broadly-selective chemical sensors targeted to discriminate between a wide variety of odors [Persaud and Dodd 69]. Sensors in the array conformed to the same rule of non-specificity as ORN in the olfactory system, and the discrimination among odor mixtures could be achieved without the use of highly specific receptors. The authors referred to the proposed device as an *electronic nose*.

1.1.2 *Biological olfaction*

The olfactory system of vertebrates and insects share a common basic architecture. Their olfactory pathway can be divided into three basic building blocks (Figure 1.1):

- olfactory epithelium;
- olfactory bulb (OB) in vertebrates and antennal lobe (AL) in insects;
- olfactory cortex (OC) in vertebrates and mushroom body (MB) in insects.

In the olfactory epithelium, the molecular properties of the odorants are transduced into electrical signals through a collection of ORN, also referred as to olfactory sensory neurons (OSN). For instance, mammals have tens of millions of ORN [Hildebrand and Shepherd 37], which belong to as many as 1000 different types of receptors [Ma and Shepherd 54].

One of the prevailing hypotheses about olfactory primary reception is that ORNs do not respond to specific molecules, but rather to specific molecular features of an odorant molecule, commonly referred to as odotopes [Shepherd 85, 86]. For example, the odotopes can be carbon-chain length, the presence of benzene rings or different functional groups such as esters or aldehydes.

Since most odorants in the environment consist of mixtures of volatile molecules and each molecule can contain several odotopes, an odorant is then detected as a large combination of specific odotopes. For instance, roasted coffee has been estimated to contain on the order of 600 volatile components.

ORNs are placed in the first relay of the olfactory system inside the brain: the olfactory bulb in vertebrates and antennal lobe in insects. ORNs project the signal in a very orderly

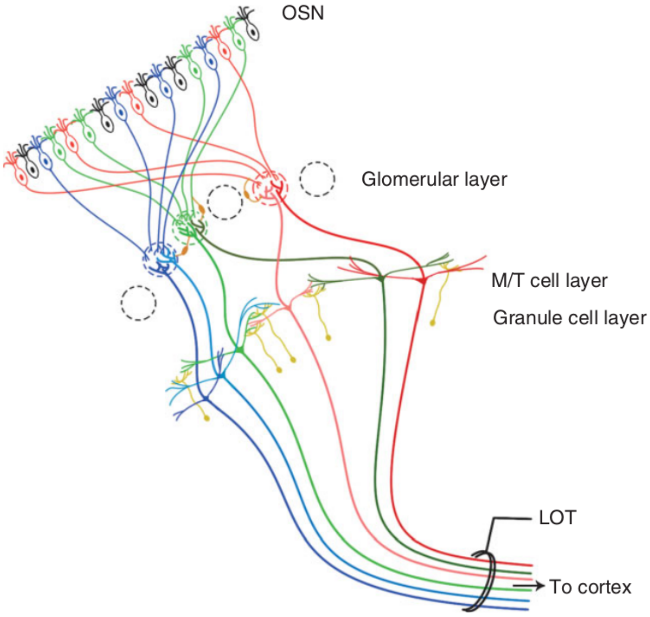


Figure 1.1: The scheme of the circuitry of the olfactory bulb and its inputs. Figure source: [Khan et al. 49].

fashion into spherical regions of neuropil known as glomeruli (GL) (Figure 1.1). Each glomerulus receives axons from one type of ORN, and each ORN type projects into one or a few glomeruli [Vassar et al. 93, Ressler et al. 76]. At the glomerular level, olfactory information can be thought to be represented by an image of the molecular features of the stimulus. Further two types of neurons can be found in the olfactory bulb: projection neurons (mitral and tufted cells) and local interneurons (periglomerular and granule cells) [Shepherd et al. 87].

In the olfactory cortex the holistic representation of an odor is formed and odors are identified [Shepherd et al. 87]. Recurrent connections are pervasive, and the olfactory system is no exception [Mountcastle 61]. There exist feedback connections from cortex to granule cells, which are believed to modulate their inhibitory effect in the olfactory bulb. The representation of odors by a particular glomeruli in the olfactory bulb is transformed in the cortex into highly distributed and multiplexed odor maps [Zou Z and LB 111]. A local population of cells respond to specific combinations of inputs from the glomeruli of the olfactory bulb acting as coincidence detectors. Different odorants elicit distinct, sparse and distributed but partially overlapping activity patterns in the piriform cortex.

1.1.3 *Artificial olfaction*

Machine olfaction device

A machine olfaction device is traditionally referred to as the *electronic nose*. This device is commonly based on an array of broadly-tuned non-specific chemical sensors, that respond to an environment in the presence of an odor and form a characteristic pattern, also known as a *smell fingerprint* [Persaud and Dodd 69, Pearce et al. 64]. The device is typically composed of the following parts:

GAS DELIVERY SYSTEM. The gas samples are delivered to the sensors via the delivery system. The system might contain a mass flow controller, a selector of channels or sample, a sensor camera and a pump. This part of the device is optional, as sensors can be used to measure in less-controlled conditions, for example, in an open-sampling system.

SENSOR ARRAY. The interaction between gas molecules and a sensing material of the sensors leads to the signal transduction from chemical-physical quantities to analytically useful signals (typically electrical signals).

INTERFACE ELECTRONICS. The implementation of electronic circuits depends on the type and operational mode of the sensors. The electronics is mainly responsible for digital-to-analog conversion with the best signal-to-noise ratio.

DATA PROCESSING UNIT. The algorithms of signal processing and pattern recognition (pattern regression) are used to evaluate the acquired data and formalize the response to a subject of the study, for example, classification of two distinct odors.

It is important to note that the link between human odor impressions and the electronic nose output patterns exists only in particular and well-defined cases, as the technology is still far from mimicking the biological olfactory system [Röck et al. 78].

1.2 CHEMOSENSORS

Chemical sensors or chemosensors combined in an array were proposed as a low-cost and high-throughput alternative to analytical instruments in machine olfaction [Persaud and Dodd 69], and such approach has been widely exploited over the last decades. The principles to transduce the chemical information contained in the measured gases into an analytical signal can be different: electrical, thermal, optical, mass-transportation and others. The chemosensor technology includes metal oxide semiconductor (MOX), CP, chemocapacitors, metal oxide semiconductors field-effect transistors (MOSFET), quartz crystal microbalance (QCM), surface acoustic wave (SAW), surface plasmon resonance (SPR), fluorescence and others [Pearce et al. 64].

Figure 1.2 shows a typical response of a CP sensor to a gas pulse at a certain concentration. The measured signal is the resistance of the sensor.

1.2.1 *Conducting polymer sensors*

Interactions between analyte and conducting polymers can be caused on different parts of the sensor device: In the *bulk*, the number of carriers and bulk mobility are changed, and the CP acts either as e-donor or e-acceptor. Between the *electrodes*, the height of the Schottky barrier is modulated. On the *surface*, the conductivity on the surface is altered. On the *substrate*, the surface conductivity of the oxide – a typical material of the substrate – is changed with the degree of hydration. Hence, the possible phenomena involved in the transduction mechanism in the CP sensor are electron/proton transfer, barrier reduction between grains, swelling and adsorption [Janata and Josowicz 42].

The term *sorption* means an action produced by either absorption or adsorption processes. *Absorption* is a chemical process in which one substance take in another substance (its atoms, molecules or ions). For example, gases are absorbed by a solid. *Adsorption* is different from absorption in the sense that the molecules are taken up by the surface rather than in the volume. For example, reagents are adsorbed to solid catalyst surface.

In 1992, Topart and Josowicz studied the interaction between methanol analyte and the popyperrole conducting polymer by observing several physical quantities: change in mass,

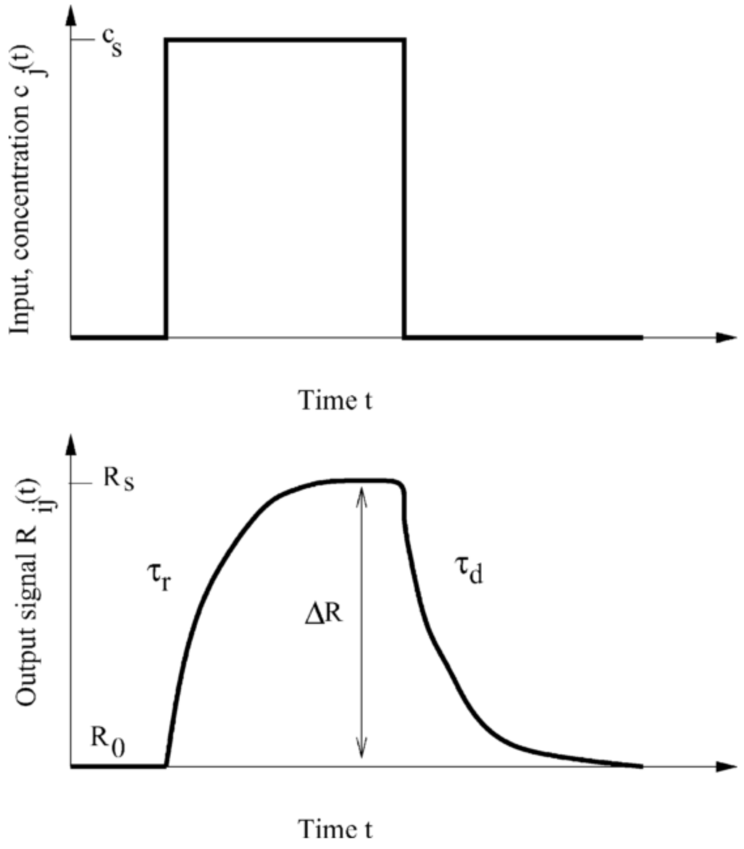


Figure 1.2: The characteristic response of a conducting polymer sensor to the odor pulse at a certain concentration. Figure source: [Distante et al. 16].

work function and optical absorbance [Topart and Josowicz 91]. It was demonstrated that the analyte sorption – adsorption and diffusion processes – was the driving force in the polymer-analyte interaction. In addition, a Langmuir-type isotherm showed a good fit to the experimental data.

When the transduction mechanism is mainly adsorption-based, the analytes are likely not reactive with conducting polymers under normal conditions. The analyte molecules are adsorbed by the polymers, that affects the properties of the sensing material and makes these analyte molecules detectable. Benzene and toluene are representative analytes for this case.

Langmuir sorption isotherm

In 1916, Irving Langmuir empirically derived the Langmuir isotherm. The isotherm describes the adsorption process on a plane surface, which is assumed to form a unimolecular layer.

The amount of the absorbed gas (*adsorbate*) is expressed by equation:

$$q = q_s \frac{bp}{1 + bp}, \tag{1.1}$$

where p is the gas pressure or concentration, b is the sorption affinity, and q_s is the sorption capacity. In 1931, Markham and Benton conducted one of the first experiments on the adsorption of oxygen, carbon monoxide and carbon dioxide by silica. It was shown that the Langmuir isotherm fits the experimental data well at low pressure.

The extended Langmuir isotherm (EL) is an extension of the Langmuir isotherm applied to a mixture of N components, when the adsorbate molecules of the gases are assumed to not interact among each other. The amount of the absorbed gas in the mixture is given by equation:

$$q_i = q_{s,i} \frac{b_i p_i}{1 + \sum_{j=1}^{N} b_j p_j}, i = 1, 2, ..., N, \tag{1.2}$$

where the index i is used to encode the values for the i-th gas component in the mixture. Consequently, the isotherm parameters $q_{s,i}$ and b_i for each gas i are independently determined from the Langmuir isotherm of a single gas (Equation 1.1).

In the derivation of Equation 1.2, it is implicitly assumed that all gas components have equal access to adsorption sites on the surface. This assumption is thermodynamically inconsistent [Bai and Yang 4], and a number of modification in EL have been proposed to account for molecular interactions in the adsorbate phase, uniform structure of the absorbent surface and other aspects. The generalized sorption model was introduced with corrections due to adsorbate size, loss of symmetry or disassociation, clustering and adsorbant molecular interactions [Martinez and Basmadjian 60]. Another model was proposed based on the multi-region extended Langmuir isotherm [Bai and Yang 4].

The Freundlich isotherm is another empirical equation that describes the adsorption process. This isotherm is also extended to the problem of a mixture of N components in the form of Langmuir-Freundlich equation:

$$q_i = q_{s,i} \frac{b_i p_i^{1/n_i}}{1 + \sum_{j=1}^{N} b_j p_j^{1/n_j}}, i, j = 1, 2, ..., N, \tag{1.3}$$

where n_i is another constant for gas component i, in addition to other two component-specific parameters $q_{s,i}$ and b_i found in Equation 1.2. Limitations of the Freundlich isotherm are similar to those reported above for EL.

The Langmuir-type isotherms cannot avoid errors on estimation of multi-component adsorption process, because such phenomena as adsorbate size or adsorbent heterogeneity are not taken into account [Sircar 88]. Despite the existence of some sophisticated Langmuir-based models [Martinez and Basmadjian 60, Nitta et al. 63], the Langmuir isotherm in its original form (Equation 1.2) remains to be the most widely used model in gas separation discipline [Yang 103].

1.2.2 *Models for conducting polymer sensors*

Model by Gardner et al.

The model assumed the Langmuir adsorption isotherm for the sake of simplicity. The diffusion equation was written for the geometry of a planar film, that allowed to use one geometrical dimension in the model. Once two dimensionless distance and time variables χ and τ were introduced, the equations were written in the dimensionless form. In particular, the function $\gamma(\chi, \tau)$ quantified the adsorption and desorption concentration of the gas distributed from one site to another side of the film , and the function $\theta(\chi, \tau)$ quantified site occupancy on the surface side of the film [Gardner et al. 24].

A family of analytical solutions for the two adsorption and diffusion equations were derived under different boundary conditions. Then the conductance of the film was derived as a linear function of the site occupancy $\theta(\chi,\tau)$:

$$\sigma_{\chi,\tau} = \sigma_0(1 - S\theta(\chi,\tau)), \tag{1.4}$$

where S is the gas sensitivity of the polymer, σ_0 is the conductivity in the absence of the gas.

It was shown that analytical results correlated well with the experimental data. The use of the model is limited when the polymer material is anisotropic or the homogeneity assumption for the diffusion and the conductance models is not valid.

Model by Bissel et al.

In contrast to the previous model by Gardner *et al.*, Bissel and colleages studied the steady-state electrical resistance of the sensor (transient change in the resistance was not considered) and employed the theory of volatile organic chemical (VOC) partition between gaseous and condensed phases [Bissell et al. 9]. It was stated that the kinetics of mass transport of analyte vapors into the sensor film obeys the linear proportion law established in the field of gas chromatography.

The partition coefficient K is defined as the ratio between analyte concentration in the stationary phase C_s and analyte concentration in in the mobile gas phase C_g:

$$K = \frac{C_s}{C_g} = \frac{RT\rho_1/M_1}{\gamma_2 p_2}, \tag{1.5}$$

where R is the molar gas constant, T is the temperature defined in Kelvin, ρ_1 and M_1 are the density and molecular weight of the polymer respectively, γ_2 is the vapour activity coefficient, and p_2 is the saturated vapour pressure of the solute vapour.

The change in the polymer resistance is proportional to the concentration of the dissolved analyte under certain conditions (mass uptake below 5%) for a homologous series of compounds:

$$\Delta R = dR/R = k_i C_s, \tag{1.6}$$

where k_i is a *transduction constant*, describing how effectively the amount of the absorbed analyte i is translated into the resistance signal.

The final equation of the model is derived from Equations 1.5 and 1.6 and has the following form:

$$\log(1/C_g) = \log(1/p_2) + \log(k_i/\gamma_2) + Const. \tag{1.7}$$

Model by Lei et al.

Lei and colleages proposed a heuristic microscopic sensor model based on the Langmuir isotherm for polypyrrole-based (PPy-based) sensors. The overall resistance of the film (R) is represented as a parallel connection of several layers (n layers), and each layer consists of several resistors (m resistors with resistance r) in series.

The sensor model states a liner relationship of the reciprocal of the resistance change against the reciprocal of the gas concentration according to the following equation:

$$\frac{1}{\Delta R_t} = \frac{n}{m(r_1 - r_0)} + \frac{n}{m(r_1 - r_0)K_m}\frac{1}{C_g}, \tag{1.8}$$

where ΔR_t is the resistance difference after and before gas sorption; K_m the adsorption equilibrium constant; C_g concentration of the gas; r_1 and r_0 site resistance when the site is occupied or empty respectively.

The model was shown to interpret well the behavior of PPy-based composite sensors exposed to ethanol and methanol vapors.

1.3 ANALYSIS OF CHEMOSENSOR ARRAY DATA

1.3.1 *Data processing chain*

Once raw data from a sensor array are measured, stored and prepared for the data analysis, the data processing unit of the machine olfaction device starts working, as described above in Section 1.1.3. A single data sample can be represented in the matrix form.

$$\mathbf{X_{M \times S}} = \begin{pmatrix} x_{1,1} & x_{1,2} \dots x_{1,S} \\ x_{2,1} & x_{2,2} \dots x_{2,S} \\ \dots & \\ x_{M,1} & x_{M,2} \dots x_{M,S} \end{pmatrix}. \tag{1.9}$$

The number of columns is equal to the number of sensors S, and the number of rows corresponds to the number of readings M extracted from a sensor. Readings of each sensor commonly represent the transient signal acquired from the sensor, as shown on the Figure 1.2. The value of M depends on the sampling frequency of the acquisition system.

Hence, a complete data set of the N samples can be represented as a three-dimensional array.

$$\mathbf{A_{N \times S \times M}} = \{\mathbf{X_1}, \mathbf{X_2}, \dots \mathbf{X_N}\}. \tag{1.10}$$

A common data processing chain for the machine olfaction device is shown on Figure 1.3. The raw measurements in the format of the A matrix from Equation 1.10 further undergo a number of processing stages or blocks. Processing blocks *Pre-processing* and *Feature Extraction* or *Dimensionality Reduction* are outlined in Sections 1.3.2 and 1.3.3, respectively. *Drift Compensation* block (not shown on the Figure 1.3) is usually considered as an advanced pre-processing block, and a special attention is given to it in Section 1.3.4. The material about *Classification* or *Pattern Recognition* is covered elsewhere in the machine learning literature [Trevor Hastie, Robert Tibshirani 92].

1.3.2 *Pre-processing of raw data*

The objective of the pre-processing stage is to compensate noise artifacts in data resulting from various experimental circumstances, that influence the measurements and make

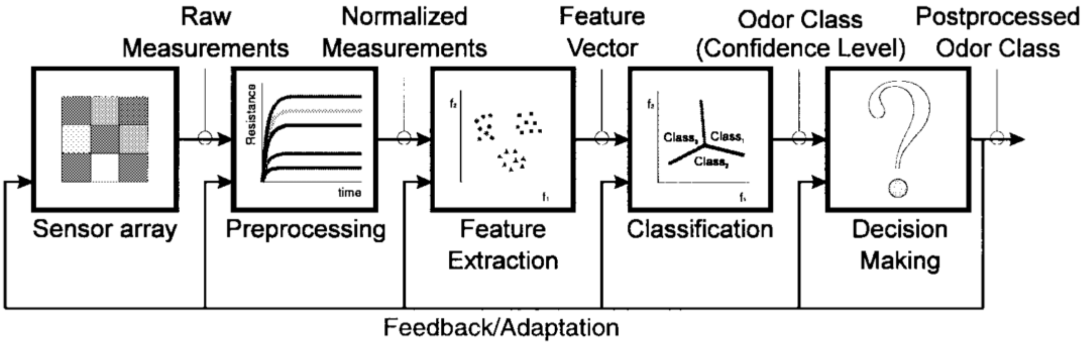


Figure 1.3: The data processing chain in the machine olfaction device. Figure source: [Gutierrez-Osuna 32].

them inconsistent [Gutierrez-Osuna and Nagle 34, Jurs et al. 43]. Pre-processing methods depend on the sensor technology, but there are some common problems. An example of such problems is the additive drift observed among samples measured under the same conditions. Figure 1.4 shows a sequence of four sensor signals in response to four gas pulses. The signal values at the start of the pulse are different for four samples, and the observed additive noise has to be corrected on the pre-processing stage.

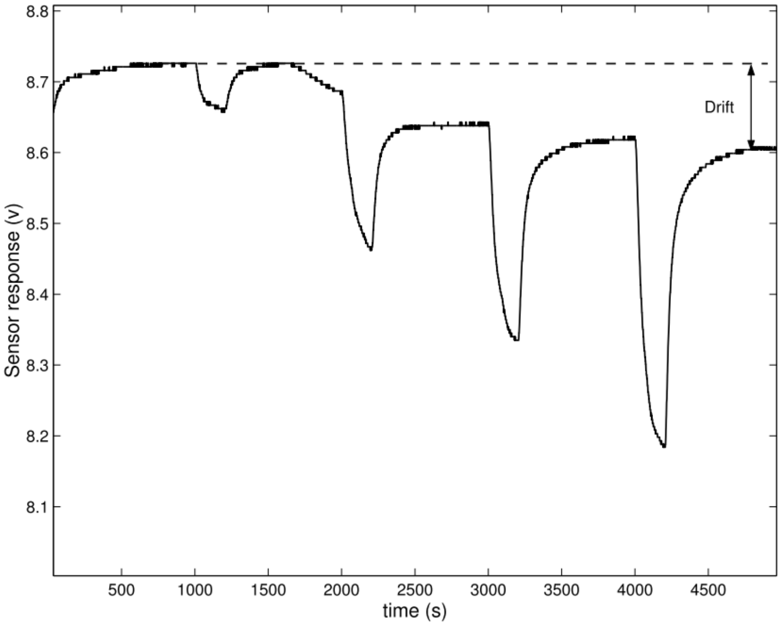


Figure 1.4: An example of the additive drift in the sensor signal in response to a sequence of four gas pulses. Figure source: [Bermak et al. 6].

One of the standard methods to compensate the sample-to-sample additive drift is the *baseline correction*. The formula of the transformation applied to the matrix $\mathbf{X}$ from Equation 1.9 is the following.

$$\text{For a given sensor j: } x'_{i,j} = \frac{x_{i,j} - x_{1,j}}{x_{1,j}}, \ \forall\ i = \overline{1, M}. \tag{1.11}$$

The intensity of the sensor signals can be equalized by the *normalization* transformation.

$$\text{For a given sensor j: } x'_{i,j} = \frac{x_{i,j}}{\sum_{i=1}^{M} x_{i,j}}, \ \forall\, i = \overline{1, M}. \quad (1.12)$$

If the sensor signals show non-linear characteristics, the *sensor linearization* transformation can be used.

$$\text{For a given sensor j: } x'_{i,j} = \log(x_{i,j}), \ \forall\, i = \overline{1, M}. \quad (1.13)$$

The *auto-scaling* transformation is commonly applied in the statistical analysis of data in the matrix **X** from Equation 1.9.

$$\text{For a given sensor j: } x'_{i,j} = \frac{x_{i,j} - \mu_j}{sd_j}, \ \forall\, i = \overline{1, M}, \quad (1.14)$$

where μ_j is the mean of the readings of sensor j, and sd_j is the standard deviation of the readings of sensor j.

1.3.3 *Dimensionality reduction*

Once the raw data **X** is cleaned in the pre-processing stage, a proper data-processing method on feature extraction or dimensionality reduction is commonly followed by the pattern recognition stage. The natural reason to perform the feature extraction procedure is due to the redundancy and cross-selectivity of the sensors in the array. This issue on processing of the sensor array data can be viewed either as the *collinearity* problem in the multivariate statistics or as the *curse of dimensionality* phenomena in the statistical pattern recognition.

A mathematical representation of a dimensionality reduction operation can be expressed as following.

$$f : \mathbf{x} \to \mathbf{y}, where \mathbf{x} \in \mathbb{R}^M, \mathbf{y} \in \mathbb{R}^P \text{ and } P < M. \quad (1.15)$$

The original vector x from the M-dimensional space is mapped to the new vector y from another space of the lower dimension P. This operation is performed by the mapping function f, that aims to compress the information contained in the original data with a minimal loss of quality.

Several considerations in the design of dimensionality reduction methods are important. Such methods applied to sensor array data (the matrix **A** in Equation 1.10) belong to two categories:

- *waveform compression* of the transient signal independent among the sensors (data vectors of the length M stored on third dimension of the matrix **A**);
- *dimensionality reduction* of data aggregated from all the sensors (data vectors from the two dimensional $S \times M$ space on the second and the third dimensions of the matrix **A**).

Two main validation criteria used in evaluation of the quality of the data compression are based on:

- *signal representation* measures of the compression quality;
- *signal classification* measures of the discrimination power.

Feature selection methods

The objective of a feature subset selection (FSS) method is to find the best subset of features that maximizes the information content (*filters*) or the discrimination power (*wrappers*). The wrappers hold a classifier inside a optimization loop and typically perform better than filters. The advantage of the filters is lower computational cost.

The time needed to traverse all the features is exponential on the number of features. Several search strategies are commonly applied to avoid the exhaustive search: exponential (branch-and-bound), sequential (sequential forward selection and sequential backward selection) and randomized (simulate annealing and genetic algorithms).

The FSS approach is rarely used in machine olfaction. The methods were typically tested on arrays with small number of sensors, and performance was optimized to resolve a particular pattern recognition problem. For example, a wrapper approach was tested on sensor array data in [Eklöv et al. 18], and a comparative study of different FSS strategies was conducted in [Gutierrez-Osuna 29]. The validation criterion in both works was defined as the classification accuracy among odor classes. The second work showed that studied methods had similar performance of 25-30% increase in the predictive accuracy and 50% reduction in the size of the feature set.

Heuristic feature extraction

A variety of heuristic features can be defined for the sensor waveform or sensor transient. The signal measured at the 60% level of the stabilized value at the rise time was studied as a feature in [Vilanova 97]. A group of features were derived based on the first and the second derivatives of the sensor signal [Roussel 81]. The proposed features were evaluated according to three criterion: repeatability, discrimination and redundancy. A review on heuristic features in chemosensor signals can be found in [Gibson 26].

The methods based on heuristic feature extraction were actively explored in the early years of machine olfaction, in the 1990s. Now these methods are mostly depreciated among the others, because the approach does not propose a general way to extract features.

Waveform-based feature extraction

The method of *waveform sub-sampling* reduces the size of the feature set from M to m by means of sub-sampling and anti-aliasing post-filtering of the signal. If F_s denotes the sampling frequency, the filter frequency is the following:

$$F_f = \frac{mF_s}{2M}. \tag{1.16}$$

The kernel based sub-sampling method is implemented in [Gutierrez-Osuna and Nagle 34]. The drawback of the sub-sampling method is that the application of the filters limits the bandwidth of the extracted information substantially.

The method of *waveform modeling* is based on an interpolation of the transient data points of the signal. Implementations of such models include multi-exponential decomposition, Pade-Z approximation [Gutierrez-Osuna 33], METS decomposition [Samitier et al. 83], and Lorentzians analytical models [Carmel 11]. A general solution is not always possible, because the methods require a prior assumption about the shape of the waveform.

The method of *spectral-based feature extraction* employs the fast Fourier transform (FFT) to transform the signal from the time domain to the frequency domain. Applications of the FFT method to sensor array data commonly included the FFT signal processing and a classifier for validation of the results [Heilig and Ba 36, Fort et al. 21]. The sensor signals are not constant in their FFT parameters over time, that limits the use of the FFT method.

The method based on *wavelet decomposition* defines time and frequency representation of the signal by means of a family of base functions called wavelets. The wavelet decomposition can be thought of as an alternative to the short time Fourier transform with finer time resolution at low frequencies due to the concept of scale functions. An example of the application of wavelet decomposition to sensor array data can be found in [Phaisangittisagul 70]. The drawback of the wavelet decomposition approach arises when the length of the transient is long that makes the computational time impractical.

Variance-based feature extraction

The variance-based methods are different from the previously described waveform-based methods by the optimization criterion. The variance-based methods are focused on the informational content of the data. Three linear methods widely used in application to sensor array data will be briefly presented: principal component analysis (PCA), linear discriminant analysis (LDA) and independent component analysis (ICA).

PCA is an unsupervised method that defines a linear projection $\mathbf{x}'$ of data vector $\mathbf{x}$ (data vector of a single sample) by estimating an orthogonal sub-space where the most of the variance is captured.

$$\mathbf{x}' = \mathbf{P}^T \mathbf{x}. \tag{1.17}$$

The vectors in the p columns of the matrix $\mathbf{P}$ are called principal components.

The correlation matrix in the new sub-space is diagonal, as the projection matrix is orthogonal.

$$\mathbf{R}_{\mathbf{x}'} = E[\mathbf{x}'\mathbf{x}'^T] = E[\mathbf{P}^T \mathbf{x}\mathbf{x}^T \mathbf{P}] = E[\mathbf{P}^T \mathbf{R}_\mathbf{x} \mathbf{P}] = \Lambda = diag(\lambda_i), \ \ i = \overline{1,p}, \tag{1.18}$$

where the eigenvalues of the correlation matrix are denoted as λ_i.

The number of principal components p is selected such that the most of the variance (information) in the data is captured and the error on estimation (compression) $\hat{\mathbf{x}}$ of the original data $\mathbf{x}$ is minimized. The error in terms of mean squared error (MSE) is defined as following.

$$E[||\mathbf{x} - \hat{\mathbf{x}}||] = \sum_{i=m+1}^{N} \lambda_i. \tag{1.19}$$

The use of the PCA model for feature extraction has a number of limitations. The analysis is prone to outliers, the higher-order components could contain valuable information, and the sources of captured variance are not related to the discrimination problem directly.

LDA is different from PCA in the sense that the LDA algorithm searches for directions of the data variance that are efficient in discrimination of a given set of classes. Hence, LDA is a supervised approach. The LDA projection can be represented as a linear combination or a scalar dot product:

$$\mathbf{y} = \mathbf{w}^T \mathbf{x}, \tag{1.20}$$

where $\mathbf{w}$ is the vector of weights of size M.

The LDA algorithm consists in a division of the samples y_i into subsets, for example, $\mathbf{y_1}$ and $\mathbf{y_2}$ for two-class problem. The algorithm finds the direction of $\mathbf{w}$ that achieves two objectives: a higher separation between the two classes, and reduces the variance in each class.

Similar to PCA, LDA can fail to produce a good septation among classes if the classification problem is not linearly separable. The data set has to be balanced in the number of samples per class, and the algorithm has to be controlled to avoid the over-fitting issue of the LDA algorithm. The use of LDA in application to sensor array data can be found in [Cerrato Oliveros 12].

Contrary to both PCA and LDA, ICA searches for mutually independent directions of variance. The ICA problem is also known as blind source separation, where unknown source signals are separated from their linear mixtures using.

One of the definitions of the blind source separation problem is Pearson ICA [Karvanen and Koivunen 46].

$$\mathbf{X} = \mathbf{A}\mathbf{S}, \tag{1.21}$$

where columns in the matrix $\mathbf{S} = [\mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s_m}]^T$ are mutually independent random variables, the matrix $\mathbf{A}$ is an unknown invertible matrix of the size $m \times m$, and m is the number of the source signals.

Application of PCA and ICA to sensor array data was studied in [Kermit and Tomic 48]. The ICA method was shown to have a better discrimination performance by counteracting the sensor drift more efficiently than PCA. The authors stress the point that the results in favor to ICA are valid for their particular data sets, and that can be changed in application to another data set. The results based on ICA also depend on a particlular implementation of the ICA algorithm.

1.3.4 *Drift compensation*

The drift phenomena observed in measured data can be defined as gradual changes in measured quantity that is assumed to be constant over time [Artursson et al. 1]. This definition is mostly related to the long-time measurements. The drift variance is often an inevitable part of the data, and it is difficult to control in an experimental set up.

Drift in chemosensors

The drift noise in chemosensor arrays can be attributed into two categories. The drift is referred to as *sensor noise* or *short-time drift* when the noise is observed locally in time for the readings from a single sensor. The drift is referred to as *common drift* or *long-term drift* when the noise is observed in the sensor array data (multivariate data) and the drift affects the whole array for a long time.

The device instability is the the main source of the drift in a single sensor. The phenomenon of reorganization of the sensing material leads to changes in the basic characteristics of the sensor device like sensitivity or selectivity, and it is known as *sensor aging*. Another phenomenon called as *sensor poisoning* occurs when the chemical reactions on the sensing layer cause irreversible binding processes, that results in contamination of the sensor. *Warming up* of the sensor and *hysteresis* effect cause local instability in the regime.

The drift common to all the sensors in array has also several sources. *Sampling* protocol and the order of exposition of gas samples lead to the memory effect. *Environment* conditions strongly influence the whole system of the machine olfaction device, and the environmental factors include the ambient temperature, the humidity and other background effects. The system of *odor delivery* can induce the noise due to the inconstant flow rate, the gas leakage and other unwanted effects.

Compensation approaches

The objective of a drift compensation method is to enlarge the life time of the sensor array in the presence of drift. Methods on the drift compensation can be divided into three categories:

1. improvements in the experimental set up;
2. signal processing methods;
3. pattern recognition methods.

A periodic re-calibration of the array is a common approach to counteract with the drift in the experimental set up. The re-calibration has an additional cost for operation of the device, and it is considered to be the last choice for the drift compensation. An alternative approach is to use some reference sensors in array for tracking of the drift [S.-Y. Choi et al. 82].

The temperature modulation for organic coated gas sensors was shown to improve the results on counteraction to the long-term drift [Roth et al. 80], while this improvement in the set up serves for a more general task to increase the data dimensionality by enriching the sensor signals.

Another approach was found in transient feature extraction from the sensor signals [Wilson and DeWeerth 101]. It was shown that the transient signal can be converted to binary patters by a thresholding algorithm, that resulted in the automatic compensation of the additive part of drift.

Drift compensation methods of interest in this work belong to the group of signal processing methods. These methods can be generally split into two groups: univariate and multivariate.

Univariate methods

The baseline correction methods described in the Section 1.3.2 are regarded as univariate methods on drift compensation.

An approach similar to the baseline correction method was proposed in [Fryder et al. 23]. The baseline level was measured by a calibrant gas which must be chemically stable over time and representative among the samples related to the other gases.

Two short-term and long terms temporal models were trained for the sensor data measured for a reference gas [Haugen et al. 35]. The use of the two models in the periodic calibration of the sensors showed good results on drift compensation (for industrial applications).

Multivariate methods

It is a common practice that the drift noise still exists in the sensor array data after application of any univariate method of drift compensation. Compensation of the drift noise, which in general is a multi-source non-linear event, is a hard task especially in the measurement periods of weeks or months. The multivariate methods are indended to improve the results of the univariate methods by taking advantage of the data from all sensors in array.

An example of multivariate drift shown on the PCA score plot is given on Figure 1.5. There are eight gas classes of pure analytes and their mixtures. A non-linear drift direction is clearly observed on the left side of the Figure. The data corrected by the method described in [Artursson et al. 1] are shown on the right side on the Figure.

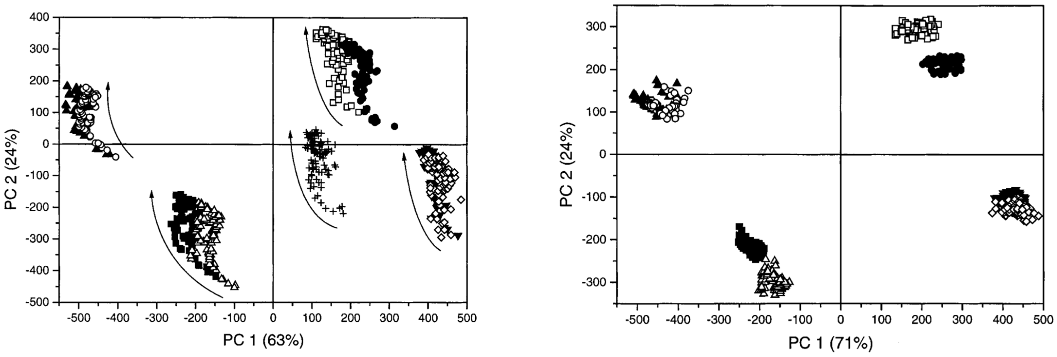


Figure 1.5: An example of the multivariate drift correction applied to sensor array data for 8 gas classes. Figure source: [Artursson et al. 1].

The approach based on *adaptive clustering* addressed the drift problem in machine learning [Freund and Mansour 22]. The method relies on periodic classification of the samples and requires all odor classes to be sampled relatively frequently. Otherwise, the learned patterns are lost. Different adaptation methods were proposed in application to sensor array data, for example, mean updating adapting clustering [Holmberg 38], adaptation based on the Kohonen self-organizing maps [Davide et al. 15, Marco et al. 58, Distante et al. 17].

The approach based on *self-identification theory* was tested by a dynamic model in [Holmberg et al. 39]. The sensors were supposed to co-vary over time showing some common components in response. The coefficients of the model were computed periodically by a recursive least-squares procedure. The method also requires an extensive and consecutive collection of samples.

The method of *calibration transfer* builds a model of the drifting calibrant gas by partial-least square regression [Tomic et al. 90] and and neural network [Balaban et al. 5]

The approach based on *orthogonal signal correction* comes from the field of analytical chemistry [Wold 102]. Given two matrices **X** of sensor array data (independent variables) and a concentration matrix **C** (dependent variables), the method performs the subtraction of the components in **X**, such that these components are orthogonal to **C** and explain the maximum of variance observed in **Y** and

The method of *component deflation* was developed in [Gutierrez-Osuna 30]. Latent variables for the calibrant and sensor data were calculated to explain the drift variance and co-vary. Canonical correlation analysis or partial-least squares were used for the estimation of the latent variables.

The method based on *component correction* was proposed in [Artursson et al. 1]. The drift multivariate direction or the subspace **V** is estimated by PCA on the data measured on the reference gas. The next step is the bilinear transformation or component correction (CC) computed on the data **X**.

$$\mathbf{X}' = \mathbf{X} - (\mathbf{X}\mathbf{V})\mathbf{V}^{\mathsf{T}} \tag{1.22}$$

The CC method belongs to a group of orthogonal projection (OP) methods of multivariate statistics. These methods, targeted to reduce the data dimensionality, search for dimensions of the subspace that describe maximum variance related or unrelated to the information of interest. The subspace **V** given in Equation 1.22 is thought to contain the drift-related information. The intention of the CC method is to make the posterior prediction of the model independent on the influence of non-desired variations observed in **V** and improves data and model interpretation. The estimation of **V** is a critical point in the given approach, and it can be accomplished by various heuristics. A problematic situation arises when this subspace in **V** contains also the discriminant information, and an optimum tuning of the method has to be found.

The CC method has been one of the most popular OP methods in the field of machine olfaction, and it can be regarded as a benchmark approach for multivariate drift correction methods, as mentioned in [Marco and Gutierrez-Galvez 55].

1.3.5 *Bioinspired signal processing*

Bioinspired engineering systems for chemical sensing is an engaging line of research in machine olfaction. These systems are targeted to mimick biological design and signal processing principles known in living organisms. Mammalian and insect species are well known to regularly perform complex behavioral scenarios based on odor recognition: appetite stimulation, food evaluation, mate recognition, navigation, detection of threats and others [Axel 2]. Models of the bioinspired signal processing are also referred to as *neuromorphic models*. The neuromorphic models are based on the knowledge accumulated in computational neuroscience and have become an active subject of research in processing data from chemosensor arrays [Raman et al. 74, Marco and Gutierrez-Galvez 55].

Very first works in the neuromorphic data processing were especially concerned about 1-of-m classification and sensitivity enhancement [Raman 72]. For example, the massive convergence from ORN to GL was studied as an instance of hyperacuity in [Pearce et al. 65]. Exploiting a large array of optical micro-beads and modeling spike trains of individual ORNs as Poisson processes, the authors showed that an enhancement in sensitivity of $\sqrt{n}$ is possible, where n is the number of convergent ORNs.

Next series of works tackled the issues related to dimensionality reduction, gain control and intensity/quality coding. Most of this research activity was concentrated in the PRISM laboratory at Texas A&M University leaded by R. Gutierrez-Osuna. Here only some contributions of the group will be highlighted, while most of that research of the group is covered in two dissertations [Raman 72] and [Gutierrez-Galvez 27].

In the following three articles, the authors proposed a set of neuromorphic models inspired by the role of the three stages of the olfactory pathway: ORNs and their convergence to GL, periglomerular cells and granule cells. All the proposed models were validated on experimental data from an array of temperature-modulated MOX sensors.

In the first article [Perera et al. 67], Perera and colleagues designed a feature extraction algorithm based on the study of the convergence seen from the population of ORNs to the glomerular layer. The features are grouped in a class-space constructed with information that takes into account the relationship between mean and variance for each feature. The algorithm is computationally efficient under the high-dimensionality of the feature space and well suited for problems with small sample size, since the computation of covariance matrices is not necessary.

In the second article [Raman and Gutierrez-Osuna 73], Raman and Gutierrez-Osuna developed a model based on the first stage of lateral inhibition in the olfactory bulb, which is mediated by periglomerular interneurons. A shunting lateral inhibitory network emulates the role of periglomerular cells following after a self-organizing model of chemotopic convergence proposed earlier [Gutierrez-Osuna 31]. The resulting model was able to remove concentration effects from the multivariate response of an array of chemosensors.

In the third article [Gutierrez-Galvez and Gutierrez-Osuna 28], Gutierrez-Galvez and Gutierrez-Osuna explored the excitatory-inhibitory circuitry of the mitral and glomerular cells. The authors developed a new Hebbian/anti-Hebbian learning rule to increase the separability of sensor-array patterns in a neurodynamics model of the olfactory system. the Hebbian term in the proposed rule is used to build associations within odors and the anti-Hebbian term is used to reduce correlated activity across odors.

The same group of authors studied the advantages of information coding in the first stages of the olfactory pathway: distributed coding with ORNs and chemotopic convergence onto GL untis [Raman et al. 75]. The proposed computational model consists of two parts: a monotonic concentration-response model for mapping of the sensor inputs into a distributed activation pattern, and a self-organizing model of chemotopic convergence. The resulting chemotopic code was shown to improve the signal-to-noise ratio in the sensor inputs while being consistent with results from neurobiology.

An overview of recent advances achieved in the neuromorphic signal processing can be found in [Raman et al. 74]. The authors of this review first describe and discuss the biological design of the olfactory system by covering such issues as the nature of the odor space, the physical domain of space and time organized within the olfactory epithelium, the design of the olfactory sensory neurons, and the computing principles. Second, the authors review recent progress in engineering approaches inspired by biological principles.

In addition to the current *in silico* algorithms described above, it is worth noting the efforts undertaken towards fabrication of neuromorphic chips. Neuromorphic implementations of olfactory networks in silicon are relatively new in the field, and these implementations typically do not yet include the sophisticated algorithms available in computer simulations [Principe et al. 71, Koickal et al. 50, Beyeler et al. 8, Imam et al. 41, Pearce et al. 66]. While a detailed description of the research related to the neuromorphic chips is out of the scope of this manuscript, it would be interesting to give an example of use of neuromorphic networks to perform real-world computing tasks [Neftci et al. 62, Schmuker et al. 84].

Figure 1.6 shows a classifier network designed in [Schmuker et al. 84] in application to the famous iris data set [Fisher 20]. This network approximates the insect olfactory system and consist of three stages: an input layer, a decorrelation layer and association layer, as shown on the Figure 1.6 A. In the scheme of the network AN denotes association neuron, LN denotes local inhibitory neuron, PN denotes projection neuron, and RN denotes receptor neuron.

RNs fire spikes at specified rates which are computed from the real-valued input data using the so called virtual receptors (VRs). VRs play the role of the GL units known in the olfactory pathway. VRs are placed in data space in a self-organized manner using the algorithm in [Martinetz et al. 59]. The Figure 1.6 B shows the projection of the complete iris dataset to the first two principal components (97.7% variance explained) and locations of 10 VRs.

In the training phase, 80% of all data points were presented, and Gorodkin's K-category correlation coefficient R_K was used to measure classification performance. The AN population activity rapidly converged to a representation, and the correct association was established after only a few spikes. The system maintained this state throughout the duration of the stimulus presentation. The neuromorphic classifier was compared to a naive Bayes classifier in 50 repetitions of fivefold cross-validation. The naive Bayes classifier yields an

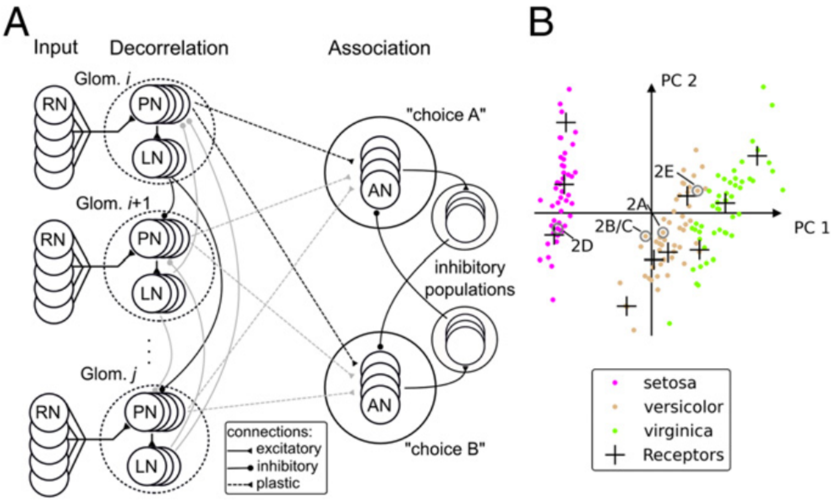


Figure 1.6: An example of a neuromorphic network applied to a real-world classification problem. Figure source: [Schmuker et al. 84].

average R_K of 0.89 (20% and 80% quantiles of 0.88 and 0.90) and slightly outperforms the neuromorphic classifier with R_K = 0.87 (20% and 80% quantiles of 0.85 and 0.89). Based on observations in the confusion class matrix, the authors claim that the neuromorphic classifier network delivers especially reliable classification performance in cases where samples from different classes overlap in feature space.

1.3.6 *Data simulations and benchmarks*

The design of the signal and data processing algorithms requires a validation stage and, thus, some data relevant for the validation procedure. The use of simulated and/or benchmark data is a common practice in many fields, such as computer science, machine learning and statistics. The web site of The University of California at Irvine (UCI) Machine Learning Repository is an example how the machine learning community provides educational resources and open-access benchmarking materials [Bache and Lichman 3]. This repository contains over 290 data sets from different domains, including results from data generators.

The research in the signal and data processing applied to sensor array data has received much attention in the field of machine olfaction in the last three decades [Gutierrez-Osuna 32, Pearce et al. 64, Marco and Gutierrez-Galvez 55]. Data simulations and benchmarks are not widely used in the domain of sensor array data processing, but several examples of such kind of works are worth mentioning here.

The optimum design or configuration of a sensor array is a computationally expensive problem with the need of a large sample size, and simulation-based approach could be an efficient solution in this case. Geng and colleagues considered a problem of selection of the best subset of p sensors for the discrimination task of q gases [Geng et al. 25]. The authors combine a multi-objective tabu search algorithm and a multivariate calibration model fed with the simulated data. The CP sensors explored in the work differ from each other in the polymer material, and the sensor responses are simulated from the analytical model described in [Lei et al. 51]. Two cases with linear and non-linear calibration models are considered, and possible configurations of a sensor array is restricted by the condition $p = q$. Thus, the resulting matrix A is quadratic of size $p \times p$, and two measures of independence and semangularity are calculated from A and are further incorporated in sensor selection criteria. The proposed optimization algorithm is compared with the exhaustive search, and the algorithm efficiently approaches the best array configuration.

Fonollosa and colleagues developed a methodology to select the optimal operating temperature of the MOX sensors in an array based on the multivariate response of the sensors. The authors employed the mutual information (MI) measure, in order to quantify the amount of information that the multivariate response can provide from a variable representing the quality of the measured gas. The optimization procedure is computationally intensive, and the computation time increases exponentially when a new sensor is added to the array. The sensor data are simulated from Clifford–Tuma model [Clifford and Tuma 13, 14], that allows to overcome the major difficulty of estimating the joint probability distribution of the events needed for entropy calculations. The measurements from four different MOX sensors (TGS-2620 and TGS-2600 by Figaro Inc., and SB-15-00 and SB-11-00 by FIS Inc.) are fitted to the Clifford–Tuma model, and the model parameters are estimated for ethanol, acetic acid, 2-butanone, and acetone in the range of 0.1–1000 ppm and for 95 operating temperatures. The authors conducted intensive data simulations generating 5000 data points per gas and considering all combinations of 94 possible temperatures. The proposed methodology shows, for instance, that the classification ability of the sensor array increases when passing from two-sensor to four-sensor arrays, but combinations of the optimal temperatures are different for each case.

Another use case of the data simulations is related to the validation of methods on noise compensation. Marco and colleagues studied the effect on the long-term drift on the pattern recognition unit based on self-organizing maps (SOM) [Marco et al. 57]. The authors induce a synthetic drift noise into the measured short-term data, which were collected from an array of 6 MOX sensors in response to 4 pure analytes and 2 mixtures. As the measured sensor signal is conductance G, the drift noise is injected into the data by following the equation: $G(t) = G_0(1 + \alpha t)$, where G_0 is the sensor signal before the drift injection, and α is a randomly selected value per sensor within the interval $(-0.4, 0.4)$. The extreme values of α correspond to the four year's drift according to the documentation from the manufacture. The study shows a decay of the SOM robustness to the simulated drift and propose the adaptation mechanism in SOMs for better counteraction to the drift.

The use of bioinspired synthetic data is a necessary step in particular neuromorphic simulations. The work presented above [Raman et al. 75] makes use of a model of population coding with broadly tuned olfactory receptor neurons and chemotopic convergence onto glomerular units. A particular receptor is modeled by a vector of log-affinities towards the different analytes in chemical space, and another model of a monotonic concentration-response curve is incorporated. Chemosensor array data are transformed onto firing rates of the olfactory receptors by a nonlinear mapping defined by the proposed population model of the receptors. The rationale behind the mapping is that the tested neuromorphic models need to operate with a similar representation to that available in the olfactory epithelium within ORNs (combinatorial and high-dimensional).

The practice of benchmarking of signal processing, pattern recognition or neuromorphic algorithms has been established relatively recently in the field of machine olfaction, although Gutierrez-Osuna declared the need of a repository of benchmarks in 2002 [Gutierrez-Osuna 32]. The research group in The University of California San Diego (UCSD) led by Ramon Huerta was the first in publishing the collected data sets. To date, there are three data sets published by the group on the web site of The University of California at Irvine (UCI) Machine Learning Repository [Bache and Lichman 3]:

1. Gas Sensor Array Drift Dataset Data Set [Vergara et al. 95], https://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset;

2. Gas Sensor Array Drift Dataset at Different Concentrations Data Set [Vergara et al. 95, Rodriguez-Lujan et al. 79], http://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset+at+Different+Concentrations

3. Gas sensor arrays in open sampling settings Data Set [Vergara et al. 94], http://archive.ics.uci.edu/ml/datasets/Gas+sensor+arrays+in+open+sampling+settings.

Alexander Vergara was the main creator of the three collected data sets, and the first data set has already been used, for example, in [Wang et al. 100, Tang et al. 89, Liu et al. 52].

In conclusion, simulated data and public benchmarks can be viewed as a complementary block in the signal and data processing chain in machine olfaction, especially in the case when an accurate and extensive validation of algorithms is required. In order to produce simulated data, the behavior of gas sensors in an array has to be described with appropriate models, which take into account a particular chemical odor space of interest and particular environmental conditions. The elaboration of a benchmark data set depends on a scenario of interest, which has to be relevant for a given pattern recognition problem.

2 GOALS

Goals of the study

This work deals with the development of a biomimetic set up, that will be able to mimic to some extent the operation of the olfactory system on a particular list of scenarios. The findings established during the development and experimental stages of the study are further used to address open issues in the field of machine olfaction. More precisely, the present work is designed to achieve the following goals:

1. To assemble a host embedded computer, that performs main functions of the set up: (1) interface with chemical and navigation sensors, peripheral and other devices; (2) reliable data acquisition from chemical sensor arrays; (3) embedded computations including the neuromorphic data processing; (4) real-time visualization capabilities on the display; and (5) development of the operating system. The design of the embedded computer is targeted to fulfill two major requirements towards a novel biomimetic architecture for chemical sensing:

   - interface with a unique large-scale array of 2^{16} (65,536) sensing elements that was designed (by collaborators) to mimic the high degree of redundancy in the population of olfactory receptro neuors;
   - design of a modular software environment able to to run a complete realistic (simplified) model of olfactory system within odor localization behavioral models in real time.

   The software for the embedded computer is developed in three main blocks: (1) design of a software development kit (SDK) to host the models; (2) integration of neuromorphic, robotic and other models into a unified framework; and (3) development of data simulation models to mimic a sensor array (the virtual array). Such design allows a virtualization of the set up, that means creating an ecosystem containing both the neuromorphic models for data processing and the virtual array for data generation. This ecosystem can be loaded from a system image on any personal computer.

2. To conduct bioinspired experiments on the assembled set up, that assumes the following steps: (1) design of scenarios appropriate to test the biomimetic artifacts of the set up; (2) collection of the relevant data sets; (3) running demonstrations of the set up; and (4) data analysis based on the standard pattern recognition methods and/or bioinspired algorithms. As the experiments could run into bottleneck due to the complexity of the design of the main sensor array of 65,536 sensing elements, two alternatives to the array are explored: (a) a smaller array of commercially available metal oxide sensors and (b) synthetic experiments based on virtual arrays. The collected experimental data sets are further used:

   - to validate the performance on the novel models of the olfactory system under synthetic and real-world conditions;
   - to perform the same simulations by employing either of the three arrays of gas sensors;

- to conduct the navigation experiments with the mobile robotic platform;
- to explore novel experimental conditions under a gas flow modulation.

3. To tackle open issues in the data analysis domain in machine olfaction with a special emphasis on creating a data simulation tool. The particular tasks are:
    - to develop a more realistic drift model based on the analysis of common variance among several groups;
    - to develop simulation models able to provide plausible synthetic sensor array data in application to algorithm testing;
    - to generate synthetic benchmarks for machine olfaction scenarios of interest;
    - to analyze the collected data under the gas flow modulation with a focus on early detection scenario.

Workflow of the study

Figure 2.1 show the workflow of the thesis. The boxes represent the work packages that needed to be accomplished in the course of the thesis, and the arrows show the dependencies among the work packages.

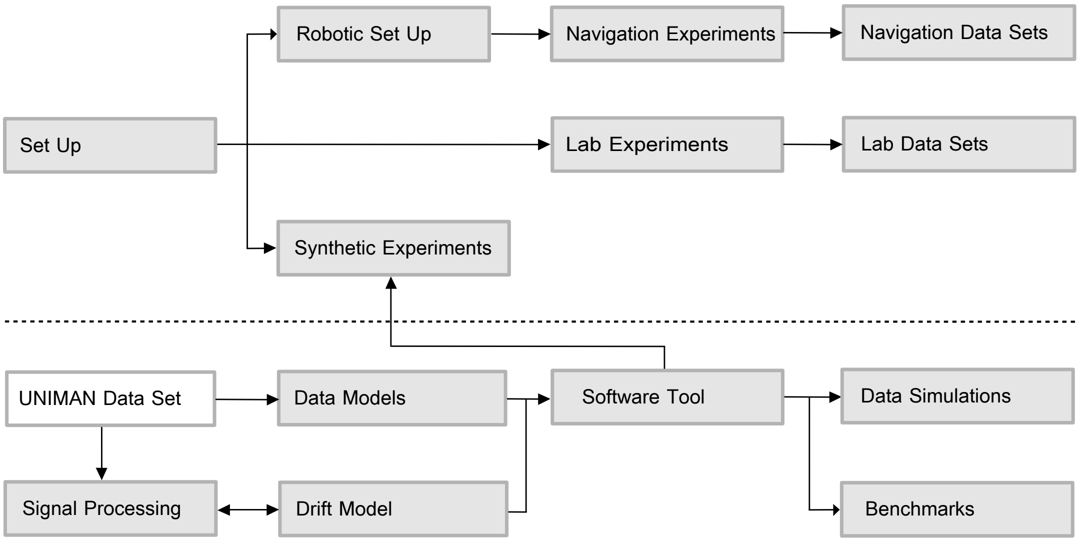


Figure 2.1: The scheme of the thesis workflow. The details are given in the main text of the manuscript.

The workflow on the Figure 2.1 is divided into two upper and lower sides, that correspond to two different research directions in the development of the set up. The work packages on the upper side were related to the design of the embedded computer (*Set Up* box on the Figure) – the core part of the set up – and its integration with the mobile robotic platform produced by collaborators (*Robotic Set Up*). Synthetic experiments were used in testing of the set up (*Synthetic Experiments*), while the navigation and laboratory experiments were carried out once the set up had been assembled (*Navigation Experiments* and *Lab Experiments*). Acquired data sets were the final results of the experiments (*Navigation Data Sets* and *Lab Data Sets*).

The work packages on the lower side of the Figure 2.1 were needed to develop the data simulation tool (*Software Tool* box on the Figure) – one of the key software components of the set up. Data simulation models, including the drift model, were developed to create a virtual sensor array for data generation (*Data Models* and *Drift Model*). A reference data set provided by Prof. Krishna Persaud from The University of Manchester (UNIMAN) was used to validate the models (*UNIMAN Data Set*). Validation of the drift model also included the design of a chain of singal-processing and pattern recognition methods, where the task was a drift compensation in the sensor array data (*Signal Processing*). Parametrized data simulations and synthetic benchmarks were the final results produced by the data simulation tool (*Data Simulations* and *Benchmarks*).

3 RESULTS

3.1 SUMMARY OF THE RESULTS

The biomimetic set up proposed in the thesis has been developed to complete the primary task of conducting the bioinspired experiments of the project. The main engineering challenge in the design of the set up was to fulfill the requirements for this kind of experiments: acquire the large-scale sensory inputs, run an embedded neuromorphic signal processing chain in real time, and resolve complex olfaction scenarios.

Virtualization of the experimental set up was one of the key engineering solutions in the development. Being a device, the set up is transformed to a tool for running the synthetic experiments, where the software environment is essentially the same, while the virtual sensors from the data simulation tool replace the real sensors.

Data models designed to emulate the sensor array were able to realistically mimic the behavior of the sensors, that in turn supports both large-scale neuromorphic synthetic experiments and conventional pattern recognition simulations in machine olfaction. The developed model of the drift – originally proposed for the signal-processing task on the drift compensation – has also been successfully incorporated in the group of models for simulation of sensor array data.

Two main engineering products have been resulted from the thesis. First, the autonomous robotic set up featured with large-scale chemosensor array and the embedded data processing units has been assembled and used to conduct bioinspired experiments and to collect data sets. Second, the data simulation tool has been released to enable the use of synthetic data in testing the experimental set up and/or data processing methods.

The scientific results have been achieved in both production and post-production stages in the design of the two engineering products of the thesis. Results of the thesis are grouped into five categories: journal articles, other publications (mainly book chapters), collected data sets, demonstrations of the set up and released software. Results for each category are visually represented on the scheme of the workflow of the thesis, as presented on Figure 3.1.

Main results of the thesis are presented further in the given copies of three journal articles and one book chapter. The book chapter contains a description of hardware and software of the assembled set up, that corresponds to the first goal of the thesis – assembling of the set up. More technical results such as demonstrations, data sets and software programs are related to the second goal of the thesis – conducting bioinspired experiments, and these results are evenly distributed among the work packages of the thesis, as shown on Figure 3.1. All the three published journal articles cover the results related to the third goal of the thesis declared in Chapter 2 – the data analysis with emphasis on creating the data simulation tool. The three copies of the journal articles are presented in Sections 3.2, 3.4 and 3.5, and the copy of one book chapter is provided in Section 3.3.

3.1.1 *Journal articles*

Main results of the thesis have been reported in three journal articles, which enabled to present the thesis as a collection of published articles.

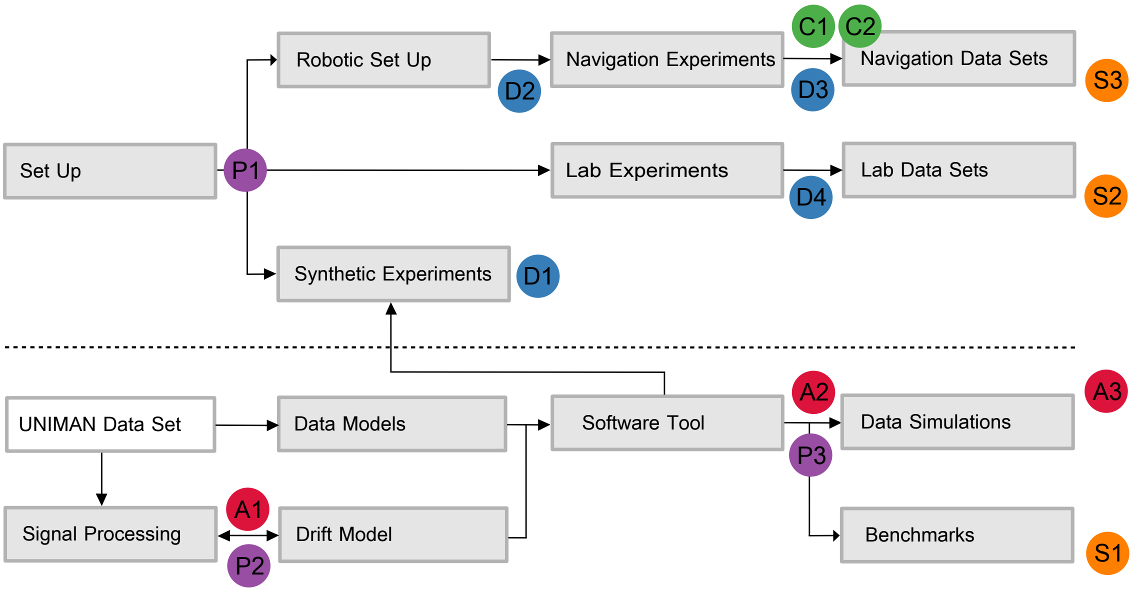


Figure 3.1: The scheme of the thesis workflow overlaid with the thesis results depicted as circles. The results are grouped into five categories: publications (red circles with the letter *P*), book chapters (green circles with the letter *C*), collected data sets (orange circles with the letter *S*), demonstrations of the set up (blue circles with the letter *D*) and released software programs (purple circles with the letter *P*). The details on each of the results are given in the main text of the manuscript.

- Ziyatdinov et al. (2010) [109] (Section 3.2)
- Ziyatdinov et al. (2013b) [108] (Section 3.4)
- Ziyatdinov and Perera-Lluna (2014) [110] (Section 3.5)

The importance of the research conducted is demonstrated by the quality of the journals.

SENSORS AND ACTUATORS B: CHEMICAL is an interdisciplinary journal publishing original peer-reviewed research articles in all aspects of research and development in chemical sensors, actuators and microsystems. Many research works in the area of data analysis applied to chemosensor array data are traditionally published in this journal. It is indexed in Journal Citation Report (JCR) for 2012 with a current impact factor 3.535 and classified in the 1st quartile of the areas *Analytical Chemistry* (ranking: 11/75) and *Instrument & Instrumentation* (ranking 2/57). It is also in the 2nd quartile of the area *Electrochemistry* (ranking: 8/26).

PLOS ONE is an open access journal publishing original research articles from all disciplines within science and medicine. This journal allows for the discovery of the connections between papers whether within or between disciplines. It is indexed in Journal Citation Report (JCR) for 2012 with a current impact factor 3.730 and classified in the 1st quartile of the area *Multidisciplinary Sciences* (ranking: 7/56).

A short summary for each article is presented below given in the chronological order. The three copies of the manuscripts are presented next in Sections 3.2, 3.4 and 3.5.

[Ziyatdinov et al. 109], published in the journal *Sensors and Actuators B: Chemical* in 2010, proposes a novel method for correction of the drift noise observed in the long-term chemosensor array data. The proposed method belongs to the class of multivariate correction methods, and, thus, is compared with one of the state of the art method of the same class. The method presented in this article is based on common principal component analysis, that allows to overcome the necessity of a reference gas.

[Ziyatdinov et al. 106], published in the journal *Sensors and Actuators B: Chemical* in 2013, introduces the data simulation software tool for synthetic experiments in machine olfaction. This articles describes the reference data set and the data generation models employed to create a virtual chemosensor array. The reported results demonstrate the ability of the data simulation tool to reproduce the reference data set and further extend these reference data in terms of the number of sensors, multicomponent gas mixtures and the amount of noise in the virtual array.

[Ziyatdinov and Perera-Lluna 110], published in the journal *PLoS One* in 2014, describes the simulation workflow that the user of the data generation tool is suggested to follow. The results of this articles show examples of the processing of the simulated data as a proof of concept of the parametrized chemosensor array data. These examples include the benchmarking of classification algorithms, the evaluation of linear- and non-linear regression algorithms, and the biologically inspired processing.

3.1.2 *Other publications*

The following list presents the conference proceedings related to the thesis.

- Ziyatdinov et al. (2009) [105]
- Perera and Ziyatdinov (2011) [68]
- Ziyatdinov et al. (2011a) [104]
- Ziyatdinov et al. (2011b) [107]
- Ziyatdinov et al. (2013a) [106]

Two book chapters were published in the course of the thesis. Both chapters were a joint work mostly between the UPC and the UPF partners in the Neurochem project. Contributions presented in the chapters and related to the thesis mainly include two activities: implementation of the robotic platform and conducting the navigation experiments. The copy of one book chapter [López et al. 53] is presented in Section 3.3.

- López et al. (2011) [53] (Section 3.3)
- Vouloutsi et al. (2013) [98]

[López et al. 53] is a chapter of the book *On Biomimetics* published in open access by *InTech* in 2001. The book covers the research and construction of biomimetic systems, and the chapter presents the moth-like approach to the chemical source localization problem tested on an indoor mobile robot in the framework of the Neurochem project. *InTech* is a world's largest multidisciplinary open access publisher of books covering the fields of Science, Technology and Medicine. The book falls into the fields of Medicine, Tissue Engineering and Regenerative Medicine http://www.intechopen.com/books/on-biomimetics.

[Vouloutsi et al. 98] is a chapter of the book *Neuromorphic Olfaction* published by *CRC Press* in 2012. The book reports the interdisciplinary biomimetic results in biology, hardware, software and sensors' technology achieved at the end of the Neurochem project.

The chapter presents the latest advances in mobile olfaction robotics from a biomimetic perspective. *CRC Press* is a premier publisher of scientific, technical and medical content, including world-class references, handbooks and textbooks. The book belongs to the series of Frontiers in Neuroengineering Series.

3.1.3 *Data sets*

Three data sets have been collected in the course of the thesis. These three data sets are shown as orange circles with the letter *S* on Figure 3.1: S1, S2 and S3. A short summary for each data set is outlined below, and a more detailed description for each data set is given in Appendix A.

s1 benchmarks A virtual array of 1020 sensors was created by the data simulation tool, in order to produce a collection of synthetic benchmark data sets. Each data set corresponds to a specific machine olfaction scenario, defined at 5 difficulty levels. The primary use of the generated data was testing the neuromorphic models designed in the scope of the NEUROChem project, while the real sensor devices of the project were under development. These data can be valuable in other projects in machine olfaction, where large-scale parametrized prototype data are required. The data sets are now can be publicly accessed on the following link http://neurochem.sisbio.recerca.upc.edu?page_id=257.

s2 laboratory data sets A custom experimental set up was designed in the laboratory conditions, in order to emulate a sniffing behavior (sampling odors actively) known in the olfactory system of the mammals. The collected transient signals, recorded from an array of 16 metal-oxide sensors under the gas flow modulation, showed two low-frequency and high-frequency (modulated by the respiration cycle) parts of the spectrum. The analysis of the data set is an on-going work of the thesis, while the preliminary results have been reported as a conference proceeding [Ziyatdinov et al. 106]. As a data-sharing initiative for the machine olfaction community, the data set is now publicly available on the web site of The University of California at Irvine (UCI) Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/Gas+senso
This public data set will allow to continue the joint research of biologically inspired chemical systems.

s3 navigation data sets The robotic set up in the navigation experiment explored the chemical odor space of the wind tunnel with one or two odor sources. The aim of the experiment was to reconstruct the odor map (segment the mixture of odors spatially) by means of signals recorded from a sensor array. Two alternative sensor arrays were involved in the experiment: an array of 16 metal-oxide sensors and an array of 4096 conducting polymer sensors. The reference measurement of the odor map was accomplished by an ion-mobility spectrometry (IMS) device, that had a higher resolution to quantify the chemicals in comparison with either of two arrays. Analysis of the collected data is thought to be done in the future, while some preliminary multivariate analysis of these data has been accomplished (unpublished results). A similar analysis based on independent component analysis (ICA) on a similar data set (array of only three metal-oxide sensors) measured in the same experimental conditions was published in the course of the thesis as a conference proceeding [Ziyatdinov et al. 104].

3.1.4 *Demonstrations*

Four demonstrations have been designed in the course of the thesis, in order to show a wide range of functional abilities of the biomimetic set up under certain experimental conditions. These demonstrations are shown on Figure 3.1 as blue circles with the letter D : D1, D2, D3 and D4. A short description for each demonstration is presented below, while a more detailed information is provided in Appendix B.

D1 IQR This demonstration shows how a complete olfactory system of the insect works on the simulated data from 100 virtual sensors. Two main models of AL and MB display some of the olfactory system properties such as the classification of presented two odors A and C.

D2 ROBOTIC PLATFORM This demonstration shows the mobile robotic platform of the set up in action, moving around the wind tunnel. The robot demonstrated the functionality of its multimodal sensing capabilities, such as compass, wind sensing, ultrasound, collision detection and vision. The chemosensors needed for the navigation task were not integrated to the robot yet.

D3 LARGE-SCALE ARRAY This demonstration shows the capability of the set up to acquire the large-scale input from the CP array of 4096 elements at 1 Hz acquisition frequency. The sensor signals were displayed in the real time in the IQR simulator, demonstrating a high level of diversity in the data.

D4 NAVIGATION This demonstration shows the robotic set up in resolving the navigation scenario targeted to discriminate between two odor sources in the chemical space of the wind tunnel. The mobile platform was updated in comparison to the D1 demonstration. The modules of the olfactory systems in IQR were also updated with the final neuromorphic models designed in the Neurochem project. The robot showed a particular behavior on attraction to one of the two odor sources with the success rate of 70% and 90% for ammonia and ethanol vapours, respectively.

All the demonstrations in the format of an IQR system [Bernardet et al. 7] are available on the Neurochem image (Appendix C).

3.1.5 *Software*

Three software products have been developed in the course of the thesis. These are shown as purple circles with the letter P on Figure 3.1: P1, P2 and P3. The first software product is the Neurochem image, and other software programs are two packages designed for the R environment for statistical computing.

More information on the the Neurochem image is available in Appendix C and on the web page of Neurochem project. The R packages are distributed in the official repository for the R packages *CRAN* http://cran.r-project.org/. A short description for each software and the links to the web pages are given below.

P1 NEUROCHEM IMAGE http://neurochem.sisbio.recerca.upc.edu/?page_id=54 The image is a custom Debian-based operating system image that includes software components required to run the embedded set up. All the necessary components such as drivers, software packages, and IQR models designed in scope of the Neurochem projects are included in the image file *neurochem.img* of 563 MB size. The image can be run either on the embedded set up or on any operating system emulator.

P2 R PACKAGE CPCA http://cran.r-project.org/package=cpca The package aims to implement statistical methods to perform the common principal component analysis. The current version of the *cpca* package 0.1.2 contains the only stepwise method, and this implementation was completed in the course of the thesis. Further development of the *cpca* package and implementation of other methods will be a collaborative project of authors of the package.

P3 R PACKAGE CHEMOSENSORS http://cran.r-project.org/package=chemosensors The package is an implementation of the data simulation tool released in the course of the thesis. Two of the journal articles published in the course of the thesis [Ziyatdinov et al. 108, Ziyatdinov and Perera-Lluna 110] are the primary references of this software.

3.1.6 *Collaborative results*

Two groups of results were obtained in collaboration with other members of the research group of the author. The first group of results is related to application of the drift correction method, which is proposed in the thesis, to data sets from other fields of science. An example of such an application is the Liquid Chromatography coupled to Mass Spectrometry (LC/MS) data in metabolomics:

- Fernández-Albert et al. (2014) [19]

The second group of collaborative results tackles the problem of joint clustering, where the data come from different views or representations. These data views are formalized by similarity matrices among the observations. The framework of the spectral clustering is extended such that several similarity matrices are diagonalized simultaneously. One of the steps of the algorithm is joint diagonalization of the matrices that can be accomplished by means of the common principal component analysis. Two works, the conference proceeding and the patent proposal, have been derived:

- Kanaan-Izquierdo et al. (2012) [44]
- Kanaan-Izquierdo et al. (2013) [45]

3.2 results 1. drift model – ziyatdinov et al., 2010

Ziyatdinov, A., Marco, S., Chaudry, A., Persaud, K., Caminal, P., & Perera, A. (2010). Drift compensation of gas sensor array data by common principal component analysis. Sensors and Actuators B: Chemical, 146(2), 460–465. doi:10.1016/j.snb.2009.11.034 [109]

ATTENTION ¡

Pages 32 to 37 of the thesis are availables at the editor's web
http://www.sciencedirect.com/science/article/pii/S0925400509008995

3.3 Results 2. Hardware and software description – Lopez et al., 2011

López, L., Vouloutsi, V., Chimeno Escudero, A., Marcos, E., Bermúdez i Badia, S., Mathews, Z., Verschure, P. F.M.J., **Ziyatdinov, A.** & Perera i Lluna, A. (2011). Moth-Like Chemo-Source Localization and Classification on an Indoor Autonomous Robot. In L. D. Pramatarova (Ed.), On Biomimetics. InTech. [53]

21

Moth-Like Chemo-Source Localization and Classification on an Indoor Autonomous Robot

Lucas L. López et al.*
SPECS, Technology Department, Universitat Pompeu Fabra and ICREA
Spain

1. Introduction

Olfaction is a crucial sense for many living organisms. Many animals, especially insects, rely heavily on the olfactory sense for encoding and processing different chemical cues in order to perform several tasks such as foraging, predator avoidance, mate finding, communication etc.(22). Yet, olfaction has not been as widely studied as vision or the auditory system in insects. At the same time, robotic platforms capable of searching, locating and classifying odor sources in wind turbulence and in the presence of complex odors have diverse applications ranging from environmental monitoring (21), detection of explosives and other hazardous substances (19), land mine detection (2) to human search and rescue operations. The main challenge thereby is the stable and fast coding and decoding of odors and the localization of the sources (17).

In our own recent work, we have proposed an insect-like mapless navigation mechanism which integrates surge-and-cast chemo search, path integration, wind detection and visual landmark navigation on an indoor mobile robot (28). Also, we have proposed a model based on insect navigation that is capable of navigating in highly dynamic environments and our model was compared directly to ant navigational data, with strikingly similar navigational behaviors (26). The problem of ambiguous information, particularly in the navigational context, is also addressed in our recent work (27). Beyond that, we have contributed significantly to modeling insect navigation and designing robotic systems such as: a model of the locust Lobula Giant Movement Detector (LGMD) tested on a high speed robot (29), moth-like odor localization for robots (30), control of an unmanned aerial vehicle using a neuronal model of a fly-locust brain (31; 32), moth-like optomotor anemotactic chemical search for robots (33), and a blimp flight control using a biologically inspired flight control system (34).

Despite these advances, several biological systems with relatively simple nervous systems solve the odor localization and classification problem much more efficiently than their artificial counterparts: bees use odor to localize nests, ants use pheromone trails to organize foraging in swarms, lobsters use odor to locate food, the *Escherichia* bacteria use odors to locate nutrients, male moths use olfaction to locate female mates etc. The odor localization

*Vasiliki Vouloutsi, Alex Escuredo Chimeno, Encarni Marcos, Sergi Bermúdez i Badia, Zenon Mathews, Paul F.M.J. Verschure (*SPECS, Technology Department, Universitat Pompeu Fabra and ICREA, Barcelona*), Andrey Ziyatdinov, Alexandre Perera i Lluna (*Departament d'Enginyeria de Sistemes, Automàtica i Informatica Industrial, Universitat Politècnica de Catalunya and CIBER-BBN in Bioengineering, Biomedicine and Nanomaterials, Barcelona*) task can be divided into three general steps (9): 1) search and identification of the chemical compounds of interest in the given environment, 2) tracking the odor until its source guided by chemical and all other available sensory modalities, 3) and finally identifying the source (either by vision or e.g. by olfaction using the odor concentration pattern that is acquired in a specific restricted area). However, in real world applications, locating the source of a chemical plume and classifying the chemical are difficult tasks due to the fact that the plume dispersion dynamics vary heavily depending on the medium. The chemical volatiles in the atmosphere are mainly transported by airflow and the interaction of the airflow with other surfaces and sources of thermal gradients produce turbulence. This chemical dispersion is best described by the Reynolds number. At low Reynolds values, there is a monotonic decrease of the chemical concentration, however at medium and high values turbulence dominates. Thus different search and classification strategies should be employed in these different environments (9).

The rich availability of insect odor coding and localization studies have inspired several biologically inspired robots that perform odor localization and classification: underwater robots (6), ground robots (14) and even flying robots (2). Nevertheless, stable odor source localization and classification using fully autonomous robots have not yet been demonstrated. We here propose a moth based model of odor localization and classification and its implementation on an embedded autonomous robot in a controlled indoor wind tunnel setup. For odor coding and localization at high Reynolds values where turbulence prevails, we use a model of odor source localization and odor classification mechanism suggested to be employed by the male moth. Our embedded robot is controlled using a neural network model of the moth olfactory pathway implemented using the large scale neuronal simulator IQR (4), that runs on board the embedded robot. Our results show the first steps towards stable odor localization and classification using a completely autonomous robot that is controlled by a neuronal model of the moth olfactory system.

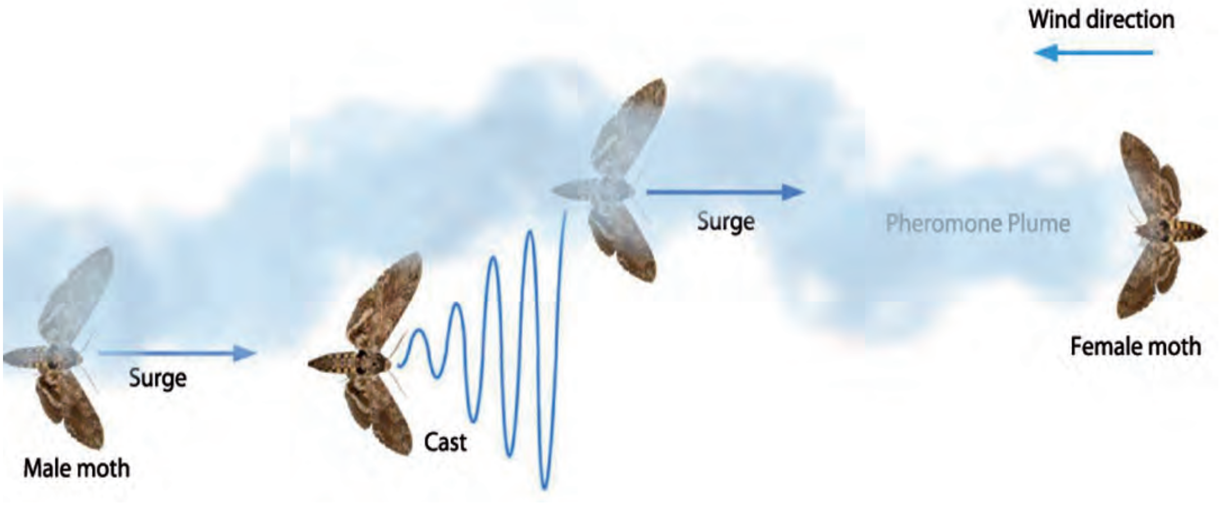


Fig. 1. Illustration of the cast and surge male moth behaviour and the female pheromone plume.

2. Methods

Insects in general and moths in particular are able to locate a source of odor and distinguish it from different other sources. Our model of olfaction is based on the male moth behavior and physiology. In this section we explain our olfactory model proposed for solving the problem of odor localization and classification, the robot platform and the experimental set-up used to assess its performance.

2.1 Cast & Surge

The male moth has been widely studied because of its unique ability to find mates by detecting low pheromone concentrations over large spatial scales. When the female moth releases a pheromone blend, this blend flows downwind creating a specific plume shape. When the male moth detects the pheromone plume, it starts flying upwind, tracing the pheromone molecules in the plume, a stereotypical behavior called *surge*. However, as the structure of the plume is quite complex and unpredictable, the male moth looses track of the pheromone plume often during the surge behavior. For this reason, the male moths have developed a behavior that allows them to re-discover the pheromone plume again. This behavior is called *cast* and is a zigzag movement orthogonal to the wind direction (17) (see Figure 1). The casting frequency increases and the speed decreases when close to the source (10).

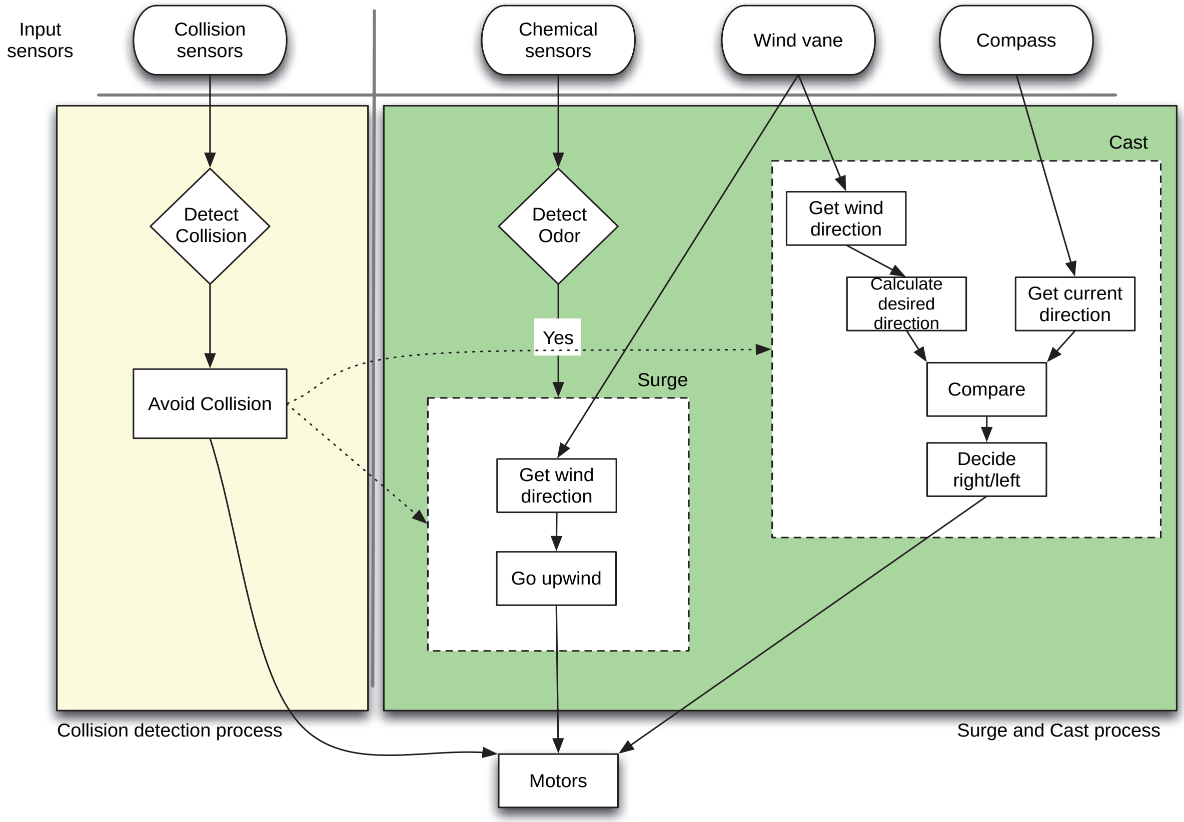


Fig. 2. Scheme of the system implemented for the cast and surge behaviour. It consists of two processes: *collision detection* and *surge and cast*. Dashed arrows indicate inhibitory influences.

Our model of odor localization is based on this cast and surge behavior of the male moth. The architecture of the system consists of two process that run in parallel: *collision detection* and *surge and cast* (see figure 2). The *collision detection process* has higher priority and inhibits dashed arrow in figure 2) the *surge and cast* process. The *surge and cast* process performs the localization of the odor source. When the chemical sensors detect an odor the robot performs a surge behaviour, and otherwise a cast is executed. The processes are implemented using leaky Integrate and Fire (IF) and leaky Linear Threshold (LT) neurons (16; 20).

2.2 Classification

While being able to locate an odor source, the male moths are also able to distinguish among similar stimuli and to classify different concentrations of the same chemical into the same stimulus category. The olfactory pathway is composed of Olfactory Receptor Neurons (ORNs) in the antenna, the Antennal Lobe (AL) and the Mushroom Body (MB) (7) (see Figure 3). ORNs are distributed over the antenna and respond to different chemical stimulus present in the air. ORNs expressing similar receptors usually converge onto a single glomerulus in the AL. The number of glomeruli is then closely related to the number of ORN classes. This convergence of ORNs into the same glomeruli makes the AL capable of dealing with noisy conditions and dynamic inputs (11).

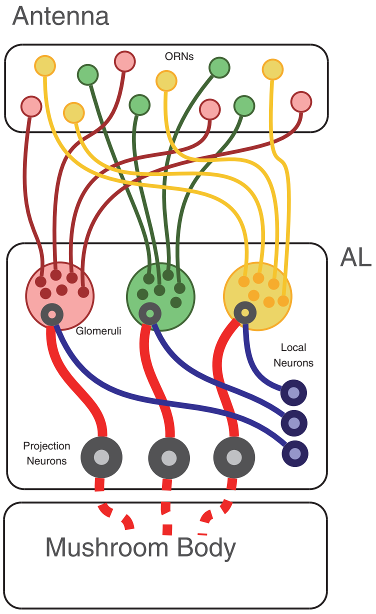


Fig. 3. Functional representation of a generic AL. ORNs belonging to the same class converge onto the same glomerulus. LNs interconnect PNs which is connected to higher brain areas such as the MB.

Two different types of neurons receive input from ORNs: Projection Neurons (PNs) and Local Neurons (LNs). PNs integrate the activity from the glomeruli and forward it to the MB, which is known to be involved in the learning and memory of odors (24). LNs laterally interconnect PNs and modify their activity by means of inhibition.

We use a modified implementation of the model proposed by (15). The original model uses a group of Integrate-and-Fire neurons as Projection Neurons, which receive constant excitation, interconnected with two groups of Local Neurons. These LNs are connected in such a way that when a specific pattern is presented to the network, concrete PNs will fire synchronously. When the pattern disappears from the input, the neurons get desynchronized. These synchronization and desynchronization processes can be explained with two concepts: a combination of transient resetting and the probability of failure of synapses between the Local Neurons and the Projection Neurons. Transient Resetting has been theoretically described by (13) as a way to enhance the spike timing precision on a group of neurons, caused by a loss of initial conditions. In the presented model the current pulse coming from the LNs to the PNs allow the latter to turn from their state to their resting potential, which makes the next spike to happen simultaneously. The presence of noise in the connection between LNs and PNs has an essential role in the network equilibrium. LNs interconnect PNs in two different ways: via fast ($GABA_A$ type) and slow ($GABA_B$ type) inhibition. The failure of these synapses has been set to 50%. The key concept is that when fast inhibition is not greatly affected by the failure of a connection and is still able to produce the transient resetting, the slow inhibition is much more sensitive and has the opposite effect, generating noise in the inter-neuron spike timing.

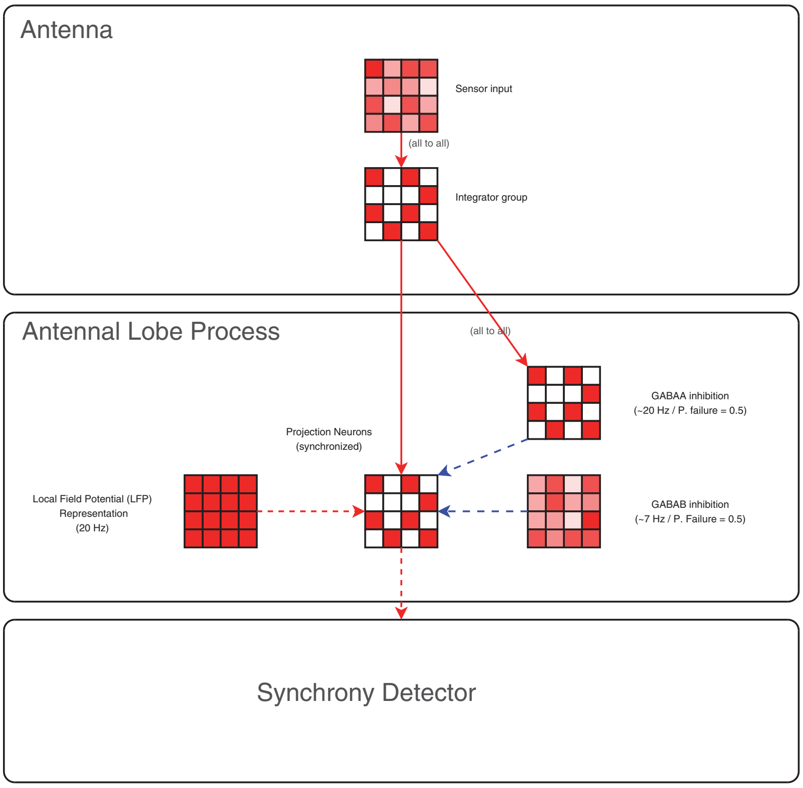


Fig. 4. Scheme of the system implemented for the odor classification. Red arrows indicate excitatory connections and blue arrows indicate inhibitory connections.

This model proposed by(15) was designed to receive only binary input patterns. The model needed to be adapted for real world conditions where the sensory input is analog. Based on the modification proposed by (12), we use a group of neurons that process the input from the sensors to extract a binary pattern that is later fed into the AL model. The numeric parameters from the original model has been respected as much as possible in order to obtain similar results. Fast $GABA_A$ inhibitions oscillate around $20Hz$, while $GABA_B$ frequency is around $8Hz$. The interconnection topology between PNs and LNs also respect the original setup: if the PN responds to the odor stimuli, it has $GABA_A$ and $GABA_B$ inhibitory interconnections, whereas if it does not respond to the odor stimuli, it has only $GABA_B$ inhibitory interconnections. Figure 4 shows a scheme of the system.

2.3 The robot

2.3.1 Robotic platform

The autonomous robot used for the experiments is composed of two parts, a mobile platform developed in SPECS at UPF and an embedded computer assembled at UPC, both designed in the scope of the Bio-ICT European project NEUROChem (Figure 5). The basic requirements applied to the robot include full autonomy, demonstration capabilities and full-functioning interface with chemical and other navigation sensors.

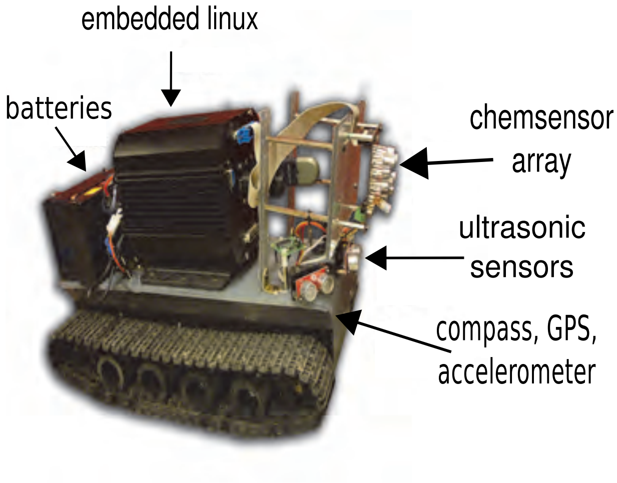


Fig. 5. Image of the autonomous robotic platform.

The mobile platform is driven by motors acting on two caterpillar tracks on both sides, and holds different sensory electronics for robot navigation such as ultra sonic distance sensors, compass, GPS and accelerometer. The mobile base is interconnected with the embedded platform either via Bluetooth or by the USB cable.

The embedded computer performs functions of a host platform targeted to high-performance sensor data acquisition as presented on Figure 6. The use of the embedded technology for the moth robot is motivated by several factors. The embedded computer runs a custom GNU/Linux image to control the complete robotic system with the aid of the standard desktop solutions. Moreover, the computational resources are needed for the real-time acquisition, processing and visualization of the sensory data coming from the real world, and especially for capturing the chemical stimuli. Moreover, the execution of the biomimetic models of the antennal lobe and the mushroom body requires a solid software framework hosted on the computer.

The success of the odor localization task highly depends on the instrumentation capabilities of the robot for odor sensing, that is traditionally based on an array of broadly-selective gas sensors (18). The robot design allows to host three types of the gas sensor arrays providing specific hardware interfaces, scanning electronic boards and signal processing software.

The main large-scale array contains 64K polymeric sensors (16 modules of 64×64 sensing elements each) and around 8 of sensor types (1). The critical parameter is the acquisition speed of a sensor, which is determined by dynamics of the chemical reactions in sensor device and limited by transient constants of the read-out electronic circuit (proportional to parasitic capacitances). The preliminary experiments (1) showed the sampling rate of $\approx 293\ \mu_s$ for a sensor. Due to the modular structure of both the sensor array and the acquisition boards, the acquisition speed for the complete number of sensors (64 K) expected to be close to 1.8 s. That seems reasonable to perform the real-time robot experiments.

The preliminary results presented in this work are obtained with the second sensor array, as the main polymeric array is still in the development phase. The current array is composed of 16 MOX sensors of 4 Figaro (Figaro Engineering Inc) types (TGS 2442, TGS 2612, TGS 2610 and TGS 2600). The third array supported by the platform, referred as to virtual sensor array (25),

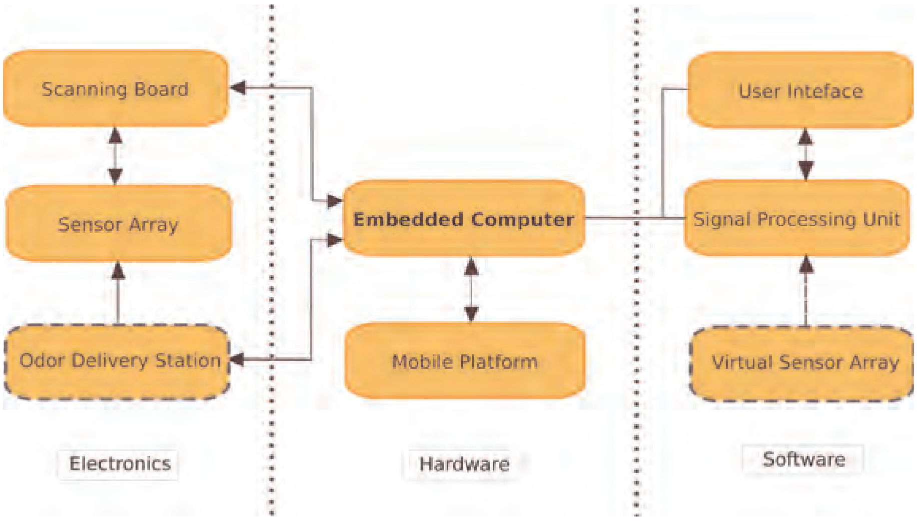


Fig. 6. Architecture of the robotic platform.

represents a software abstraction of sensor signals used during testing the insect olfactory models.

2.3.2 PC104-based embedded computer

The architecture of the embedded computer is based on the well-established PC104 standard that was originally proposed as an extension of the IEEE-P996 standard (Standard for Compact Embedded - PC Modules). PC104 systems are typically industrial rugged embedded applications where reliable data acquisition is needed in an extreme environment.

The key features of the PC104 bus in comparison with the regular PC bus (IEEE P996) include: compact form-factor (reduced from from 3.8 to 3.6 inches), the unique self-stacking bus, the pin-and-socket connectors and lower power consumption.

Figure 7 shows the structure of the embedded computer and its PC104 component boards: CPU board PCM-3372F-S0A1E (Advantech), data acquisition board PC104-DAS16Jr/16 (Measurement Computing), Power Supply Unit HESC104 and Battery Pack BAT-NiMh45 (Tri-m Systems).

The main CPU board is a single-board computer (no division into the mother-board and other daughter-boards, instead, the design is centered on a single board), of which the specification characteristics make it close to a small laptop computer. The board has Intel Ultra-Low Voltage fanless VIA Eden V4 1.0 GHz processor, 1GB RAM of DDR2 standard at 533 MHz, and the system chipset VIA CX700 with 64MB VRAM.

The I/O periphery consists of two serial ports, six USB 2.0, keyboard/mouse slots, audio and 8-bit GPIO ports, 10/100 Mbps Ethernet interface, and a slot for flash type I card.

The data acquisition unit is a 16-channel board with ADC 16 channels with 16 bit resolution. Such configuration of the card allows that the data acquisition from the sensor array from 16 channels in parallel, that in turn speeds up the processing by a factor of 16. The maximum acquisition rate of 100KHz is more than enough to read the signals from the sensor array, as the maximum read-out speed on the sensor scanning electronics is not greater than 4KHz. The input range in the unipolar mode is set to [0; 5]V and [0; 10]V, for polymeric and MOX sensor array respectively. The DMA mode support is implemented to reduce the CPU overhead during the data read-out.

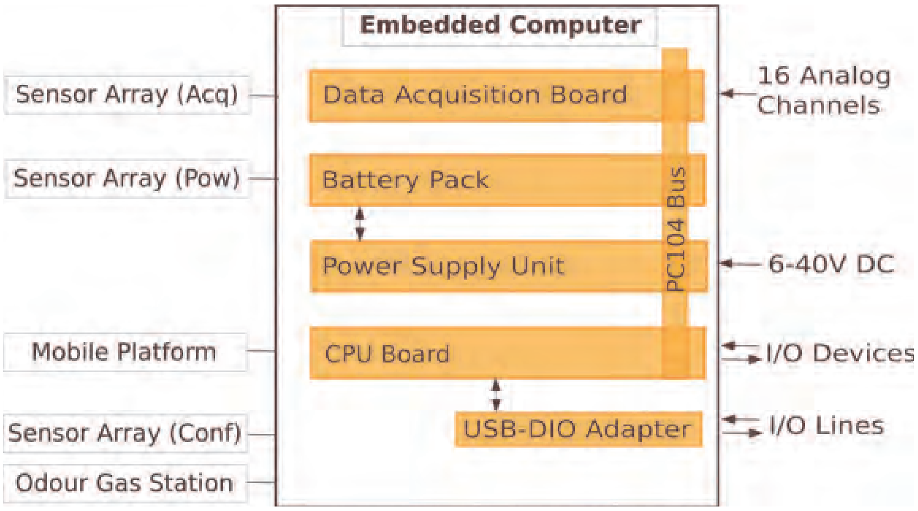


Fig. 7. The PC104-based embedded computer.

The power supply unit is a DC-DC converter with a wide range of input voltage from 6V to 40V DC and the output power if 60W. The UPS mode is supported with board configuration stored in the EEProm memory.
The power consumption of the embedded computer in the complete configuration is typically 9W (maximum of 15.5W). The polymeric sensor array with 64K elements requires from 4W to 10W. Given the maximum power consumption 25.5W, the selected battery pack with capacity 500 mA per hour will guaranty an autonomous operation around 1.1 hour.

2.3.3 Software layer

The models for odor localization and classification have been implemented using IQR (see figure 8), a multilevel neuronal simulation environment that provides a tool for graphically designing large-scale real-time neuronal models (3). It is designed to visualize and analyze data on-line and interfacing to external devices like robots are possible thanks to its modular structure. IQR applications thus acquire data from the robot sensors, process them using the above described models of odor source localization and classification and finally sends motor commands to the robot in real-time.

2.4 Experimental set-up

The experimental scenario is a controlled indoor environment. The robot is tested in two main tasks: (1) odor localization; (2) odor classification. The scenario uses a wind tunnel that creates an odor plume where the robot can freely move. To track the trajectory of the robot and compute its heading direction inside the wind tunnel we use an overhead tracking camera. The chemical compounds used to test the odor classification are ethanol and acetone diluted in distilled water. An ultrasonic source is used to disperse the chemical compounds and generate a rapidly evaporating mist.

2.4.1 The wind tunnel

The conducted experiments took place in a wind tunnel which was located at the SPECS lab in Barcelona, Spain. The wind tunnel is made of a wooden skeleton and is covered with a transparent polyethylene sheet of low density. It consists of two main modules: the first one is the main tunnel - a controlled space where the robot is placed and can freely move. The second part is where the air-flow is generated, using four exhaust ventilators to create

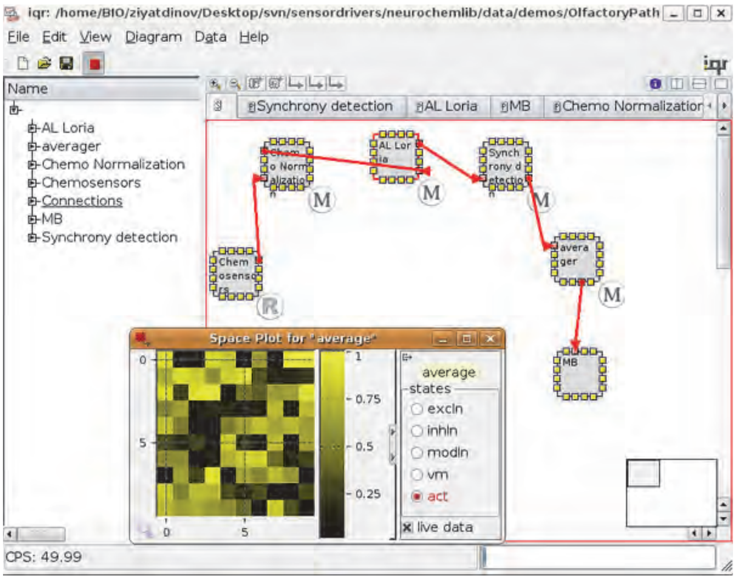


Fig. 8. The iqr simulation environment running the olfactory system of the insect. Space plot shows the neuronal activity of a prototypical neuronal group.

negative pressure. The plume that is created moves across the whole wind tunnel from the point of the odor source to the ventilators where the air is extracted out of the experiment room. Each ventilator is a 4.4KW centrifugal fan and the flow velocity within the wind tunnel is up to 1.0 m s^{-1}. The wind tunnel has to be large enough for the robot, and therefore is 3 meters wide, 4 meters long and 54 centimeters high. For the odor localization experiment, the starting point of the robot was set in front of the fans which is the outlet of the wind tunnel and the odor source was placed in the upwind end of the wind tunnel (see figure 9).
As for the classification experiment we needed to have more stable conditions we placed the robot in the mid spatial position inside the wind tunnel. The odor was spread through the tunnel during five minutes before running the experiment. Additionally, the robot remained in the initial position during the whole the experiment. These two restrictions kept the sensory input as stable as possible. This task was tested with two different odors composed of ethanol (20%) or acetone (20%).

2.4.2 Vision based tracking system (AnTS)

To track the robot's trajectory, a monochrome camera is placed 3 meters above the testing arena. An IR filter is added to the camera to allow the system to track the robot independently of the light conditions. AnTS, a vision based tracking system is used to identify the three points created by the robot's IR LEDs. It computes the robot's orientation and absolute position inside the wind tunnel.

3. Results

Two main experiments were conducted to test the odor classification and the casting behavior of the robot. The latter was performed to assess the odor localization strategy implemented on the robot and the former to assess the robot's ability to classify chemical compounds. Both experiments were conducted in the wind tunnel.

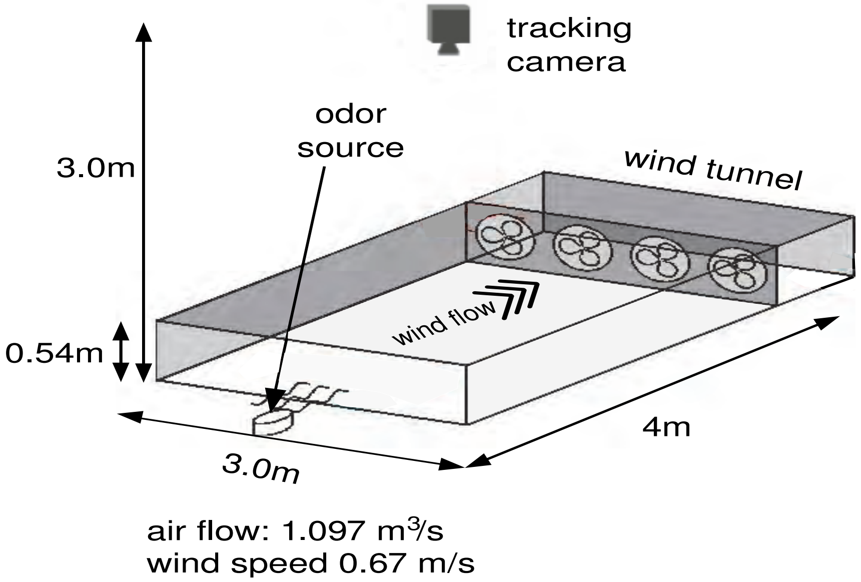


Fig. 9. Layout of the wind tunnel including the position of the camera of tracking system. Red arrows show the wind flow direction from the odor source towards the end where four fans extract the flow out of the tunnel.

3.0.3 Chemical plume in the wind tunnel

First we performed a guided tour of the robot through the wind tunnel to log the sensory data together with the robot position in order to assess the general pattern of chemosensor readings. Figure 10 shows the summed response of the chemosensors for the different robot positions inside the wind tunnel with two chemical sources (Ethanol 1% and Acetone 1%), showing the plume pattern inside the tunnel.

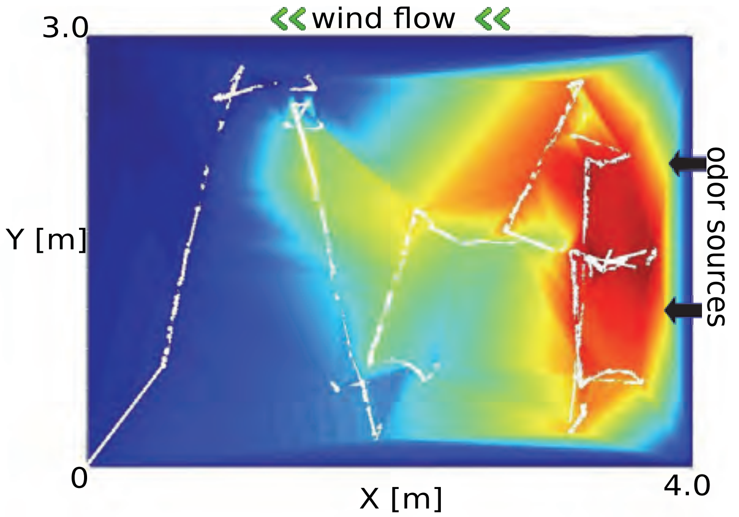


Fig. 10. Chemo sensor readings sampled at different points (white dots) by the robot inside the wind tunnel. The overall plume intensity is captured by the heat plot using the summed input of all 16 chemical sensors.

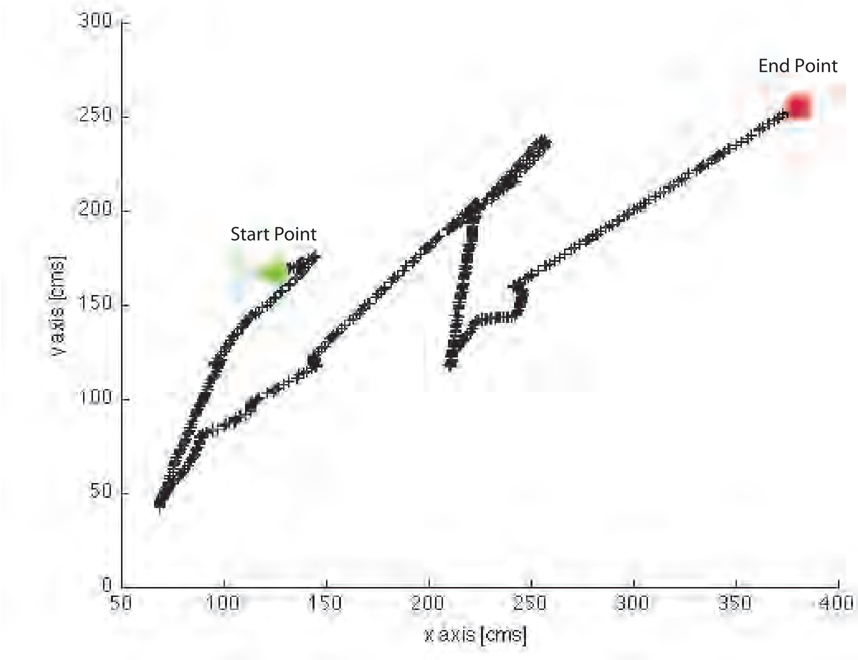


Fig. 11. Robot position plot while casting. The robot is placed in the downwind end of the wind tunnel, in front of the ventilators facing upwind with the chemosensors. The initial point of the robot is marked in the green spot and the end point in red.

3.0.4 Chemosearch

We first discuss the casting behavior of the robot. The robot was placed in the downwind end of the wind tunnel facing upwind. As mentioned above, the moth olfactory model performs a surge when a chemical plume is detected by the sensors and a cast when the plume is lost. In the first experiment we tested the casting model to investigate the explorative behavior with no chemical compounds present. To calculate the robot's trajectory, we performed an offline analysis of the collected robot position data. Figure 11 shows the trajectory of the robot while casting. Our results show a correct crosswind casting movement as no chemicals are detected. However, the casting does not cover the wind tunnel breadth, the main reason being the restricted maneuverability of the current robotic platform. Nevertheless, this preliminary result is promising since the casting model works as expected, reproducing a crosswind cast.

3.0.5 Classification

The results in classification show a successful synchronization of the foreground neurons corresponding to the pattern in both experiments. The Projection Neuron (PN) output is fed to a synchrony detector group implemented in iqr. The plume testing experiments were conducted for a variety of concentration ranges from 1% to 20%.

4. Conclusions and discussion

We have demonstrated the implementation of an autonomous embedded robot that performs moth-like chemosearch and classification strategies. Our models are implemented using the IQR large-scale neuronal simulator and runs on-board the embedded computer. The robot is capable of performing autonomous casts inside the wind tunnel and of classifying two different odors. Nevertheless, we observe that the maneuverability of the robot is restricted:

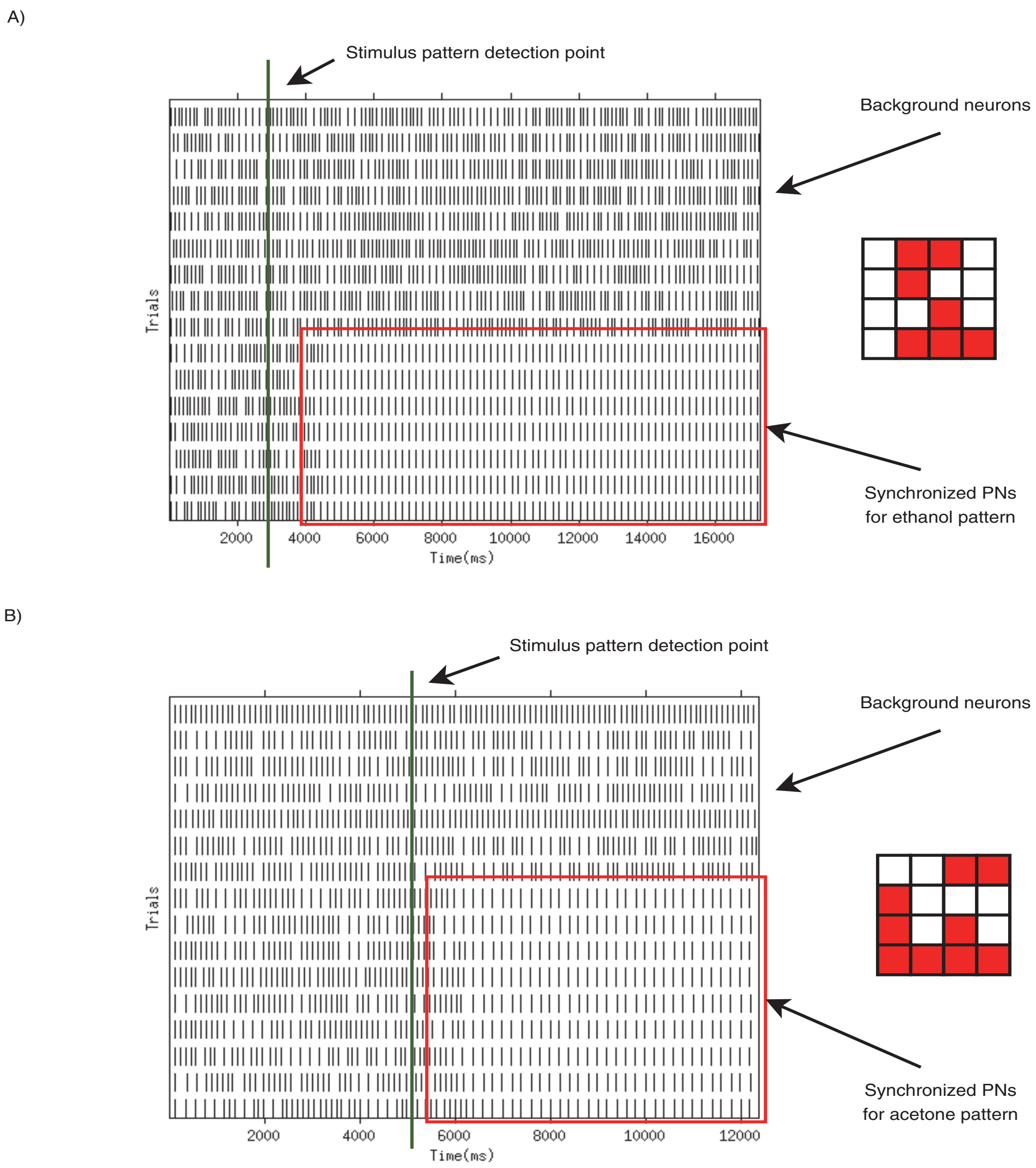


Fig. 12. Raster-plot of the two experiments for the 16 input neurons. The neurons have been grouped into foreground and background neurons of the corresponding patter. The first period corresponds to the time the integrator needs to recognize the pattern (5 and 3 seconds respectively). Once the pattern was input to the network (green line), it would make the corresponding neurons to spike synchronously in about 1 second. The synchrony detector effectively shows the pattern at the output when the specific neurons were synchronized with respect to the background neurons.

the motors are too fast to perform controlled surge and cast. We currently are building a new robotic platform that achieves lesser speed and has a lesser turning radius. The classification results can be considered as a proof of concept for the possibility to classify odors with the antennal Lobe model proposed in (15) and adapted in (12). However, the capability of this model to actually distinguish mixtures of components with real sensor signals and with dynamic input (i.e. a moving robot), or to act in the presence of a distractor in the environment are not yet clear. Further work is intended as extensions of this model with Temporal Population Coding (TPC) strategies, which has been suggested and is consistent with both vertebrate and invertebrate physiology (5; 8; 23).

5. Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216916 for the NEUROCHEM project.

6. References

[1] Beccherelli, R., Zampetti, E., Pantalei, S., Bernabei, M. & Persaud, K. C. (2009). Very large chemical sensor array for mimicking biological olfaction, *OLFACTION AND ELECTRONIC NOSE: Proceedings of the 13th International Symposium on Olfaction and Electronic Nose* 1137(1): 155–158.

[2] BermÃždez, S., Bernardet, U., Guanella, A., Pyk, P. & Verschure, P. F. (2007). A biologically based chemo-sensing uav for humanitarian demining, *International Journal of Advanced Robotic Systems* 4(2): 187–198.

[3] Bernardet, U., Blanchard, M. J. & Verschure, P. F. M. J. (2002a). Iqr: a distributed system for real-time real-world neuronal simulation, *Neurocomputing* 44-46: 1043–1048.

[4] Bernardet, U., Blanchard, M. & Verschure, P. F. M. J. (2002b). Iqr: a distributed system for real-time real-world neuronal simulation, *Neurocomputing* 44-46: 1043–1048.

[5] Carlsson, M. A., KnÃijsel, P. & P. F. M. J. Verschure, B. S. H. (2005). Spatio-temporal ca2+ dynamics of moth olfactory projection neurones., *Eur J Neurosci* pp. 647–657.

[6] Grasso, F. W., Consi, T. R., Mountain, D. C. & Atema, J. (2000). Biomimetic robot lobster performs chemo-orientation in turbulence using a pair of spatially separated sensors: Progress and challenges, *Robotics and Autonomous Systems* 30(1-2): 115–131.

[7] Hansson, B. S. (2002). A bugâĂŹs smellâĂŞresearch into insect olfaction., *Trends Neurosci* pp. 270–224.

[8] Knüsel, P., Carlsson, M. A., Hansson, B. S., Pearce, T. C. & Verschure, P. F. M. J. (2007). Time and space are complementary encoding dimensions in the moth antennal lobe, *Network (Bristol, England)* 18(1): 35–62.

[9] Kowaldo, G. & Russell, R. A. (2008a). Robot odor localization: A taxonomy and survey, *The International Journal of Robotics Research* 27(8): 869–894.

[10] Kowaldo, G. & Russell, R. A. (2008b). Robot odor localization: a taxonomy survey, *The International Journal of Robotics Research* 27(8): 869–894.

[11] Laurent, G. (1999). A systems perspective on early olfactory coding., *Science* pp. 723–728.

[12] Lechón, M., Martinez, D., Verschure, P. F. M. J. & i Badia, S. B. (2010). The role of neural synchrony and rate in high-dimensional input systems. the antennal lobe: a case study., *International Joint Conference on Neural Networks*.

[13] Li, C., Chen, L. & Aihara1, K. (2006). Transient resetting: a novel mechanism for synchrony and its biological examples., *PLoS Computational Biology* .

[14] Liu, Z. & Lu, T. (2008). Odor source localization in complicated indoor environments, *2008 10th International Conference on Control, Automation, Robotics and Vision* pp. 371–377.

[15] Martinez, D. & Montejo, N. (2008). A model of stimulus-specific neural assemblies in the insect antennal lobe, *PLoS Computational Biology* .

[16] McCulloch, W. & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics* 5: 115–113.

[17] Pearce, T., Chong, K., Verschure, P. F., i Badia, S. B., Chanie, M. C. F. & Hansson, B. (2004). Chemotactic search in complex environments: From insects to real-world applications, *Electronic Noses & Sensors for the Detection of Explosives* (159): 181–207.

[18] Persaud, K. & Dodd, G. (1982). Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose., *Nature* 299(5881): 352–355.

[19] Rachkov, M. Y., Marques, L. & de Almeida, A. (2005). Multisensor demining robot, *Autonomous Robots* 18(3): 275–291.

[20] Stein, R. (1967). Some models of neuronal variability, *JBiophys* 7: 37–68.

[21] Trincavelli, M., Reggente, M., S.Coradeschi, Loutfi, A., Ishida, H. & Lilienthal, A. J. (2008). Towards environmental monitoring with mobile robots, *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 2210–2215.

[22] von Frisch, K. (1974). Decoding the language of the bee, *Science (New York, N.Y.)* 185(4152): 663–668.

[23] Wyss, R., KÃűnig, P. & Verschure, P. F. M. J. (2003). Invariant representations of visual patterns in a temporal population code., *Proc Natl Acad Sci U S A* pp. 324–329.

[24] Zars, T., Fischer, M., Schulz, R. & Heisenberg, M. (2000). Localization of a short-term memory in drosophila, *Science* 288(5466): 672–675.

[25] Ziyatdinov, A., Fernandez-Diaz, E., Chaudry, A., Marco, S., Persaud, K. & Perera, A. (2011). A large scale virtual gas sensor array, *International Symposium on Olfaction and Electronic Nose (ISOEN 2011)* .

[26] Z.Mathews and Sergi Bermúdez i Badia and Paul F.M.J. Verschure, *An Insect-Based Method for Learning Landmark Reliability Using Expectation Reinforcement in Dynamic Environments* IEEE International Conference on Robotics and Automation (ICRA2010)

[27] Zenon Mathews and Sergi Bermúdez i Badia and Paul F. M. J. Verschure, *Action-Planning and Execution from Multimodal Cues: An Integrated Model for Artificial Autonomous Systems* Springer-Verlag, Studies in Computational Intelligence 2010

[28] Zenon Mathews and Miguel Lechón and Jose Maria Blanco Calvo and Anant Dhir and Armin Duff and Sergi BermÃždez i Badia and Paul F. M. J. Verschure, *Insect-Like Mapless Navigation Based on Head Direction Cells and Contextual Learning Using Chemo-Visual Sensors* The 2009 IEEE/RSJ International Conference on Intelligent RObots and Systems (IROS2009), 2009

[29] Sergi Bermúdez i Badia and Ulysses Bernardet and Paul F.M.J. Verschure , *Non-Linear Neuronal Responses as an Emergent Property of Afferent Networks: A Case Study of the Locust Lobula Giant Movement Detector*. PLoS Computational Biology. 2010, 6(3)

[30] Sergi Bermúdez i Badia and Paul F. M. J. Verschure, *Learning from the Moth: A Comparative Study of Robot-Based Odor Source Localization Strategies*, 13th International Symposium on Olfaction and Electronic Nose 2009.

[31] Sergi Bermúdez i Badia and Paul F. M. J. Verschure, *Humanitarian Demining Using an Insect Based Chemical Unmanned Aerial Vehicle*, Humanitarian Demining 2008

[32] Sergi Bermúdez i Badia and Pawel Pyk and Paul F. M. J. Verschure, *A fly-locust based neuronal control system applied to an unmanned aerial vehicle: the invertebrate neuronal principles for course stabilization,altitude control and collision avoidance*, The International Journal of Robotics Research. 2007, 26(7), 759-772.

[33] Pawel Pyk, Sergi Bermúdez i Badia, Ulysses Bernardet, Philipp Knüsel, Mikael Carlsson, Jing Gu, Eric Chanie, Bill S. Hansson, Tim C. Pearce, Paul F. M. J. Verschure , *An artificial moth: Chemical source localization using a robot based neuronal model of moth optomotor anemotactic search*, Autonomous Robots 2006, 20(3), 197-213.

[34] Sergi Bermúdez i Badia, Pawel Pyk, Paul F. M. J. Verschure, *A Biologically Inspired Flight Control System for a Blimp-based UAV*, ICRA 2005. 3053-3059.

3.4 results 3. sensor array modeling – ziyatdinov et al., 2013

Ziyatdinov, A., Fernández Diaz, E., Chaudry, A., Marco, S., Persaud, K., & Perera, A. (2013). A software tool for large-scale synthetic experiments based on polymeric sensor arrays. Sensors and Actuators B: Chemical, 177, 596–604. doi:10.1016/j.snb.2012.09.093 [108]

ATTENTION ¡

Pages 56 to 64 of the thesis are availables at the editor's web

http://www.sciencedirect.com/science/article/pii/S0925400512010076

3.5 Results 4. Data Simulation Tool – Ziyatdinov et al., 2014a

Ziyatdinov, A., & Perera-Lluna, A. (2014). Data Simulation in Machine Olfaction with the R Package Chemosensors. PLoS ONE, 9(2), e88839. doi:10.1371/journal.pone.0088839 [110]

Notice of Republication

This article was republished on April 22, 2014, to correct errors in Figures 3, 4, and 5 that were introduced during the production process. Please download this article again to view the correct figures.

PLOS style does not allow for the code in the PDF version of this article to be displayed as the authors intended. Please see the author-provided PDF attached to this notice to view the correctly formatted code.

The publisher apologizes for the figure errors and the code formatting in the PLOS PDF.

The originally published, uncorrected article and the republished article with corrected figures are also provided here for reference.

Source: http://www.plosone.org/article/info:doi/10.1371%2Fjournal.pone.0097796

Data simulation in machine olfaction with the R package *chemosensors*

Andrey Ziyatdinov[1,2,*] , Alexandre Perera-Lluna[1,2]
1 Department of ESAII, Universitat Politènica de Catalunya, Pau Gargallo 5, Barcelona, Spain
2 Centro de Investigación Biomèdica en Red en Bioingeniería, Biomateriales y Nanomedicina (CIBER-BBN), Barcelona, Spain
∗ E-mail: andrey.ziyatdinov@upc.edu

Abstract

In machine olfaction, the design of applications based on gas sensor arrays is highly dependent on the robustness of the signal and data processing algorithms. While the practice of testing the algorithms on public benchmarks is not common in the field, we propose software for performing data simulations in the machine olfaction field by generating parameterized sensor array data. The software is implemented as an R language package `chemosensors` which is open-access, platform-independent and self-contained. We introduce the concept of a virtual sensor array which can be used as a data generation tool. In this work, we describe the data simulation workflow which basically consists of scenario definition, virtual array parameterization and the generation of sensor array data. We also give examples of the processing of the simulated data as proof of concept for the parameterized sensor array data: the benchmarking of classification algorithms, the evaluation of linear- and non-linear regression algorithms, and the biologically inspired processing of sensor array data. All the results presented were obtained under version 0.7.6 of the `chemosensors` package whose home page is `chemosensors.r-forge.r-project.org`.

Introduction

Data sharing plays an important role in the fields of computer science, statistics and machine learning. In statistical genetics, The Human Genome Project made the full human genome publicly available on the NCBI website in 2001 [1]. That has been one of the key factors in enabling impressive developments, not only in fields related to biological science, but also in statistical genetics and bioinformatics. The web site of The University of California at Irvine (UCI) Machine Learning Repository is an example of the way the machine learning community sets data repository standards and provides educational resources and open-access benchmarking material. This web site contains over 200 data sets from different theoretical domains, including results from data generators. Simulated data is an option when data collection is complicated by issues related to technological limitations, large problem size, privacy agreements or the time required to gather the data. In statistical genetics, The Genetic Analysis Workshops approach current analytical problems by making both real and simulated data sets available to investigators worldwide. The use of simulated data is a widely accepted practice for evaluating the performance of computer algorithms and can be found in many computer science publications.

The purpose of machine olfaction is to design systems able to recognize smells. An experimental device typically consists of an array of gas sensors, acquisition electronics and a software unit for pattern recognition. Such a device, also known as an *electronic nose*, was originally proposed by G. Dodd and K. Persaud in 1982 [2]. The authors introduced a principle for discrimination among complex odourant mixtures inspired by the way the olfactory system processes the signals from broadly tuned receptor cells. In the work of the authors, it was shown that discrimination between odour classes can be performed by means of an array composed of sensors with overlapping performance profiles, instead of highly specific sensors. Signals recorded from the sensors form a special fingerprint in response to odours, however data processing of such multivariate responses was always a crucial stumbling block in the design of the electronic nose.

The practical application of instruments based on sensor arrays is very sensitive to the robustness of the data processing methods involved [3]. In last three decades, substantial advances have been made in signal and data processing of sensor array data [3–5], including in biomimetic or bio-inspired approches [6, 7], although no public repository of data sets has yet been established. The need for a repository of benchmarks has been already mentioned [5], but there are still few data sets publicly available. The UCI Machine Learning Repository contains an archive of 13910 measurements from 16 chemical sensors aimed at tackling the problem of drift compensation in sensor array data [8]. As far as we can ascertain, this is the unique example of an open data set in machine olfaction. We believe that data generators for simulation experiments might be a step forward for the development and testing of data processing algorithms, while the setting up of a data repository and the collection of data sets for this repository would be a productive long-term activity for the machine olfaction community.

The need for data sets specifically designed for machine olfaction applications arises from the fact that this field has a list of practical problems, which are not common to other machine learning domains. Signals acquired from gas sensors are prone to drift due to the intrinsic instability of sensor devices and environmental changes over the course of the experiment. Any transfer of the applications from the original experimental conditions to a new set up also results in certain instrument re-calibration problems. Scenarios important for testing the application include: sensor replacement and sensor failure (for evaluation the robustness of the array), adaptation and habituation tasks (for design of event-based pattern recognition algorithms), and a number of biologically inspired scenarios such as background suppression (for running neural models to simulate the biological olfactory pathway). Parameterization of the difficulty of each scenario is another important issue for the benchmarking of algorithms designed to address the above problems. For further information on the topic, the reader is referred to the most recent review of signal and data processing in machine olfaction [3] and to the thesis of B. Raman for the introductory material relating to neuromorphic data processing in machine olfaction [9].

The development of the package *chemosensors* was initiated within the framework of the NEUROChem project [10]. The testing of the neuromorphic computational models designed in the project necessitated large scale sensor array data (a large number of sensors in the array) and support for multi-component gas mixtures. Although neuromorphic simulations were the first application of the generator tool, the simulated data can be used for a general-purpose experiments in machine olfaction. These typically comprise three steps. In the first step, the practitioner considers an experimental scenario. The scenario typically is defined by a list of analytes and their concentrations and the task type, for example, classification or regression. In the second step, transient signals are acquired from the sensors in array. Common practice is to pre-process the signals to compensate for noise and to extract the features relevant for the discrimination task in the specific scenario. In the third step, data analysis relevant to the given scenario is performed. The decisions made in the first step are the most crucial, in the sense that any further improvement is now difficult, if any critical errors were made at the beginning. The *chemosensors* package is mainly focussed on helping the design of a signal processing toolchain by providing the facilities for data simulation. The challenge of this initial step is to find the best possible combination of analytes and sensors which can discriminate between the analytes. Different types of sensors are evaluated by looking at their key response characteristics for the analytes involved in the specific scenario. Typically, the main characteristics of interest are the sensitivity to target analytes, the selectivity to target analytes across the interferents, and the stability of the sensors.

Our *chemosensors* package allows one to parametrically design an array of virtual sensors and to use it as a data generation tool. The simulation of a single sensor is based on a set of physico-chemical models for conducting polymers, which were derived under simplified assumptions and were presented in our earlier work [11], where models emulating different types of noise (including drift) in sensors also were constructed. The software is written in the R language, is organized as a standard package, available on the R-Forge repository and includes installation instructions and code documentation [12, 13]. The package presented is aimed at providing an open framework of data simulation to tackle the specific issues in machine olfaction previously mentioned. We propose defining the difficulty level of scenarios as the similarity between gas classes, this is independent of the sensor data or simulation models for data generation.

The R language environment is a widely used framework for the distribution of data sets and software for data generation. Published packages for data simulation include the **fwsim** package for functional magnetic resonance imaging [14], the packages **IBDsim** and **hapsim** in statistical genetics [15, 16] and the **simFrame** package for building a general-purpose framework for statistical simulations [17].

Our manuscript is organized as follows. We begin with a description of the materials and methods used to create the *chemosensors* package. Then we explain the parameterization of simulations, and show examples for three machine olfaction tasks: the benchmarking of a classification algorithm, the evaluation of linear and non-linear based regression algorithms and the modelling of the chemotopic convergence of receptor neurons in the early olfactory pathway. Finally, we summarize our work in a Conclusions section.

Materials and Methods

Reference data set

The software package includes the simulation models, which were trained with a reference data set as described in [11]. The reference data used in that work (UNIMAN data set) was collected in The University of Manchester (UNIMAN, UK). The long-term measurements of three analytes ammonia, propanoic acid and n-butanol, at different concentration levels, were performed on an array composed of 17 conducting polymer sensors. The measurement protocol implied that sensors were exposed to a rectangular gas pulse of 329 s, and transient signals from the sensors were recorded at 1 Hz sampling frequency. The periodic measurements lasted over 10 months and resulted in 3925 samples stored in the raw data format. Hence, the UNIMAN data set can be represented as a three-dimensional data array of size 3925 × 329 × 17.

The UNIMAN data set is unique, due to the methodology and precision on the gas delivery station jointly with the long-term experiment. The applications on processing of these data are related to scenarios of gas identification complicated by the noise observed in the sensor signals (mainly the long-term drift noise). The detailed information about the UNIMAN data set and list of related applications can be found in [11] and references therein.

Input protocol

Three different analytes can be used for data simulation, which correspond to the three analytes: ammonia, propanoic acid, and n-butanol in the reference data set. For the sake of simplicity, we use the letters A, B and C to refer to these. Table 1 reports the concentration range for each analyte with concentration units expressed in volume fraction *vol.%*.

The input concentration is defined by a step function, and the lengths of both the exposition and the cleaning phase are equal to 60 time units. This corresponds to the protocol given in the reference data set.

The dynamic range of the virtual sensors is limited to the range from 0.01 to 0.1 vol.% for analytes A and B and to 0.1 to 1 vol.% for analyte C. This corresponds to the range of analyte concentrations in the reference data set given in Table 1.

A transient sensor signal, the output vector $x(t)$, is generated in response to a mixture of analytes, with input concentration matrix $C_0(t)$. The columns of the matrix $C_0(t)$ encode the concentration of three analytes A, B and C. We use i to index the columns of $C_0(t)$, where i takes values 1, 2 and 3. The response of an array of sensors can be expressed as a matrix $X(t)$ comprised of signals from the sensors given in the columns. The number of rows, in both matrices $C_0(t)$ and $X(t)$, is equal to the number of samples per unit time.

Function $C_0(t)$ is defined to be a step function of length 60 time units and the amplitude of the step is denoted by C_0. A time stamp, when the exposition phase ends and the cleaning phase starts, is known as quasi stabilization time and the value of the signal at this point, here x_{ss}, is known as the steady-state value.

Simulation Models

In the *chemosensors* package we used the models designed for polymer based gas sensors and validated these models on the seventeen sensors and three analytes at different concentrations from the UNIMAN data set [11]. This group of models took a matrix of concentrations $C_0(t)$ as input and produced a matrix of sensor array data $X(t)$ as output. Two models, sorption and calibration, emulated the time response of the sensors in the array under noise-free conditions. Three models, concentration noise, sensor noise and drift noise, injected noise to the generated data at different steps of the simulation flow. The response of a single sensor to a mixture of analytes is controlled by the Langmuir isotherm being part of the sorption model. The Langmuir isotherm implies a competitive sorption behaviour and results in a non-linear response to a mixture of analytes. The maximum number of analytes in the mixture is three, as the UNIMAN data set was measured only for three analytes.

The parametrization of the simulation models is summarized in Appendix S1, while the complete description of the models is available in our previous work [11]. Appendix S2 also presents a quantitative comparison between simulated and real data to give the reader the confidence in the data generated by the *chemosensors* package.

Virtual sensor array

The simulation models described in Appendix S1 are implemented in the *chemosensors* package as S4 classes in R [12]. The main class of the package `SensorArray` represents a virtual sensor array and inherits classes from the simulation models, which are `SorptionModel`, `SensorModel`, `ConcNoiseModel`, `SensorNoiseModel` and `DriftNoiseModel`f. Table 2 shows the relationship between the simulation models and the classes in the first two columns. The parameters derived from the reference UNIMAN data are stored in the data sets reported in the third column of Table 2. In addition, the data set `UNIMANshort` contains the short-term reference UNIMAN sub-set of the first 200 samples. All the data sets are distributed with the *chemosensors* package and can be loaded into the R environment by the `data` function.

In this Section, we describe the basic slots of the `SensorArray` class and report their relationship to the parameters of the simulation models. Table 3 summarizes the information about the basic slots of `SensorArray`class.

Virtual sensors can be thought as replicas of the 17 UNIMAN sensors. The data sets of the package store parameters related to the simulation models computed for the UNIMAN sensors (See Table 2). When a virtual sensor is initialized, it adopts one of the pre-computed 17 profiles. By means of such model assembly, one can create a virtual sensor array by controlling only two slots of `SensorArray` class in the basic configuration.

- The `num` slot represents the types of sensors in the array. It is an integer vector whose length is equal to the number of sensors in the array. The elements of the vector `num` can take values from 1 to 17, corresponding to one of the seventeen sets of parameters derived from the UNIMAN sensors. These parameters include K_i, $\beta_{i,k}$, $\tau_{1,i}$, and $\tau_{2,i}$ as presented in Appendix S1.

- The `nsensors` slot stores the number of the sensors in the array.

For instance, a virtual array created with parameters `num 1:2` and `nsensors` 2 has two sensors that represent the first two sensors in the UNIMAN data set. That two UNIMAN sensors were different by the polymer material the film of the sensors was composed from, and the sensors had different chemical selectivity and sensitivity characteristics in response to the three examined analytes: ammonia, propanoic acid, and n-butanol. The two virtual sensors possess the same relationships from the UNIMAN sensors, which are expressed in the parameters of the simulation models, please see [11] for further details.

If one needs an advanced configuration of the array, other slots of `SensorArray` class are available. Many slots are implemented as easy-to-use scaling factors.

- The `alpha` slot is a scaling factor for controlling the non-linearity of a sensor. If `alpha` is equal to 1, then the scaling is omitted and the virtual sensors take the sorption affinities K_i from the `UNIMANsorption` data set according to their types (slot `num`). If `alpha` is not equal to 1, then the magnitudes of the affinity coefficients K_i are scaled up (`alpha` > 1) or scaled down (`alpha` < 1) proportionally, so that the relative relationship along the seventeen sorption profiles is preserved. Non-linearity in a sensor increases with an increase in `alpha`, this is a consequence of the fact that sensors under the Langmuir relation in the sorption model tend to a non-linear behaviour when the coefficients K_i are large. The value of zero is not allowed, because then the sorption model given in Equation (1) in Appendix S1 would be meaningless.
  - Another role of the scaling operation by `alpha` is the regulation of a response to a mixture of analytes. As the output of the sorption model is a weighted (or penalized) sum of the inputs, more penalization is induced with greater magnitudes of K_i and, thus, a greater value of `alpha`. The default value of the slot (2.25) has been selected to favour a more balanced penalization of sensors' responses to different mixtures of the three analytes.
- The `beta` slot is a scaling factor for controlling the diversity across sensors in the array. If `beta` is equal to 0, then the scaling is omitted and the sensitivity coefficients $\beta_{i,k}$ in the calibration model of virtual sensors are taken from the coefficients estimated for the UNIMAN sensors. If `beta` is greater than 0, than the coefficients $\beta_{i,k}$ are derived from the uniform distributions with parameters stored in `UNIMANdistr` data set. The value of `beta` defines the spread of the distributions. The diversity across sensors increases with an increase in `beta`. The default value of `beta` (2) corresponds to a moderate level of diversity.

Note that one can create a copy of the UNIMAN array of the seventeen sensors under the simulation models by setting up `alpha` to 1 and `beta` to 0. Thus, the virtual array will replicate the same properties of non-linearity and diversity as the UNIMAN array.

The magnitude of noise generated by the simulation models is mainly controlled by three scaling slots `csd`, `ssd` and `dsd`, which correspond to concentration, sensor and drift noise models respectively. Values of `csd`, `ssd` and `dsd` typically range from 0 to 1. A value 0 implies a noise-free mode, and the value of 1 has been selected to correspond to the level of noise observed in the reference UNIMAN data set. The default values of the three slots are equal to `0.1`, which supposes a moderate level of noise.

- The `csd` slot is a scaling factor for controlling the concentration noise. It scales the covariance matrix Σ_c in the concentration noise model. The default value is 0.1.
- The `ssd` slot is a scaling factor for controlling the sensor noise. It scales all the covariance matrices $\sigma_{i,k}$ in the sensor noise model. The default value is 0.1.
- The `dcsd` slot is a scaling factor for controlling the drift noise. It scales the covariance matrix Σ_d in the drift noise model. The default value is 0.1.

- The `ndcomp` slot encodes the number of drift components. Its value is equal to the number of columns in the matrix P of the drift noise model. The default value is 1. This corresponds to the one drift component which has been observed in the reference UNIMAN sensor array data [18]. The slot can possess the values 1, 2 or 3.

- The `ndvar` slot defines the structure of the drift noise and encodes the importance of drift components. The slot is a vector which contains the diagonal elements of the covariance matrix Σ_d of the drift noise model. The values of the elements in `ndvar` vector lie in the range $[0, 1]$. The default value is 0.86, given that the value of `ndcomp` slot is 1. The slot can be a vector of up to 3 elements, as limited by the `ndcomp` slot. If three drift components are given, then the default values of `ndvar` are 0.86 0.06 and 0.05.

Workflow

The workflow of data simulations in the *chemosensors* package consists of several steps. In the first step, the practitioner defines analytes and concentration levels for a scenario and the sensors required to build an appropriate array. The basic initialization parameters to build a virtual array include the sensor types `num` and the number of sensors `nsensors` (along with others for more advance configurations). The package contains a special class `Scenario` for the representation of analytes and concentrations. The plot methods of the `SensorArray` class have been designed to perform the exploratory data analysis on the sensor array data.

In the second step, the practitioner generates sensor array data by a single command. In particular, the `predict` method of the `SensorArray` class takes as input a matrix of analyte concentrations and returns as output a matrix of sensor array responses. Parallelized computation of sensor signals is supported, this is necessary in the case of long-term scenario or a large number of sensor elements.

In the third step, the practitioner performs a data analysis on the sensor array data by means of any convenient software tool. In general, the software for data analysis can be an external program, and both matrices of concentrations and sensor signals can be easily exported in a format like *csv* by standard R facilities, as no specific data format is assumed in the package.

The noise level in the array is a simulation parameter which can be updated on-the-fly in the simulation. We consider such flexibility in controlling noise to be a useful option, when the performance of a specific sensor is evaluated under drift-free conditions or when the level of noise is a parameter in benchmarking data analysis algorithms.

Installation

The source code of the *chemosensors* package is hosted on the R-Forge web page [13, 19]. The package is also available on the official CRAN repository of the R packages and can be installed by typing the following command in R:

```
install.packages("chemosensors")
```

That will install the latest stable version of the package and all its dependencies from the CRAN repository. The distributed package is platform-independent and self-contained.

Results

The *chemosensors* package is organized around the S4 classes of simulation models (See Table 2), and the implementation of the classes shares some common features.

- Class constructors can be called in the standard form for S4 classes using the `new` function. For the sake of simplicity, every class has a function, which serves as a wrapper for the class constructor and has the same name as the class.
- The standard methods `show`, `print` and `plot` have been designed for all classes, this makes the output more verbose.
- One uses `@` to access slots of a S4 object. Special *get* and *set* methods have been implemented to access most slots of the simulation models, and the methods have the same names as the slots.

The following code shows a quick-start example of a simulation, where one defines a custom matrix of concentrations, creates a sensor array and generates the data. This is an example of the regression scenario of one single gas A given at several concentration values.

```
conc <- matrix(0, nrow = 120 * 3, ncol = 3)
conc[61:120, 1] <- 0.01
conc[181:240, 1] <- 0.02
conc[301:360, 1] <- 0.05

sa <- SensorArray(num = 1:4, tunit = 60)

sdata <- predict(sa, conc)
```

The concentration matrix `conc` encodes three pulses of analyte A at different concentrations 0.01, 0.02 and 0.05 %. vol. The array `sa` is composed of four sensors of four different sensor types, and the `tunit` parameter is set to 60 to enable the sensor dynamic model for pulses with step 60. Each gas pulses consists of two parts of equal length 60, the gas exposition phase and the cleaning phase (the gap between two consequent exposition phases). Figure 1 (a) depicts the change in analyte concentrations over time, and Figure 1 (b) depicts the signals from the four sensors in response to the concentrations. One can suppress the drift noise in the array by setting the `dsd` slot to zero and repeat the simulation, as shown in the code below. Figure 1 (c) depicts the sensor signals under drift-free conditions.

```
dsd(sa) <- 0
sdata <- predict(sa, conc)
```

In this section, we present some examples of the use of the *chemosensors* package. Firstly, we introduce some basic topics related to the use of the `Scenario` class, the configuration of a sensor array and the generation of sensor array data. Secondly, we give examples of data analysis performed on the simulated data produced by the package. In particular, we show examples of benchmarking a classification algorithm, the evaluation of two regression algorithms and some biologically-inspired modelling.

To perform the classification and regression analyses we use the *caret* package developed by Max Kuhn [20]. This package provides a unified workflow for the process of constructing a predictive model with the support of automated tools for data pre-processing, resampling procedures, feature selection and model tuning. We also use Self-Organizing Maps (SOM) as implemented in the *kohonen* package for some biologically-inspired modelling [21].

Defining scenarios

The `Scenario` class has been introduced to serve as a more compact representation of a concentration matrix. The labels of analytes and the length of pulses are the main parameters required to specify a scenario. For instance, the `conc` matrix in the previous example can be alternatively constructed by creating an object of the `Scenario` class and applying the `getConc` method to extract a concentration matrix, as shown in the code below.

```
sc <- Scenario(c("A 0.01", "A 0.02", "A 0.05"), tunit = 60)
conc <- getConc(sc)
```

The Scenario class also encodes a training set and a validation set (or test set) at the time of initialization. The parameters T and nT respectively encode gas labels and the number of samples per label for the training set, and the parameters V and nV also obtain for the validation set. The training set is followed by a validation set, as is typically accepted in machine olfaction experiments. Randomization of the samples is controlled by the logical parameter randomize. One can re-create the previously created sc scenario by specifying more parameters, as shown in the following code.

```
sc <- Scenario(name = "Regression", tunit = 60, concUnits = "perc",
  T = c("A 0.01", "A 0.02", "A 0.05"), nT = 30,
  V = c("A 0.01", "A 0.02", "A 0.05"), nV = 30,
  randomize = TRUE)
sc

>  Scenario ‘Regression‘ of 180 samples, tunit 60, randomize TRUE
>  - gases A, B, C
>  - Training Set: A 0.01 (30), A 0.02 (30), A 0.05 (30)
>  - Validation Set: A 0.01 (30), A 0.02 (30), A 0.05 (30)
```

The show method prints the basic information about sc object. The plot method provides the same information by depicting the unique gas labels in the training and validation sets. Figure 2 shows the graphics produced by the plot method for the scenario object sc showed above.

```
plot(sc)
```

The resulting scenario sc represents a regression problem for analyte A given at three concentrations 0.01, 0.02 and 0.05. In both training and validation sets there are 30 samples per concentration. It may sometimes be necessary to update a scenario once it is initialized. In the code given below, the add method is used to supplement the training set with two more gas labels; this might improve the accuracy of the model because of a more representative set of concentrations.

```
add(sc) <- list("A", 0.03, 30, "T")
add(sc) <- list("A", 0.04, 30, "T")
```

In practice, it might be necessary to retrieve extra data from the scenario in addition to the matrix of concentrations. The sdata.frame method returns a data frame with additional columns which represent gas labels, time units and set index (training or validation set). In the code given below, the sdata.frame method is applied to the regression scenario created above, and samples indexed from 58 to 62 are printed.

```
cf <- sdata.frame(sc)
cf[58:62, ]

>    index    A B C glab   lab tpoint time set
> 58     1 0.00 0 0  Air   Air    air   58   T
> 59     1 0.00 0 0  Air   Air    air   59   T
> 60     1 0.00 0 0  Air   Air airout   60   T
> 61     1 0.01 0 0    A A0.01  gasin   61   T
> 62     1 0.01 0 0    A A0.01    gas   62   T
```

The resulting cf data frame contains both air and gas A labels in the 6th column lab, because every label entry, for example A0.01, in either training or validation set encodes a gas pulse consisting of two parts, the exposition phase of the length tunit and the cleaning phase of the same length tunit. Note that the cf data frame has a special column tpoint for encoding events on changes between the exposition and cleaning phases of the gas pulse. This variable takes values air, airin, airout, gas, gasin and gasout, and is used for transient feature extraction from transient sensor signals.

- transient feature: All samples are used.
- steady-state (alias ss) feature: Samples with tpoint labels equal to gasout are extracted, this corresponds to the time stamp when the exposition phase is finished and the cleaning phase is to be started.
- step feature: The same samples as for steady-state feature are used, but the sensor data with tpoint labels equal to airout are subtracted. This method of feature extraction also reduces the drift noise.

For example, the concentration matrix depicted on Figure 1 (a) has three time stamps of gasout at 120, 240 and 360 time units, which correspond to the time of extraction of the steady-state signal.

Ten scenarios for machine olfaction proposed in the framework of the NEUROChem project [10] are given File S1. The document contains the description of each scenario in terms of training and validation sets, definition of scenario difficulty and the R code to create an object of Scenario class.

Configuring sensor array

From now on, we will use the default value 1 of the tunit parameter to create any virtual sensor array. Such parametrization means the only steady-state feature in the sensor response, instead of, for example, 120 transient features in the case of the tunit parameter equal to 60. This strategy seems to be reasonable, as that allows us to significantly reduce the number of samples needed to be simulated for testing pattern recognition models, while we will exploit one the most commonly used features from the transient sensor response (steady-state). Hence, the input for the simulation models will be trivial gas pulses each parametrized with tunit 1, that results in one sample of a gas in the exposition phase and one sample of the air in the cleaning phase. The response to the air sample represents a baseline level in the signal, which typically is subtracted from the response to the gas sample, being a standard drift-correction method in the stage of the signal processing (that corresponds to the feature parameter equal to step in the sdata.frame method).

There are several ways to configure a virtual sensor array in the *chemosensors* package. Basic selection of sensor types is controlled by num parameter among other parameters. Information stored in the data sets given in Table 2 characterize the UNIMAN sensors (or sensor prototypes) and can be used for the selection of particular sensor types. The SensorArray class has a group of plot methods plotPolar, plotPCA, plotBox and plotResponse for a visual representation of the relation between analytes and sensors. Here, we show an example of a configuration of a sensor array targeted at discriminating between a set of gas classes: pure analytes A and C at different concentrations and binary mixtures of them.

```
set.AC <- c("A 0.01", "A 0.05", "C 0.1", "C 1", "A 0.01, C 0.1", "A 0.05, C 1")
```

The affinity coefficients K_i in the sorption model are important sensor characteristics for the discrimination task posed. The code given below shows how one creates an array composed of all the 17 sensor types and gets the coefficients K_i by the coefficients method.

```
sa <- SensorArray(num = 1:17)
coef <- coefficients(sa, "SorptionModel")
str(coef)
```

```
>  num [1:3, 1:17] 53.1 43.2 136 65.2 44.1 ...
>  - attr(*, "dimnames")=List of 2
>   ..$ : chr [1:3] "A" "B" "C"
>   ..$ : chr [1:17] "1" "2" "3" "4" ...
```

The relative importance of the sorption coefficients for analytes A and C is estimated by the following code.

```
sort(coef["A", ] / coef["C", ])
```

```
>      2      3      1      5      6     16      9      8      4     15
> 0.3752 0.3809 0.3906 0.4085 0.6864 0.8087 0.8308 1.1584 1.2877 1.3308
>     10     11     12     13      7     14     17
> 1.3837 1.3980 1.7380 2.1962 2.3077 3.6603 6.3235
```

The same comparison can be performed by looking at pre-computed sorption coefficients for the seventeen UNIMAN sensors and stored in the data set `UNIMANsorption`.

```
str(UNIMANsorption)
```

```
> List of 1
>  $ qkc: num [1:17, 1:3, 1:4] 10.02 9.51 9.52 6.57 9.19 ...
>   ..- attr(*, "dimnames")=List of 3
>   .. ..$ : chr [1:17] "1" "2" "3" "4" ...
>   .. ..$ : chr [1:3] "A" "B" "C"
>   .. ..$ : chr [1:4] "Q" "K" "KCmin" "KCmax"
```

```
K <- UNIMANsorption$qkc[, , "K"]
sort(K[, "A"] / K[, "C"])
```

```
>      2      3      1      5      6     16      9      8      4     15
> 0.4307 0.4526 0.4581 0.4820 0.7187 0.8278 0.8666 1.1213 1.2308 1.2881
>     10     11     12      7     13     14     17
> 1.2933 1.3123 1.6083 2.0277 2.0775 3.1035 5.0328
```

The order of sensors is slightly different, as sensors in a virtual array are not exact copies of the UNIMAN sensors, but replicas derived from the UNIMAN parameters.

Now we create three different arrays composed of sensors which are different in affinities to analytes A and C. All the arrays are configured to have 12 sensor elements and zero level of the drift noise.

```
sa1 <- SensorArray(num = 1:3, nsensors = 12, dsd = 0)
sa2 <- SensorArray(num = c(13, 14, 17), nsensors = 12, dsd = 0)
sa3 <- SensorArray(num = c(1:3, 13, 14, 17), nsensors = 12, dsd = 0)
```

Arrays `sa1` and `sa2` include sensors having greater affinity to analyte C and A, respectively. The last array `sa3` is composed of sensor types present in both previous arrays.

Principal component analysis (PCA) is one the most widely used shrinkage methods to represent sensor array data in a low-dimensional space [3, 5]. Principal components, as data projections, are mutually uncorrelated and ordered in variance. It is well known that the principal components of a data set provide a sequence of best linear approximations to that data [22]. We use the PCA technique to evaluate sensor arrays `sa1`, `sa2` and `sa3` in response to a set of gas labels `set.AC`. In particular, we plot the PCA scores of data projected onto the first two principal components.

The *chemosensors* package contains a list of plot methods suitable for evaluating sensor arrays on a set of analytes by means of exploratory graphics. The plot methods are applied to objects of the `SensorArray` class, the input is either a concentration matrix or a set of gas labels, sensor array data are generated on the fly, and feature selection from sensor transients is parameterized.

- `plotPolar` method (default): Sensor array data are computed for a given concentration matrix or a set of gas labels and are plotted in polar coordinates, where sensor numbers are angles and sensor signals are radii.
- `plotPCA` method: A principal component analysis (PCA) is computed on sensor array data, and the graphics show a plot of scores on the first two principal components. The percentage of data variance captured by components also is presented.
- `plotBox` method: Sensor array data are grouped according to gas labels and are shown as a box plot.
- `plotResponse` method: Both input concentration matrix and output sensor array data, given for a sensor array object, are plotted over time as lines.

All the plot methods share the same list of parameters.

- `x`: an object of the `SensorArray` class.
- `conc`: a matrix of analyte concentrations.
- `sdata`: a matrix of sensor data in response to a matrix of concentrations `conc`.
- `set`: a set of gas labels, which is a parameter alternative to `conc` (a further concentration matrix is created via `Scenario` class).
- `feature` (default value `transient`): the name of a method for transient feature extraction from sensor array data.
- `air` (default value `FALSE`): a boolean value as to whether air samples are to be included or not.
- `gcol` (default value `FALSE`): a boolean value as to whether colours for gas labels are to be computed with the method `gcol`.

Now we apply the `plotPCA` method to three sensor arrays `sa1`, `sa2` and `sa3` in response to the set of gas labels `set.AC`.

```
plotPCA(sa1, set = rep(set.AC, 10), air = FALSE)
plotPCA(sa2, set = rep(set.AC, 10), air = FALSE)
plotPCA(sa3, set = rep(set.AC, 10), air = FALSE)
```

We induce 10 repetitions for each gas label and exclude samples of the air in the PCA plot. The default transient feature extraction `transient` is appropriate for the analysis, as the drift noise was set to zero level when creating the arrays of sensors.

Figures 3, 4 and 5 show the distribution of PCA scores for the three arrays. In Figure 3 the scores of two groups for binary mixtures `A 0.01, C 0.1` and `A 0.05, C 1` are closer to the scores of groups for pure analyte C; this means that sensors of the `sa1` array tend to have a greater affinity for analyte C. On the contrary, Figure 4 shows that sensors of the `sa2` array have greater affinity for analyte A. The horizontal line `PC2 = 0` can be used to visually pick up such kinds of observations. Figure 5 shows a balanced distribution of classes in terms of affinities for analytes A and C. In addition, this plot shows more diversity in the PCA scores for `sa3` array; this can be noted by looking at the amount of variance captured by the two principal components PC1 and PC2 (labels on x and y axis).

Generating data

Data generation is performed when one has defined a matrix of analyte concentrations and a sensor array. The `predict` method of `SensorArray` class takes as input the `sa` object of `SensorArray` class and the `conc` concentration matrix and produces as output the `sdata` matrix of sensor signals. Typically, data generation is accomplished by running a single command, as shown in the following code.

```
sdata <- predict(sa, conc)
```

To parallelize the computation, one passes the `cores` (alias `nclusters`) parameter to the `predict` method. For example, two cores are specified in the code example given below.

```
sdata <- predict(sa, conc, cores = 2)
```

Another way to configure the computation on several cores is by using the `options` command, as shown in the following code.

```
options(cores = 2)
```

The are several facilities available in the *chemosensors* package to process the data stored in the `conc` and `sdata` matrices. The `Scenario` class automates the process of creation of concentration matrices. In particular, the `getConc` method returns a concentration matrix encoded by an object of `Scenario` class, and the `sdata.method` method allows the retrieval of such additional variables as `set` and `tpoint` for separation into training and validation sets and for parameterization of transient feature extraction, respectively. The same method `sdata.frame` applied to an object of the `SensorArray` class takes as input four basic parameters: an `sa` object of the `SensorArray` class, the `conc` concentration matrix or a `cf` data frame (obtained from an object of `Scenario` class by the `sdata.frame` method), the `sdata` matrix of sensor signals and the `feature` parameter to define a method for feature extraction. The following code shows an example of using the `sdata.frame` method to construct the `df` data frame, which contains both concentration- and sensor-related information.

```
df <- sdata.frame(sa, conc = conc, sdata = sdata, feature = "step")
```

Benchmarking of a classification algorithm

In this Section, we present a procedure for benchmarking a particular classification algorithm to discriminate a set of gas classes. How one defines the difficulty of the scenarios used for testing is important. Since the level of difficulty has to be independent of the sensor data or simulation models for data generation, we propose determining the difficulty of a scenario by the similarity between analytes in mixture. Such a definition is possible, as the simulation models in *chemosensors* package support mixtures of analytes.

We will use only two classes in the scenarios, constructed as mixtures of two analytes A and C. The first three columns in Table 4 present three scenarios at different difficulty levels. We apply the k-nearest neighbors (KNN) algorithm for classification. It is known that predictions of this method are often accurate, but can be unstable [22]. Thus, we will perform a 10-fold cross-validation procedure (10 repetitions) for the selection of the best parameter k on the training stage with a sufficient number of samples.

In the first step, we generate the gas labels and sensor array data with the *chemosensors* package. We will construct an array based on 17 sensors from all sensor types, and the noise level of all three types will be set to 1. The code below shows an example of producing a data frame `df` for a scenario of difficulty 1. The size of both the training and validation (or test) set has been selected so that each gas label is represented by 100 samples. This results in 10 samples per fold in the 10-fold cross-validation at the time of the model training.

```
set <- c("A 0.02", "C 0.5")

sc <- Scenario(T = set, nT = 100, V = set, nV = 100, randomize =  TRUE)
conc <- getConc(sc)
cf <- sdata.frame(sc)

sa <- SensorArray(num = 1:17, csd = 1, ssd = 1, dsd = 1)
sdata <- predict(sa, conc = conc, cores = 2)
df <- sdata.frame(sa, cf = cf, sdata = sdata, feature = "step")
```

In the second step, we train a model based on the KNN algorithm with the **caret** package. For model tuning, we will explore values 3, 5, 7 and 9 of the parameter k. PCA will be applied for preprocessing of sensor array data; this is one of the common options for building predictive models in machine olfaction [3]. Separation of the training and validation (testing) set will be controlled by the variable `lab` in data frame `df`.

```
Xt <- as.matrix(subset(df, set == "T", select = snames(sa)))
Xv <- as.matrix(subset(df, set == "V", select = snames(sa)))

lab <- subset(df, set == "T", "lab", drop = TRUE)
lab <- gsub(",| ", "", lab)
Yt <- as.factor(lab)

lab <- subset(df, set == "V", "lab", drop = TRUE)
lab <- gsub(",| ", "", lab)
Yv <- as.factor(lab)

library(caret)
fit <- train(Xt, Yt, method = "knn", tuneGrid = data.frame(.k = c(3, 5, 7, 9)),
  trControl = trainControl(method = "cv", number = 10, repeats = 10),
  preProcess = c("center", "scale", "pca"))
```

The results of the training are stored in the object `fit`, and new data can be obtained by the `predict` method applied to this object. The final model with the best tuned parameters (stored in the `finalModel` slot of object `fit`) will be used for the prediction.

```
Yp <- predict(fit, newdata = Xv)
```

Table 4 shows the results of a benchmarking of the KNN algorithm. The fourth column reports the parameter `k` of the best tuned KNN model, and the last two columns contain the accuracy measure for the training and validation set respectively. The accuracy was computed as the ratio of gas classes correctly predicted by the model. We clearly observe that the model complexity, as expressed by greater values of `k`, increases with the greater scenario difficulty. It is reasonable that the discrimination of gas classes at higher levels of difficulty should require a more complex predictive model. The three models fitted to the scenarios at different difficulty levels also show differences in performance: the first model is able to classify 100% of the gas classes in both training and test sets, the second model shows quite good performance, and the third model performs poorly, giving the accuracy of 0.74 on the test set.

Evaluation of regression algorithms

In this Section, we show an example of the regression scenario, which aims to quantify the concentration of a single analyte based on the sensor signals. To simulate data for benchmarking with the *chemosensors* package, one needs to define the analyte concentrations for the `Scenario` class and to configure a virtual sensor array for the `SensorArray` class. Further, one selects a method for the prediction model to perform the regression analysis on the simulated data, where the regression model will use the sensor signals as predictors and the concentrations as responses.

We consider two regression problems: one for analyte A at concentrations 0.01, 0.02, 0.05 and 0.1 vol.% and another for analyte C at concentrations 0.1, 0.4, 1 and 2 vol.%. The concentration range has been selected for each analyte in order to cover the dynamic range and to include the greatest concentration value in the saturation region. The following code shows the definition of a set of gas labels for each analyte.

```
conc.A <- c(0.01, 0.02, 0.05, 0.1)
set.A <- paste("A", conc.A)

conc.C <- c(0.1, 0.4, 1, 2)
set.C <- paste("C", conc.C)
```

We select the types of sensors by means of exploratory graphics available in the *chemosensors* package. We will also shorten the list of candidate types to six: 1, 2, 3, 13, 14 and 17, as they seem to be good candidates according to the characteristics of sorption affinity, as presented above. To evaluate these types of sensors in response to analytes A and C in different concentrations, we will create a virtual array composed of six sensors under drift-free conditions and apply the `plotBox` method, as shown in the code given below.

```
sa <- SensorArray(num = c(1:3, 13:14, 17), dsd = 0)

plotBox(sa, set = rep(set.A, 10), feature = "step",
  sensors = 1:6, sensor.names = "long", gcol = TRUE, scales = "free_y")
```

Figure 6 shows the box plots for the six types of sensors in response to four concentrations of analyte A. The same graphics for analyte C and its set of labels `set.C` is presented on Figure 7. All the sensors show a non-linear response to analytes A and C, as was expected due to the selection of the concentration ranges. In particular, the response to the lowest concentration is quite distinct from the others, whereas the responses to the two largest concentrations are quite close. One can also observe that the three sensors of types 13, 14 and 17 are very noisy in response to analyte A, this corresponds to sensor noise, as the drift noise has been suppressed in the `sa` array.

Since there is not an obvious choice of sensor type, we will try three different arrays composed of 24 sensor elements, as shown the following code.

```
sa1 <- SensorArray(num = c(1:3), nsensors = 24)
sa2 <- SensorArray(num = c(13:14, 17), nsensors = 24)
sa3 <- SensorArray(num = c(1:3, 13:14, 17), nsensors = 24)
```

In the first step, we simulate the data and store them in the `df` data frame, as shown in the following example of code given for the `sa1` array and a set of gas labels `set.A`. We encode the `Scenario` object to make 100 repetitions of each gas label in both training and validation (test) set, this will allow us to have enough data to build a prediction model with validation by the 10-fold cross-validation procedure (10 repetitions).

```
sc <- Scenario(T = set.A, nT = 100, V = set.A, nV = 100, randomize = TRUE)
cf <- sdata.frame(sc)
conc <- getConc(sc)
```

```
sdata <- predict(sa1, conc, cores = 2)
df <- sdata.frame(sa1, cf = cf, sdata = sdata)
```

In the second step, we train two regression models for each combination of sensor array and scenario. We will try one linear method based on Partial Least Squares (PLS) and another non-linear method based on Support Vector Regressor (SVR) with Gaussian radial basis function [22]. The following code shows the training of the two models `fit1` and `fit2`, corresponding to the PLS and the SVR methods, respectively. The computation is given for the scenario for analyte A and the previously generated data stored `df` data frame.

```
Xt <- subset(df, set == "T", select = snames(sa))
Xv <- subset(df, set == "V", select = snames(sa))

Yt <- subset(df, set == "T", select = "A", drop = TRUE)
Yv <- subset(df, set == "V", select = "A", drop = TRUE)

library(caret)
fit1 <- train(Xt, Yt, method = "pls",
  tuneLength = 24, preProc = c("center", "scale"),
  trControl = trainControl(method = "cv", number = 10, repeats = 10,
    selectionFunction = "tolerance"))

fit2 <- train(Xt, Yt, method = "svmRadial",
  tuneLength = 10, preProc = c("center", "scale"),
  trControl = trainControl(method = "cv", number = 10, repeats = 10,
    selectionFunction = "tolerance"))
```

To train both models, we pre-processed the sensor signals by performing centring and scaling operations and applied the 10-fold cross-validation procedure repeated 10 times. We also used the `tolerance` rule from the *caret* package to select the most appropriate model in the model tuning. This rule allows us to avoid overfitting of a regression model and suggests picking the simplest model which is within some percentage tolerance of the best model. The root-mean-square error in prediction (RMSEP) was used to evaluate the performance of the models and score them (the default error measure for regression analysis in the `train` function of the `caret` package). The `fit1` model based on the PLS method has a single parameter `ncomp` which stands for the number of latent variables used in the regression. Tuning of the model was set to explore all the possible values for the `ncomp` parameter from 1 to 24. The `fit2` model based on the SVR method has two parameters, the `C` parameter associated with the cost function and the parameter `sigma` of the kernel. By default, the `train` function of the *caret* package allows the estimation of the value of `sigma` from the data passed for training the model. Thus, tuning of the model was configured to explore 10 possible values of `C` parameter from 0.5 to 128, while the value of `sigma` parameter was pre-calculated and fixed in the procedure of model tuning.

For prediction of concentrations for new data, one applies the `predict` method to the model, as shown in the code below for the `fit1` model and sensor signals stored for validation in `Xv` variable.

```
Yp <- predict(fit1, Xv)
```

The first results obtained for the initial experimental set up described above were confusing in terms of comparison among the arrays and the methods, and the error in both training and prediction was rather high and even comparable with the minimum concentration value of the analytes. The reason for experimental failure was explained by the substantial amount of drift-related noise observed in the sensor signals. Poor performance of the predictive models was attributable to the absence of any drift compensation procedure, this is a compulsory step in the most of the data processing methods in machine olfaction [4]. Hence, we repeated the step of data generation for all three sensor arrays `sa1`, `sa2` and `sa3` by setting the level of drift noise to zero. This strategy is reasonable, as the application of signal processing methods for drift compensation is outside the scope of this study, whose objective is the comparison of different arrays and regression methods on the quantification task for analyte concentrations.

Tables 5 and 6 summarize the results obtained from the drift-free experimental set up for analytes A and C, respectively. Three arrays `sa1`, `sa2` and `sa3` are numbered by indexes 1, 2 and 3, as given in the first column. All the arrays are composed of 24 sensors and differ in the types of sensors, which are listed in the second column. The regression method and the best set of parameters for it (as derived after the model tuning) are given in the next two columns. The last two columns report the RMSEP for the training and test sets.

The comparison between PLS and SVR methods in terms of RMSEP values clearly shows that the non-linear models outperform the linear models for each of the arrays. The difference is more noticeable for analyte C than for analyte A. That seems reasonable, as Figures 6 and 7 show that sensor signals in response to analyte C exhibit more a non-linear structure than in response to analyte A (at the given concentrations of the analytes). The best performance (in terms of RMSEP for the test set) for the task of quantification of analyte A is exhibited by the `sa1` array and the SVR model. The `sa2` array, composed of sensors from different types than `sa1`, shows a significantly higher error in prediction; this is assumed to be related to a higher level of the sensor noise in response to analyte A, as was depicted on Figure 6. The performances (in terms of RMSEP for the test set) of the three arrays, for the task of quantification of analyte C, are very similar for the SVR model, and it is difficult to select a preferred configuration of array for this task.

Example of a large-scale simulation

In this Section, we show an application of the *chemosensors* package in performing biologically-inspired data processing of sensor array data. In particular, we will be interested in the modelling of chemotopic convergence of receptor neurons occurring in the early olfactory pathway. We will implement a simple neuromorphic model based on the Self-Organizing Map (SOM) technique and will repeat the experiment conducted in [23] by using data produced from a virtual sensor array.

Since neuromorphic models require a large number of sensors in the array and a sufficient level of diversity across the sensors, we will create an array constructed of 1K elements parametrized with all 17 sensor types and a `beta` parameter of diversity set to 5 (the default value of `beta` is 2).

```
sa <- SensorArray(num = 1:17, nsensors = 1000, beta = 5)
```

Then we compute the matrix of affinity characteristics `aff` for each sensor and for each analyte by the method given in [23]. Further, the `aff` matrix will be used to evaluate the SOM of size 10x10 by means of the `kohonen` package, as given in the code below.

```
aff <- computeAffinity(sa, method = "inverse", norm = "norm")

library(kohonen)

map <- som(scale(aff),
  grid = somgrid(xdim = 10, ydim = 10, topo = "rectangular"), rlen = 500)
```

In the next step, we use three types of gas labels: pure analyte A at concentration of 0.01, 0.02, 0.05 and 0.1 vol.%, pure analyte C at concentration of 0.1, 0.05, 1 and 2 vol.%, and four binary mixtures of analytes A and C. We will suppress all the noise models by means of the `nsd` method and will run the simulation of sensor signals on a machine with 8 cores to get results in a reasonable amount of time.

```
set.A <- paste("A", c(0.01, 0.02, 0.05, 0.1))
set.C <- paste("C", c(0.1, 0.4, 1, 2))
set.AC <- paste(set.A, set.C, sep = ", ")
set <- c(set.A, set.C, set.AC)

sc <- Scenario(set)
conc <- getConc(sc)

nsd(sa) <- 0

sdata <- predict(sa, conc, cores = 8)
df <- sdata.frame(sa, conc = conc, sdata = sdata, feature = "step")
```

The generated sensor array data are stored in the `df` data frame with 12 rows, this corresponds to 12 gas labels stored in the `set` variable. Further, we project signals from 1K sensors onto the 100 cells of the SOM. Figure 8 show the heatmaps of the SOM, where the colours encode the magnitude of the sensor signals in the SOM cells computed by averaging the signals assigned to the given cell. We observe an increasing activity of the map, as expressed in the change from yellow to red, as the concentration of analytes increases in the gas (direction from left to right). Another observation is related to the distribution of sensors or sensor types across the map. The right part of the map is more active in response to analyte A, and the left part of the map shows more activity in response to analyte C. The heatmaps presented in the lowest raw of the figure correspond to the measurements of the binary mixtures, and the SOM maps show activity of both left and right parts of the map.

Conclusions

The *chemosensors* package is a new R package for data simulation targeted at generating gas sensor array data for signal and data processing in machine olfaction applications. The package contains a set of simulation models organized as S4 classes, which are unified in the main class `SensorArray`. This class allows the creation of a virtual sensor array, serves as a data generation tool, and offers a large list of configuration parameters. The class `Scenario` makes it easier to define scenarios and then generate data together with the virtual array. In summary, the *chemosensors* package provides a compact and extensively configurable workflow for data generation, supports parallelization of large-scale computations and offers many graphical facilities to explore sensor array data. In future, the proposed computational framework for the simulation of sensor arrays can be extended to new reference data sets of different types of sensors and/or of different combinations of analytes, that, in turn, will allow addressing new challenges in machine olfaction, for instance, simulation of the sensor response for high-dimensional multicomponent chemical input.

Acknowledgments

The authors would like to acknowledge the effort made by the anonymous reviewers of this manuscript which have greatly contributed to the improvement of the quality of the work. The authors also wish to thank Dr Oleg N. Osychenko and Mrs Marta Oliete for their assistance in revising the English of this manuscript.

References

1. Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, et al. (2001) Initial sequencing and analysis of the human genome. Nature 409: 860–921.

2. Persaud K, Dodd G (1982) Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose. Nature 299: 352–355.

3. Marco S, Gutiérrez-Gálvez A (2012) Signal and Data Processing for Machine Olfaction and Chemical Sensing : A Review. IEEE Sensors Journal 12: 3189–3214.

4. Pearce T, Schiffman S, Nagle H, Gardner J (2003) Handbook of Machine Olfaction - Electronic Nose Technology. John Wiley & Sons.

5. Gutiérrez-Osuna R (2002) Pattern analysis for machine olfaction: a review. IEEE Sensors Journal 2: 189–202.

6. Di Natale C, Martinelli E, Paolesse R, D'Amico A, Filippini D, et al. (2008) An Experimental Biomimetic Platform for Artificial Olfaction. PLoS ONE 3: e3139.

7. Berna AZ, Anderson AR, Trowell SC (2009) Bio-Benchmarking of Electronic Nose Sensors. PLoS ONE 4: e6406.

8. Vergara A, Vembu S, Ayhan T, Ryan Ma, Homer ML, et al. (2012) Chemical gas sensor drift compensation using classifier ensembles. Sensors and Actuators B: Chemical : 1–10.

9. Raman B (2005) Sensor-based machine olfaction with neuromorphic models of the olfactory system. Ph.D. thesis, Texas A&M University.

10. Fonollosa J, Gutierrez-Galvez A, Lansner A, Martinez D, Rospars J, et al. (2011) Biologically Inspired Computation for Chemical Sensing. Procedia Computer Science 7: 226–227.

11. Ziyatdinov A, Fernández Diaz E, Chaudry A, Marco S, Persaud K, et al. (2013) A software tool for large-scale synthetic experiments based on polymeric sensor arrays. Sensors and Actuators B: Chemical 177: 596–604.

12. R Core Team (2013). R: A Language and Environment for Statistical Computing. URL `http://www.r-project.org/`.

13. Zeileis A (1999) Collaborative Software Development Using R-Forge : 9–14.

14. Eriksen MMA, Svante P (2012). fwsim: Fisher-Wright Population Simulation. URL `http://cran.r-project.org/package=fwsim`.

15. Vigeland MD (2012). IBDsim: Simulation of chromosomal regions shared by family members. URL `http://cran.r-project.org/package=IBDsim`.

16. Montana G (2012). hapsim: Haplotype Data Simulation. URL `http://cran.r-project.org/package=hapsim`.

17. Alfons A, Templ M, Filzmoser P (2010) An Object-Oriented Framework for Statistical Simulation: The R Package simFrame 37.

18. Ziyatdinov A, Marco S, Chaudry A, Persaud K, Caminal P, et al. (2010) Drift compensation of gas sensor array data by common principal component analysis. Sensors and Actuators B: Chemical 146: 460–465.

19. Ziyatdinov A (2012). Home page of chemosensors package. URL `http://chemosensors.r-forge.r-project.org/`.

20. Kuhn M (2008) caret Package 28.

21. Wehrens R, Buydens LMC (2007) Self- and Super-organizing Maps in R : The kohonen. Journal of Statistical Software 21.

22. Trevor Hastie, Robert Tibshirani JF (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, second edi edition. URL `http://www-stat.stanford.edu/ tibs/ElemStatLearn/`.

23. Raman B, Gutiérrez-Gálvez A, Perera-Lluna A, Gutiérrez-Osuna R (2004) Sensor-based machine olfaction with a neurodynamics model of the olfactory bulb. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat No04CH37566) : 319–324.

Supporting Information Legends

File S1

Ten scenarios for machine olfaction in the framework of the NEUROChem project [10].

Appendix S1

Parameterization of Simulation Models

Appendix S2

Quantitative Comparison with Chemical Sensor Data

Figure Legends

Tables

Table 1. Dynamic range of concentrations for three gases used in the *chemosensors* package.

Gas Label	Analyte	Concentration range, vol.%
A	Ammonia	0.01 – 0.05
B	Propanoic acid	0.01 – 0.05
C	n-Butanol	0.1 – 1

Dynamic range of concentrations for three gases A, B and C, which correspond to three analytes in the reference UNIMAN data set: ammonia, propanoic acid and n-butanol, respectively.

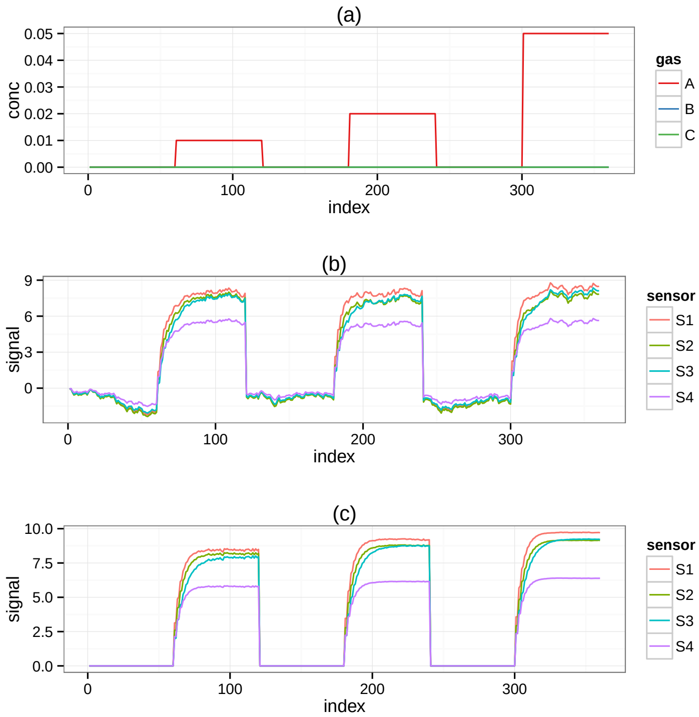


Figure 1. Matrices of analyte concentrations and sensor signals in a simulation with a virtual array of four sensors. On the X axis of each panel, the index values correspond to the row index in the two input concentration and output sensor data matrices of the data generator. Consequently, the values in the columns of these matrices are plotted jointly on the Y axis, while the legend on the right annotates the column names. Panel (a) shows three pulses of analyte A at three different concentrations 0.01, 0.02 and 0.05 vol.%, while the concentration of the other two analytes B and C are at zero level. Panel (b) shows transient signals of four sensors labelled as S1, S2, S3 and S4 in response to the pulses from Panel (a) when all three noises in the sensor array are set up at the 0.1 level. Panel (c) shows sensor signals in response to the pulses under drift-free conditions, while the other two concentration and sensor noises are remained at the 0.1 level. The signals allow for a visual discrimination between the three pulses.

Table 2. Organization of simulation models in the *chemosensors* package.

Simulation Model	Class	Data set
Sorption Model	SorptionModel	UNIMANsorption
Calibration Model (steady-state)	SensorModel	UNIMANdistr
Calibration Model (transient)	SensorDynamics	UNIMANtransient
Concentration Noise Model	ConcNoiseModel	–
Sensor Noise Model	SensorNoiseModel	UNIMANsnoise
Drift Model	DriftNoiseModel	UNIMANdnoise

Simulation models, their classes and associated data sets of parameters computed for the seventeen UNIMAN sensors.

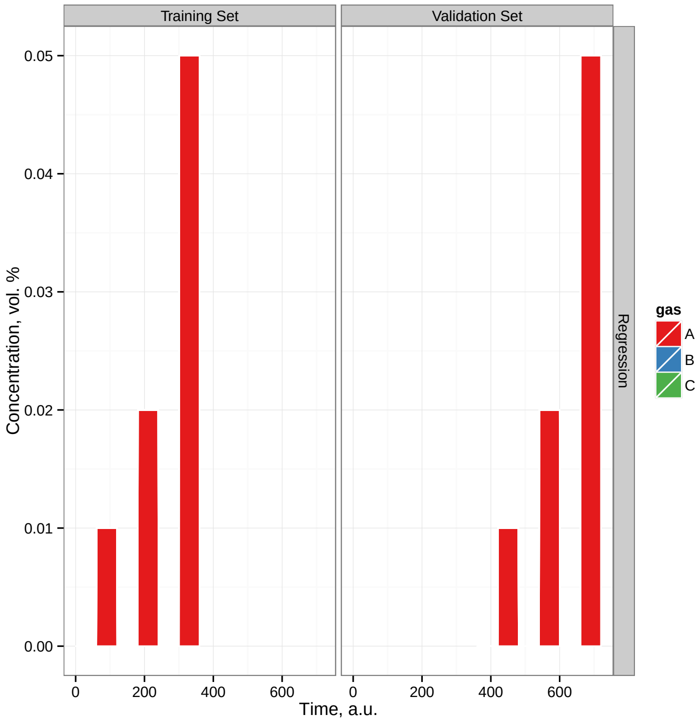


Figure 2. Plot showing the training and validation set, product of the `plot` method applied to a regression scenario. The scenario is defined as a regression on analyte A with both training and validation sets consisting of three pulses of concentrations of 0.01, 0.02 and 0.05 vol.%. The `plot` method applied to a scenario object shows only the unique labels given at training and validation sets. One can apply the `show` method to a scenario object to get more detailed information.

Table 3. Basic slots of `SensorArray` class in *chemosensors* package.

Slot	Default Value	Range of values	Short Description
num	1:2	1, 2, ... 17	type of sensors
nsensors	2	1, 2, ...	number of sensors
ngases	3	1, 2, 3	number of gases
gnames	c('A', 'B', 'C')	any strings	names of gases
concUnits	'perc'	supported string	concentration units
alpha	2.25	> 0	sensor non-linearity
beta	2	≥ 0	sensor diversity
csd	0.1	≥ 0	concentration noise sd
ssd	0.1	≥ 0	sensor noise sd
dsd	0.1	≥ 0	drift noise sd
ndcomp	1	1, 2, 3	number of drift components
ndvar	0.86	[0, 1]	importance of drift components
tunit	1	1, 2, ...	length of a gas pulse

Description of basic slots of `SensorArray` class necessary to parameterize a virtual sensor array.

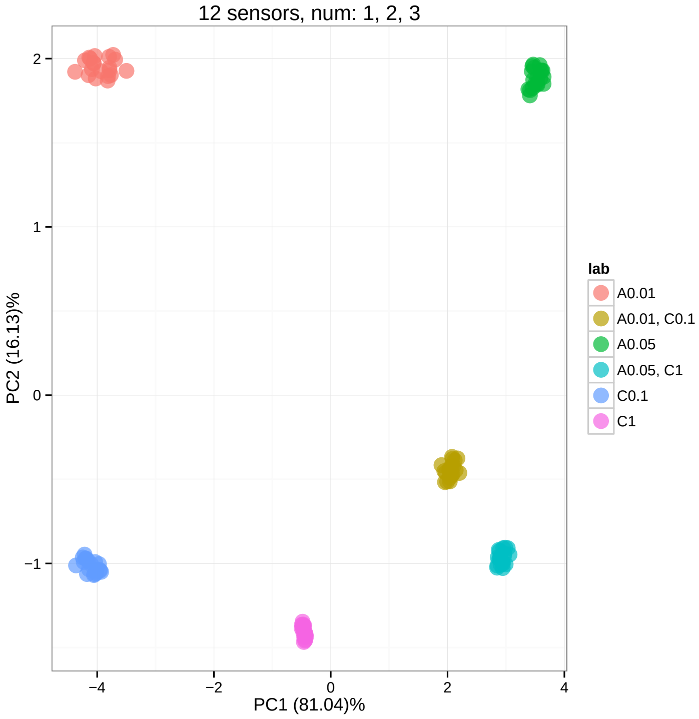


Figure 3. Scoreplot corresponding to the Principal Component Analysis of the sensor array data gathered from the array consisting of 12 sensors of types 1, 2 and 3. The array was exposed to six gas classes: pure analyte A at concentrations 0.01 and 0.05 (labels A 0.01 and A 0.05), pure analyte C at concentrations 0.1 and 1 (C 0.1 and C 1), and two binary mixtures of A and C (A 0.01, C 0.1 and A 0.05, C 1). The concentrations were given at volume fraction units *vol.%*, and the measurement of each gas class was repeated 10 times. The distribution of the scores shows that the sensors in array have more affinity to analyte C that to analyte A. The plot is produced by the `plotPCA` method applied to the sensor array.

Table 4. Classification performance on scenarios given at three different difficulty levels.

Difficulty	Class 1	Class 2	k	Acc. (train)	Acc. (test)
1	A 0.02	C 0.5	3	1.00	1.00
2	A 0.01, C 0.6	A 0.03, C 0.4	5	0.99	0.94
3	A 0.015, C 0.55	A 0.025, C 0.45	7	0.86	0.74

The k-nearest neighbors algorithm was tested on three two-class classification scenarios at three difficulty levels. The scenario difficulty was defined as the similarity between two gas classes. The classification model was trained under 10-fold cross-validation procedure with 10 repetitions, and the best value of the `k` parameter was estimated along possible values 3, 5, 7 and 9 for each classification model. The accuracy in prediction of class labels was used to score the models. The model complexity, expressed in value of parameters `k`, is observed to increase with greater scenario difficulty. The first model provides a perfect performance with a 100% rate of classification, while the last model displays poor accuracy with a classification rate of 0.74 on the test set.

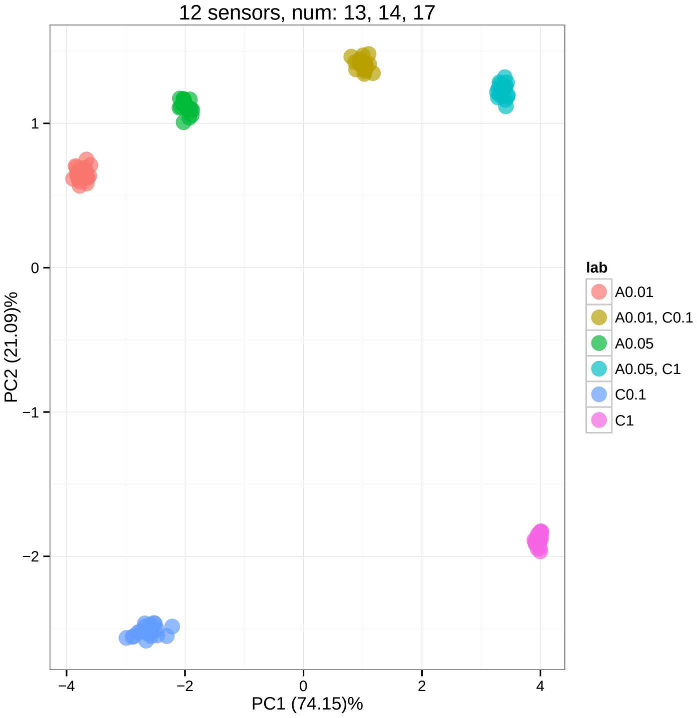


Figure 4. Scoreplot corresponding to the Principal Component Analysis of the sensor array data gathered from the array consisting of 12 sensors of types 13, 14 and 17. The array was exposed to six gas classes: pure analyte A at concentrations 0.01 and 0.05 (labels A 0.01 and A 0.05), pure analyte C at concentrations 0.1 and 1 (C 0.1 and C 1), and two binary mixtures of A and C (A 0.01, C 0.1 and A 0.05, C 1). The concentrations were given at volume fraction units *vol.%*, and the measurement of each gas class was repeated 10 times. The distribution of the scores shows that the sensors in the array have more affinity to analyte A than to analyte C. The plot is produced by the `plotPCA` method applied to the sensor array.

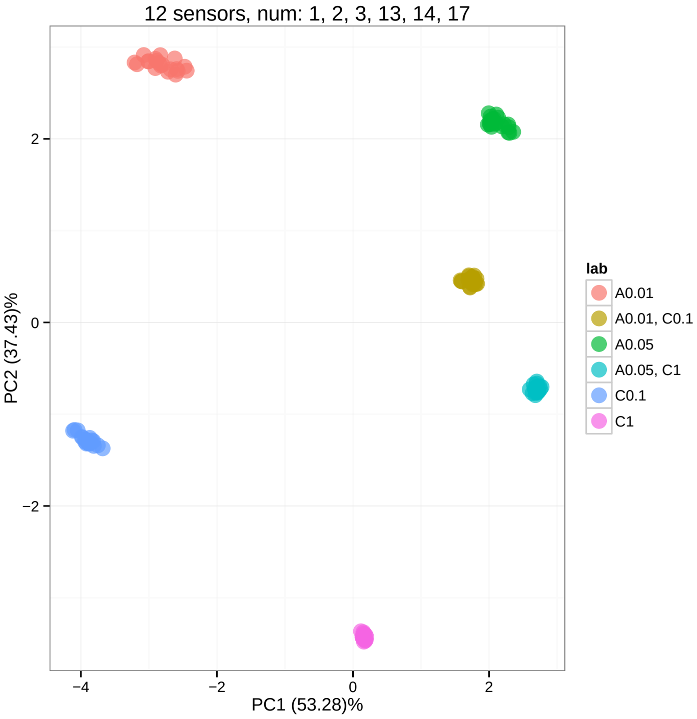


Figure 5. Scoreplot corresponding to the Principal Component Analysis of the sensor array data gathered from the array consisting of 12 sensors of types 1, 2, 3, 13, 14 and 17. The array was exposed to six gas classes: pure analyte A at concentrations 0.01 and 0.05 (labels A 0.01 and A 0.05), pure analyte C at concentrations 0.1 and 1 (C 0.1 and C 1), and two binary mixtures of A and C (A 0.01, C 0.1 and A 0.05, C 1). The concentrations were given at volume fraction units *vol.%*, and the measurement of each gas class was repeated 10 times. The distribution of the scores shows that the sensors in array are balanced in terms of affinity to analytes A and C. The plot is produced by the `plotPCA` method applied to the sensor array.

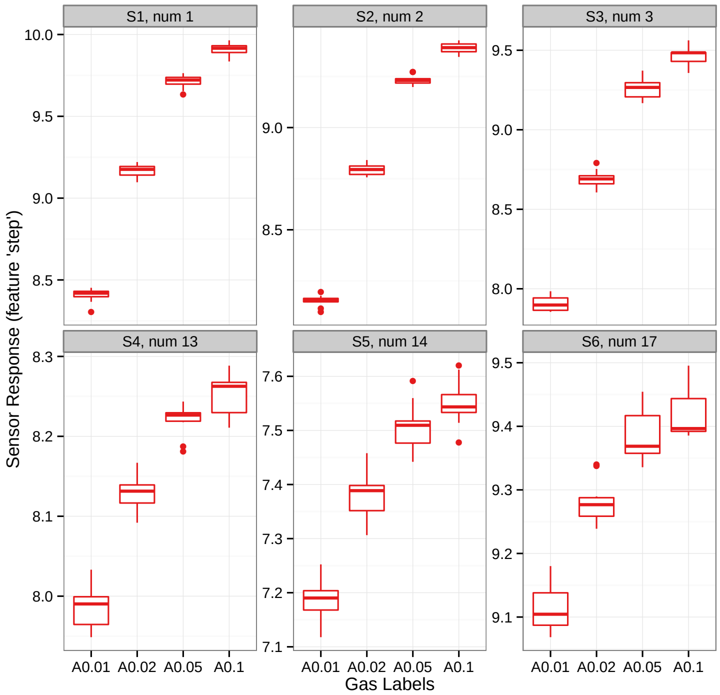


Figure 6. Boxplots for array of six sensors of types 1, 2, 3, 13, 14 and 17 show the distribution of sensor signals in response to analyte A at concentrations 0.01, 0.02, 0.05 and 0.1 vol.%. The concentration values were selected to cover the dynamic range of analyte A and to include the value in the saturation region. All the sensors show a non-linear response to analyte A at the selected concentration range. The three sensors of types 13, 14 and 17 show rather noisy responses. The plot is produced by the `plotBoxplot` method applied to the sensor array under drift-free conditions.

the

Table 5. Performance on prediction of concentration of gas A under drift-free conditions.

Array	Types of sensors	Method	Parameters	RMSEP (train)	RMSEP (test)
1	1, 2, 3	pls	ncomp 9	0.0094	0.0208
1	1, 2, 3	svmRadial	C 2, sigma 10.7	0.0029	0.0039
2	13, 14, 17	pls	ncomp 2	0.0135	0.0133
2	13, 14, 17	svmRadial	C 2, sigma 91.2	0.0028	0.0105
3	1, 2, 3, 13, 14, 17	pls	ncomp 8	0.0086	0.0290
3	1, 2, 3, 13, 14, 17	svmRadial	C 2, sigma 20.1	0.0028	0.0045

Two methods, linear PLS and non-linear SVR, were tested on the regression task of analyte A given at concentration 0.01, 0.02, 0.05 and 0.1 vol.%. Three arrays composed of 24 sensors, different in the types of sensor, were compared in terms of the root-mean-square error in prediction (RMSEP). For each array, the non-linear models outperform the linear models. The first array and the SVR method yield the best performance.

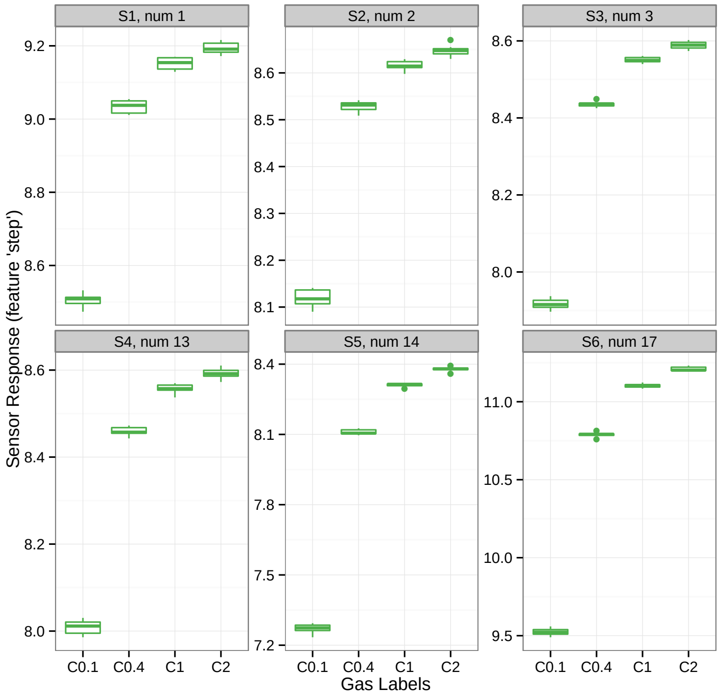


Figure 7. Boxplots for array of six sensors of types 1, 2, 3, 13, 14 and 17 show the distribution of sensor signals in response to analyte C at concentrations 0.1, 0.4, 1 and 2 vol.%. The concentration values were selected to cover the dynamic range of analyte C and to include the value in the saturation region. All the sensors show a non-linear response to analyte C at the selected concentration range. The plot is produced by the `plotBoxplot` method applied to the sensor array under drift-free conditions.

Table 6. Performance on prediction of concentration of gas C under drift-free conditions.

Array	Types of sensors	Method	Parameters	RMSEP (train)	RMSEP (test)
1	1, 2, 3	pls	ncomp 2	0.3373	0.3384
1	1, 2, 3	svmRadial	C 0.5, sigma 237.7	0.0589	0.0837
2	13, 14, 17	pls	ncomp 7	0.2573	1.1317
2	13, 14, 17	svmRadial	C 0.5, sigma 74.9	0.0593	0.0790
3	1, 2, 3, 13, 14, 17	pls	ncomp 10	0.2365	2.8198
3	1, 2, 3, 13, 14, 17	svmRadial	C 0.5, sigma 114.5	0.0593	0.0877

Two methods, linear PLS and non-linear SVR, were tested on the regression task of analyte C given at concentration 0.1, 0.4, 1 and 2 vol.%. Three arrays composed of 24 sensors, different in the types of sensor, were compared in terms of the root-mean-square error in prediction (RMSEP). For each array, the non-linear models outperform the linear models. All three arrays show similar performance with the SVR method, and it is hard to pick the best array.

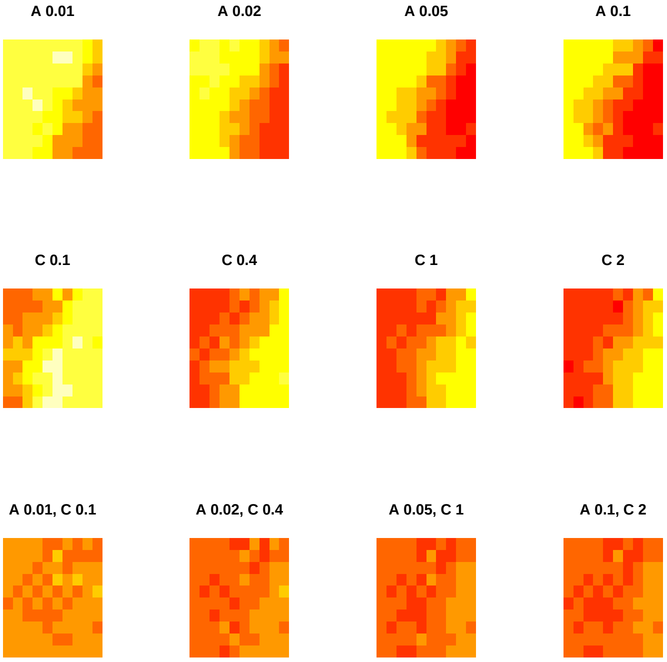


Figure 8. Heatmap of a self-organizing map (SOM) of size 7x7 showing the response to 12 different gases composed of analytes A and C. The map was constructed for the array of 1K sensors based on the affinity coefficients computed per three analytes A, B and C for each sensor, as proposed in [23]. The response of sensor array for each gas was projected onto the map, and the colour on the heatmaps encode the magnitude of the signals in the SOM cells computed by averaging the signals from sensors assigned to the given cell. The activity of the SOM increases as the concentration of analytes increases (direction from left to right). The distribution of the SOM activity in response to different gases show that the right part of the map contain sensors with more affinity to analyte A, while the left part has sensor with more affinity to analyte C.

Appendix S1: Parameterization of Simulation Models

Andrey Ziyatdinov[1,2], Alexandre Perera-Lluna[1,2]
1 Department of ESAII, Universitat Politènica de Catalunya, Pau Gargallo 5, Barcelona, Spain
2 Centro de Investigación Biomèdica en Red en Bioingeniería, Biomateriales y Nanomedicina (CIBER-BBN), Barcelona, Spain

The virtual sensors available in *chemosensors* package are derived from the seventeen UNIMAN sensors based on the model parameters computed for the UNIMAN data set in [1]. In this section, we briefly review the simulation models and their parameters, in order to demonstrate the mechanism of creating virtual sensors.

The sorption model defined in Equation 1 establishes a relation between the environmental concentrations of analytes C_{0i} and the concentrations of analytes C_i when adsorbed by the sensor device. This relationship underlines the Langmuir isotherm for a multi-component gas mixture with two parameters for each analyte i, sorption capacity Q_i and sorption affinity K_i [2].

$$C_i = \frac{Q_i \; K_i \; C_{0i}}{1 + \sum_{j=1}^{3} K_j \; C_{0j}}, \; i \; = 1, \; 2, \; 3 \tag{1}$$

The parameters of the sorption model can be used to control such characteristics of virtual sensors as non-linearity and affinity to analytes in a mixture.

- The *Non-linearity* of a sensor depends on a relation between the numerator and the denominator in the equation. Smaller values of the affinity coefficients K_i make the denominator closer to one, resulting in linear behaviour of the sensor. On the contrary, greater values of K_i lead to saturation mode, where the magnitude of the output concentrations does not depend on the input concentrations.
- The *Affinity* property of a sensor to analyte i in a mixture is controlled by parameter K_i, and is to be estimated by comparison with the affinities for the other analytes.

A static calibration model was defined in Equations (2) and (3) in [1] and simulated the steady-state signal x_{ss} of a sensor in response to the concentrations C_i derived from the sorption model. The calibration model explicitly assumed that the response to a mixture of analytes is the sum of the individual responses to analytes. The main parameters of the model were the sensitivity coefficients $\beta_{i,k}$ to analyte i on the concentration interval k. The calibration model defines such characteristics of a virtual sensor as its sensitivity, selectivity and diversity.

- The *Sensitivity* coefficients β_i give a quantitative estimate of how sensitive a sensor is in response to the analyte i on the given concentration interval k.
- The *Selectivity* of a sensor across two analytes i and j can be evaluated by comparing the sensitivity coefficients β_i and β_j along the analytes.
- The *Diversity* property of a group of sensors is related to the redundancy of the sensor sensitivity coefficient β, and is to be estimated by some multi-variate method.

A dynamic calibration model was defined in Equation (5) in [1] and described the dynamic part of the calibration model. The model derived the temporal signal $x(t)$ from the steady state value x_{ss}. The model had two time constants per analyte as parameters, $\tau_{1,i}$ and $\tau_{2,i}$ for the analyte i. The transient model was rather simple, and we suggest relying on the steady state feature of the signal x_{ss}, rather than on transient features which could be extracted from the signal $x(t)$.

In summary, the sorption and calibration models simulate the seventeen UNIMAN sensors by a set of parameters K_i, $\beta_{i,k}$, $\tau_{1,i}$ and $\tau_{2,i}$ for each sensor. When one defines an array of virtual sensors in the *chemosensors* package, the UNIMAN sensors are replicated by varying the parameters of the simulation models. Parameters $\beta_{i,k}$, $\tau_{1,i}$ and $\tau_{2,i}$ are generated from univariate uniform distributions with control for non-negative values and the level of spread. The parameters K_i are estimated from the seventeen UNIMAN profiles, this allows preservation of the intrinsic number of sensor types given in the reference UNIMAN data set. Hence, one can imagine a virtual sensor as a *replica* of one of the seventeen UNIMAN sensors with similar characteristics on their sensitivity and selectivity profiles, the dynamic ranges for the three analytes and their signal-to-noise performance. The diversity of sensors come from two sources: the relationship between sensors found the reference UNIMAN data set and the distribution of $\beta_{i,k}$ coefficients.

The second group of simulation models defined three types of noise to be injected into the sensor signals. These types were characterised as additive, multiplicative and common noise, corresponding respectively to the concentration, sensor and drift noise models. Data in all three noise models were generated by means of a multi-variate normal distribution of independent variables with diagonal covariance Σ-matrices and zero mean, as shown in Equations (6), (7) and (8) in [1].

The concentration noise model defined the noise term ΔC_0 to be added to the matrix of analyte concentrations C_0. The data in the columns of the matrix ΔC_0 corresponded to the analytes A, B and C, and were derived from the normal distribution with zero mean and diagonal covariance matrix Σ_c. The diagonal form of the covariance matrix underlined the fact that the analytes do not interact with each other.

The sensor noise model generated noise in the sensitivity coefficients $\beta_{i,k}$ from the calibration model. A one-dimensional random walk based on the normal distribution with zero-mean and a single parameter, the standard deviation $\sigma_{i,k}$, was used for analyte i on the concentration interval k.

The drift noise model defined the drift noise ΔX_P to be injected into the matrix of sensor array data X in a multi-variate manner which consisted of several steps. Firstly, a drift-related subspace P was computed by means of Common Principal Component Analysis (CPCA) [3]. Secondly, the noise ΔX_P within this subspace P was generated via a random walk. A multi-dimensional random walk based on a multi-variate normal distribution with zero mean and diagonal covariance matrix Σ_d was used. Thirdly, the generated noise ΔX_P was induced by means of the inverse component correction method [1].

The magnitude and the structure of the noise in the noise models are mainly controlled by the three standard deviation parameters, along with some other parameters.

References

1. Ziyatdinov A, Fernández Diaz E, Chaudry A, Marco S, Persaud K, et al. (2013) A software tool for large-scale synthetic experiments based on polymeric sensor arrays. Sensors and Actuators B: Chemical 177: 596–604.

2. Bai R, Yang RT (2003) Improved Multisite Langmuir Model for Mixture Adsorption Using Multiregion Adsorption Theory. Langmuir : 2776–2781.

3. Ziyatdinov A, Marco S, Chaudry A, Persaud K, Caminal P, et al. (2010) Drift compensation of gas sensor array data by common principal component analysis. Sensors and Actuators B: Chemical 146: 460–465.

Appendix S2: Quantitative Comparison with Chemical Sensor Data

Andrey Ziyatdinov[1,2], Alexandre Perera-Lluna[1,2]

1 Department of ESAII, Universitat Politènica de Catalunya, Pau Gargallo 5, Barcelona, Spain

2 Centro de Investigación Biomèdica en Red en Bioingeniería, Biomateriales y Nanomedicina (CIBER-BBN), Barcelona, Spain

We show the validity of the computational framework for the operation of a virtual sensor array by performing a quantitative comparison between the predictions of the simulation models and reference chemical sensor data. The reference data used in this study is the UNIMAN data set that contains records from the array of 17 conducting polymer sensors in response to 8 gas classes (A 0.01, A 0.02, A 0.05, B 0.01, B 0.02, B 0.05, C 0.1 and C 1.0). The simulated data set is generated from a virtual array of 17 sensors with the noise parameters `csd`, `ssd` and `dsd` set to 0.8 in response to the same gas classes. One design goal of the software tool is the inclusion of sensor specificities in the models (such as sensor drift). Therefore, the comparison between the two data sets has been conducted from two perspectives: first, the validation of the physico-chemical models emulating the responses of the 17 UNIMAN sensors, and, second, the evaluation of the deviations from the sensor responses due to the three types of noises included in the software (concentration noise, sensor noise and drift noise).

The physico-chemical model of a single conducting polymer sensor was implemented in the sorption model under the non-linear relation of the Langmuir isotherm. The so-called short-term UNIMAN data set was used to estimate the two model parameters per analyte i and per sensor, sorption capacity Q_i and sorption affinity K_i, by means of fitting linear regression models (please check [1] for further details). The goodness of the fit of the models was evaluated by means of R^2 statistics. For analyte C, these statistics do not fall below than 0.973, whereas analytes A and B show a slightly worse performance, but always above 0.779.

To show that the three noise models are able to reproduce variance observed in the long term responses of the UNIMAN sensors, we replicated the first 1000 samples of the UNIMAN data set by means of an array of 17 virtual sensors. The qualitative comparison between the two data sets can be performed by means of principal component analysis, where one can observe that the simulated data matches the variance structure of the real data in terms of the class-dependency and noise-related data features. An example of this analysis was previously presented in [1], Section 3.1, Figure 5. For a quantitative analysis, we computed mean and standard deviation statistics for each combination of sensor and gas class. Table 1 reports these statistics for all 17 sensors and for A 0.05 gas class, being the A analyte the one showing the most dispersion. We report the relative difference defined as the absolute difference between UNIMAN and simulated values divided by the UNIMAN value. By collecting the statistics on the relative errors for all combinations of sensor and gas class (136 samples), we show that the relative differences in the means are always below 14.5% (in absolute values) and have 25%, 50% and 75% quantiles equal to -0.0340, -0.0125 and 0.0036, respectively. Similarly, the relative differences in the standard deviations are always below 48.0% (in absolute values) and have 25%, 50% and 75% quantiles equal to -0.2697, 0.0001 and 0.2034, respectively. It is worth to note that the multivariate and multi-component model of drift noise is the dominant component for the long term simulation, as it is for the actual chemical sensor behavior in the UNIMAN data set.

References

1. Ziyatdinov A, Fernández Diaz E, Chaudry A, Marco S, Persaud K, et al. (2013) A software tool for large-scale synthetic experiments based on polymeric sensor arrays. Sensors and Actuators B: Chemical 177: 596–604.

Table 1. Comparison between chemical sensor and simulated data sets (gas class A 0.05).

Sensor	Mean (UNIMAN)	Mean (Simulated)	Diff. in Mean	SD (UNIMAN)	SD (Simulated)	Diff. in SD
1	9.41	10.38	-0.10	0.59	0.53	0.10
2	9.04	8.83	0.02	0.52	0.35	0.33
3	8.92	9.36	-0.05	0.58	0.58	-0.01
4	6.25	6.68	-0.07	0.24	0.21	0.12
5	8.57	9.35	-0.09	0.59	0.48	0.18
6	7.61	8.39	-0.10	0.44	0.35	0.20
7	4.65	4.84	-0.04	0.17	0.21	-0.21
8	5.61	6.14	-0.09	0.27	0.34	-0.25
9	4.34	4.54	-0.05	0.25	0.23	0.09
10	4.73	4.94	-0.04	0.24	0.27	-0.11
11	11.27	12.42	-0.10	0.65	0.42	0.36
12	11.55	12.01	-0.04	0.54	0.43	0.19
13	8.04	8.62	-0.07	0.26	0.30	-0.15
14	7.02	7.77	-0.11	0.30	0.29	0.04
15	8.98	10.22	-0.14	0.36	0.45	-0.23
16	9.89	10.59	-0.07	0.43	0.43	-0.01
17	8.99	9.38	-0.04	0.33	0.52	-0.48

The chemical sensor UNIMAN data set (1000 samples, 17 sensors, 3 analytes and 8 gas classes) is compared to its replica simulated with the *chemosensors* package. The comparison is performed by means of mean and standard deviation statistics computed for each combination of sensor and gas class. This table shows the statistics for A 0.05 gas class as this is the combination that shows the most dispersion and higher discrepancy. The relative differences are computed as the absolute difference between UNIMAN and simulated values divided by the UNIMAN value. The statistics on the relative errors collected for all combinations of sensor and gas class (17*8 = 136 samples) show that (1) the relative differences in the means are below 14.5% (in absolute values) and have 25%, 50% and 75% quantiles equal to -0.0340, -0.0125 and 0.0036, respectively; and (2) the relative differences in standard deviation are below 48.0% (in absolute values) and have 25%, 50% and 75% quantiles equal to -0.2697, 0.0001 and 0.2034, respectively. The negative sign of the relative difference in the means for almost all sensors and A 0.05 gas class indicates the direction of drift effect which tend to increase the value of the sensor responses.

File S1: Ten scenarios for machine olfaction in the framework of the NEUROChem project

Andrey Ziyatdinov, Alexandre Perera-Lluna

21/01/2014

About

This documents introduces a list of ten scenarios for machine olfaction, which were initialy thought in the framework of the NEUROChem project. Each scenario is described in terms of training and validation sets and scenario difficulty.

The R code to create an object of `Scenario` class is given for all ten scenarios, except two scenarios Habituation and Event Detection. The point is that `Scenario` class is not suitable for these two scenarios, because the given class is thought to be constructed only by means of gas pulses, while the two scenarios require another time profile.

First of all, the user needs to load the package.

```r
library(chemosensors)
```

An object of `Scenario` class is initialized with `Scenario` function, that has the following parameters:

- `tunit`: the length of the pulse.
- `concUnits`: the concentration units.
- `randomize`: whether the gas classes need to be randomized in order.
- `T` and `nT`: gas classes for the training set and the number of repetitions for each class.
- `V` and `nV`: gas classes for the training set and the number of repetitions for each class.

In addition to the initialization code, the results of `print` and `plot` methods for objects of `Scenario` class are shown.

Scenarios

Classification

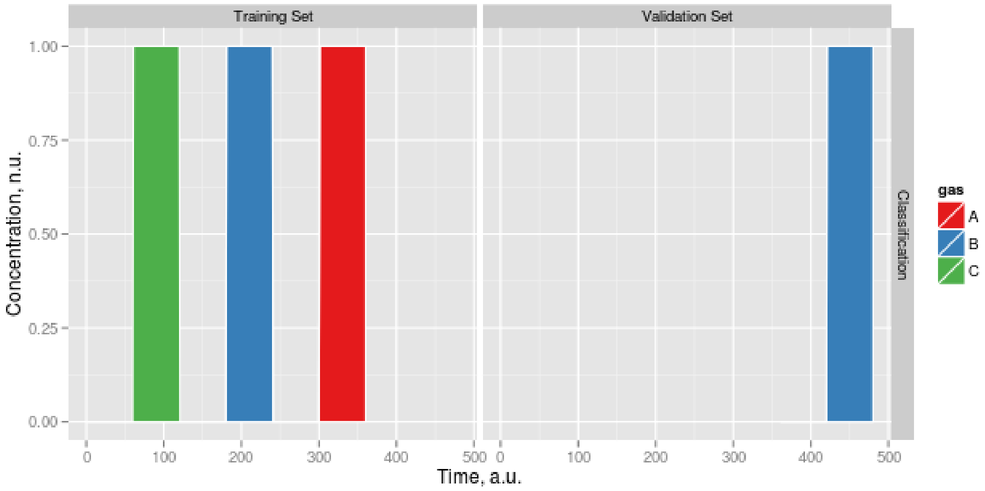
John has three vessels with three odours A, B, C. The system is trained with all three compounds separately. John approaches the vessel B to the system. The machine identifies correctly odour B. The difficulty is the similarity between the odours to be identified.

```r
sc.class <- Scenario(name = "Classification",
  tunit = 60, concUnits = "norm", randomize = TRUE,
  T = c("A", "B", "C"), nT = 30, V = "B", nV = 30)
```

```
sc.class
```

```
##  Scenario `Classification` of 120 samples, tunit 60, randomize TRUE
##  - gases A, B, C
##  - Training Set: A (30), B (30), C (30)
##  - Validation Set: B (30)
```

```
plot(sc.class)
```



Quantification

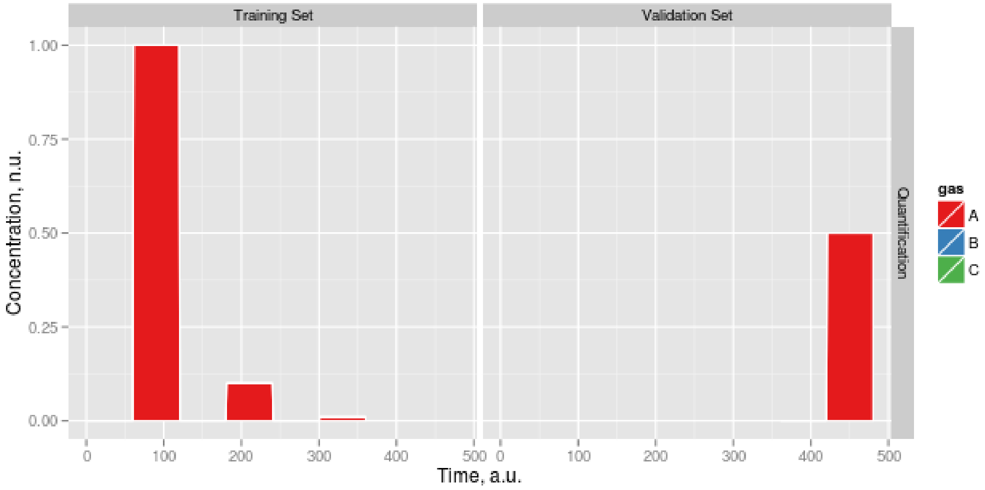
John has five vessels with 100%, 50%, 10% and 1% dilution of A. The system is trained with 100%, 10% and 1% dilution of A. John approaches the 50% A dilution vessel to the system. The machine correctly marks the level of A. The difficulty is the number of different concentration examples available for training.

```
sc.quant <- Scenario(name = "Quantification",
  tunit = 60, concUnits = "norm", randomize = TRUE,
  T = c("A 0.01", "A 0.1", "A"), nT = 30, V = "A 0.5", nV = 30)

sc.quant
```

```
##  Scenario `Quantification` of 120 samples, tunit 60, randomize TRUE
##  - gases A, B, C
##  - Training Set: A (30), A 0.01 (30), A 0.1 (30)
##  - Validation Set: A 0.5 (30)
```

```
plot(sc.quant)
```



Segmentation

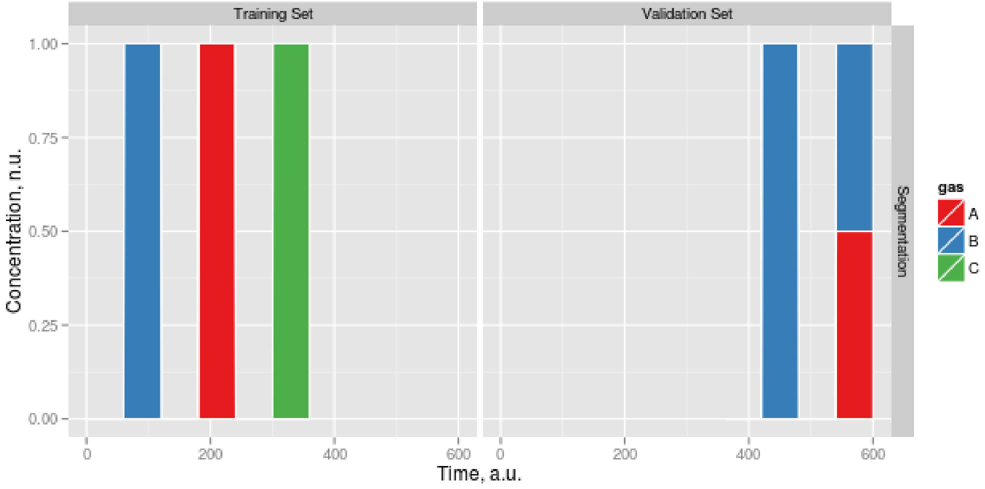
John has three vessels with three odours A, B and C. The system is trained with all three compounds separately. John approaches vessel B to the system. The machine identifies correctly odour B. John approaches A+B to the system. The machine identifies A and identifies B sequentially. The difficulty is the similarity between the odours to be segmented.

```
sc.seg <- Scenario(name = "Segmentation",
  tunit = 60, concUnits = "norm", randomize = TRUE,
  T = c("A", "B", "C"), nT = 30, V = c("B", "A 0.5, B 0.5"), nV = 30)

sc.seg
```

```
##  Scenario `Segmentation` of 150 samples, tunit 60, randomize TRUE
##  - gases A, B, C
##  - Training Set: A (30), B (30), C (30)
##  - Validation Set: A 0.5, B 0.5 (30), B (30)
```

```
plot(sc.seg)
```



Habituation

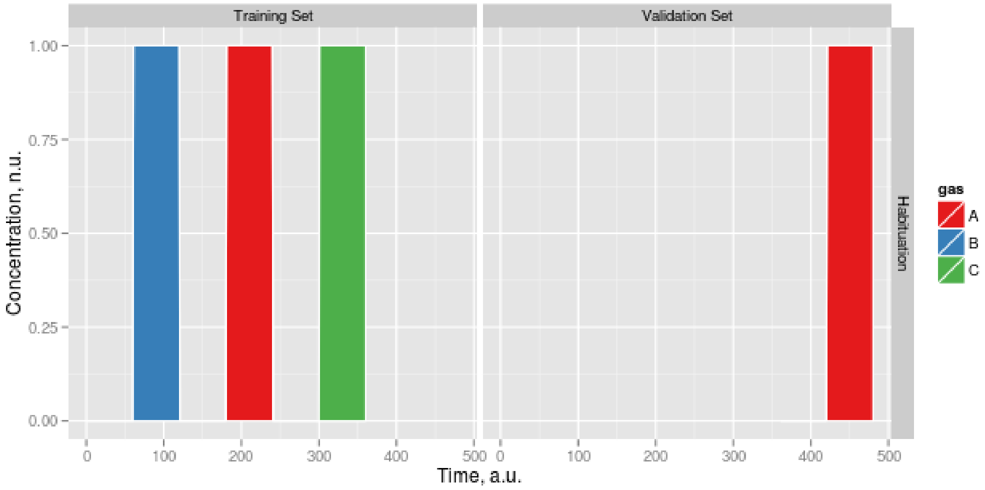
John has three vessels with three odours A, B, C. The system is trained with all three compounds separately. John approaches vessel A to the system. The machine identifies vessel A. After a certain time the machine marks that no odour is present despite the vessel is still exposed to the system. The difficulty is the concentration of odour A, as the higher the concentration the more difficult is to adapt to the odour.

```
sc.hab <- Scenario(name = "Habituation",
  tunit = 60, concUnits = "norm", randomize = TRUE,
  T = c("A", "B", "C"), nT = 30, V = "A", nV = 30)

sc.hab
```

```
##  Scenario `Habituation` of 120 samples, tunit 60, randomize TRUE
##  - gases A, B, C
##  - Training Set: A (30), B (30), C (30)
##  - Validation Set: A (30)
```

```
plot(sc.hab)
```



Event Detection

The system is not trained with any compound. No substance is exposed to the machine. The machine marks that no odour is present. John approaches the vessel B to the system. The machine marks that one odour is present. John approaches the vessel A, in addition to already present B vessel to the system. The machine marks that two odours are present. The difficulty is the magnitude of the second odour B added to the first odour A. The first odour A will be fully delivered on a 100%.

Novelty Detection

John has two vessels with odours A and B respectively. The system is trained only with odour A. No substance is exposed to the machine. The machine marks that no odour is present. John approaches the vessel A to the system. The machine marks that odour A is present. John approaches the vessel B, in addition to already present A vessel to the system. The machine marks that a new odour is present. The difficulty is the concentration of odour B, as the lower the concentration of odour B the more difficult the detection of the novel odour become.

Drift Compensation I

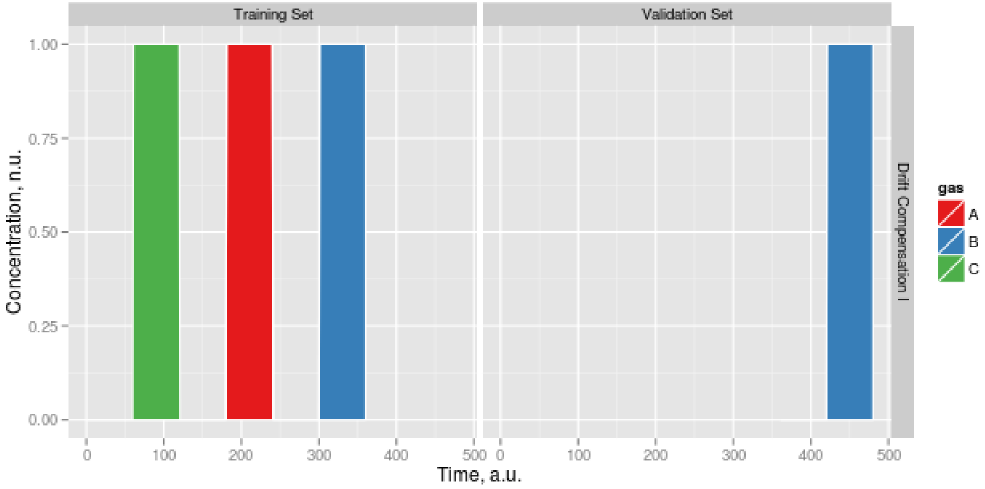
John has three vessels with three odours A, B and C. The system is trained with all three compounds separately. John approaches the vessel B to the system. The machine identifies correctly odour B. A drift process is occurring in the sensor array. John approaches the vessel B to the machine. The machine identifies correctly odour B. The difficulty is the distance in time between the validation set and the training set.

```
sc.drift1 <- Scenario(name = "Drift Compensation I",
  tunit = 60, concUnits = "norm", randomize = TRUE,
  T = c("A", "B", "C"), nT = 30, V = "B", nV = 30)

sc.drift1
```

```
##  Scenario `Drift Compensation I` of 120 samples, tunit 60, randomize TRUE
##  - gases A, B, C
##  - Training Set: A (30), B (30), C (30)
##  - Validation Set: B (30)
```

```
plot(sc.drift1)
```



Drift Compensation II

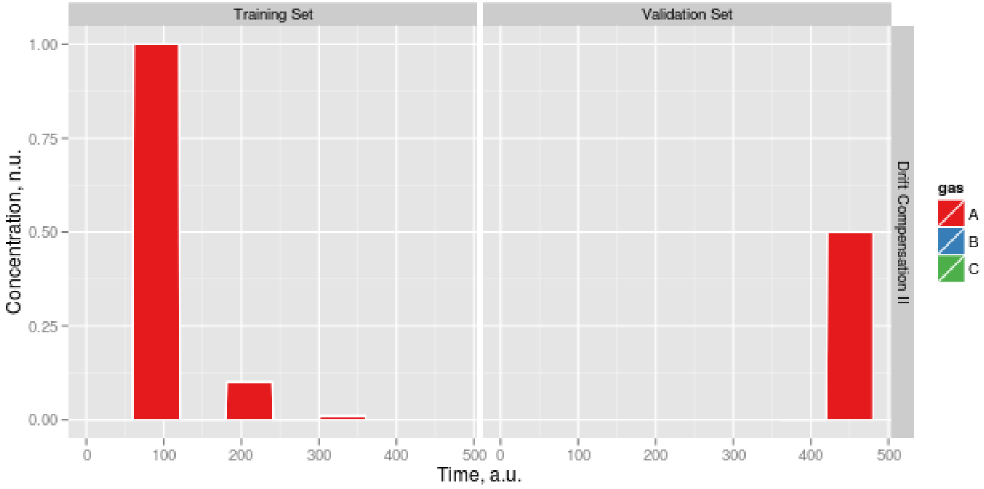
John has five vessels with 100%, 50%, 10% and 1% dilution of A. The system is trained with 100%, 10% and 1% dilution of A. A drift process is induced into the sensor array. John approaches the 50% A dilution vessel to the system. The machine correctly marks the level of A. The difficulty is the distance in time between the validation set and the training set.

```
sc.drift2 <- Scenario(name = "Drift Compensation II",
  tunit = 60, concUnits = "norm", randomize = TRUE,
  T = c("A 0.01", "A 0.1", "A"), nT = 30, V = "A 0.5", nV = 30)

sc.drift2
```

```
##  Scenario `Drift Compensation II` of 120 samples, tunit 60, randomize TRUE
##  - gases A, B, C
##  - Training Set: A (30), A 0.01 (30), A 0.1 (30)
##  - Validation Set: A 0.5 (30)
```

```
plot(sc.drift2)
```



Sensor Replacement I

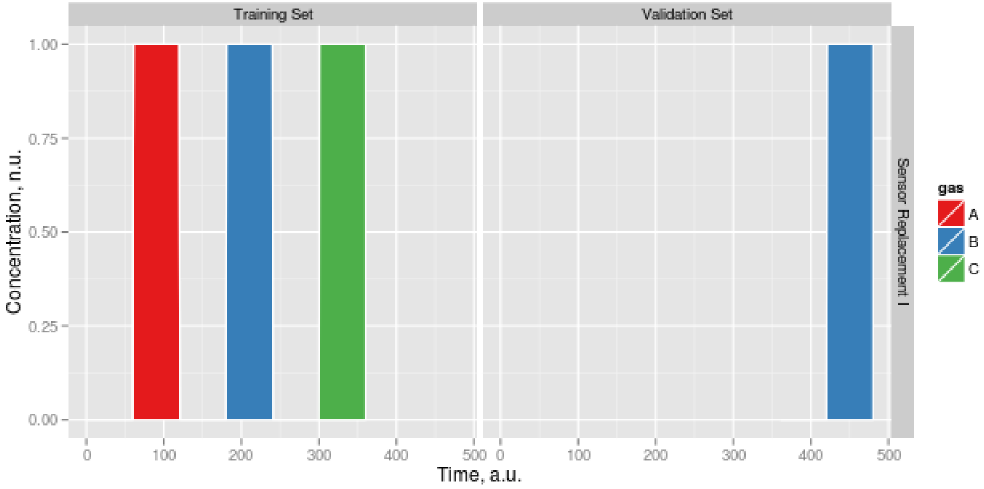
John has three vessels with three odours A, B and C. The system is trained with all three compounds separately. John approaches vessel B to the system. The machine identifies correctly odour B. A certain proportion of specific sensors in the array are (virtually) damaged, so John replaces them with new sensors of the same type. John approaches vessel B to the system. The machine identifies correctly odour B without new training. The dificulty is the proprtion of sensors to be replaced.

```
sc.replace1 <- Scenario(name = "Sensor Replacement I",
  tunit = 60, concUnits = "norm", randomize = TRUE,
  T = c("A", "B", "C"), nT = 30, V = "B", nV = 30)

sc.replace1
```

```
##  Scenario `Sensor Replacement I` of 120 samples, tunit 60, randomize TRUE
##  - gases A, B, C
##  - Training Set: A (30), B (30), C (30)
##  - Validation Set: B (30)
```

```
plot(sc.replace1)
```



Sensor Replacement II

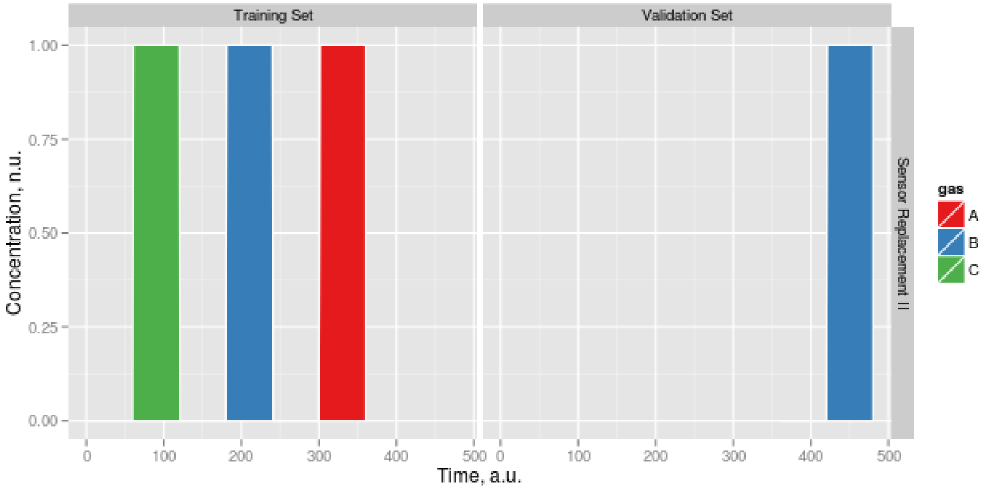
John has three vessels with three odours A, B and C. The system is trained with all three compounds separately. John approaches vessel B to the system. The machine identifies correctly odour B. A certain proportion of sensors in the array are (virtually) damaged, so John replaces them with new sensors. John approaches vessel B to the system. The machine identifies correctly odour B without new training. The dificulty is the proprtion of sensors to be replaced.

```
sc.replace2 <- Scenario(name = "Sensor Replacement II",
  tunit = 60, concUnits = "norm", randomize = TRUE,
  T = c("A", "B", "C"), nT = 30, V = "B", nV = 30)

sc.replace2
```

```
##  Scenario `Sensor Replacement II` of 120 samples, tunit 60, randomize TRUE
##  - gases A, B, C
##  - Training Set: A (30), B (30), C (30)
##  - Validation Set: B (30)
```

```
plot(sc.replace2)
```



Session Information

```
sessionInfo()
```

```
## R version 3.0.1 (2013-05-16)
## Platform: x86_64-pc-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=C                 LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
```

```
## other attached packages:
##  [1] chemosensors_0.7.8 pls_2.3-0          gridExtra_0.9.1
##  [4] ggplot2_0.9.3.1.99 reshape2_1.2.2     plyr_1.8
##  [7] ascii_2.1          xtable_1.7-1       knitr_1.3.3
## [10] devtools_1.4.1.99
##
## loaded via a namespace (and not attached):
##  [1] codetools_0.2-8    colorspace_1.2-2   dichromat_2.0-0
##  [4] digest_0.6.3       evaluate_0.4.4     formatR_0.8
##  [7] gtable_0.1.2       httr_0.2           labeling_0.2
## [10] LearnBayes_2.12    MASS_7.3-27        memoise_0.1
## [13] munsell_0.4        parallel_3.0.1     proto_0.3-10
## [16] quadprog_1.5-5     RColorBrewer_1.0-5 RCurl_1.95-4.1
## [19] scales_0.2.3       stringr_0.6.2      tools_3.0.1
## [22] whisker_0.3-2
```

4 DISCUSSION

4.1 DISCUSSION OF THE RESULTS

In this work, the developed set up was used to mimick biological olfactory system and to address its different aspects in a bioinspired way. The engineering part of the work was centered on the assembly of the embedded computer and its integration with main biomimetic artifacts (developed by the collaborators): the large scale array of 65,536 conducting polymer sensors and the neuromorphic models implemented in the IQR framework. Validation of the biomimetic approach was accomplished through a broad variety of experiments, demonstrations and simulations. The main bioinspired experiment on odor localization and classification, including a complete description of the robotic set up, was reported in the book chapter [López et al. 53].

Simulations with synthetic data proposed in this work represent a research line complementary to experiments based on the real sensor arrays. The data simulation workflow consists of a scenario definition, virtual array parametrization, generation of sensor array data, and data processing with exactly the same neuromorphic models as used in the robotic set up. Three journal articles covered the main results on this research [Ziyatdinov et al. 109, 106, Ziyatdinov and Perera-Lluna 110].

The following paragraphs expose the main findings of this work and provide the discussion of the achieved results.

On the navigation experiments with embedded neuromorphic processing

Undoubtedly, the main experimental validation of the set up are the results obtained in the scenario of autonomous robotic odor source classification and localization [López et al. 53]. The demonstrated approach belongs to the group of strategies, which draw inspiration from biological organisms, in particular silkworm moths. Several biological organisms with relatively simple nervous systems (bees, ants, lobsters, male moths and others) are known to efficiently resolve the task on odor localization despite the number of difficulties of the real world, such as turbulent chemical plumes, obstacles or interfering odors.

The study performed in this work differs from other numerous studies (emulating the general behavior patterns of odor-seeking insects) in the use of neural simulations in guiding olfactory behavior [Rhodes and Anderson 77]. The localization capabilities of the developed robot were examined in the presence of two odors, whereas most of the current studies are limited to only one odor source. The implemented model of the AL allowed for odor-specific transient representation under the continuous excitation by the odor stimuli.

Robots implemented for odor source localization are remarkable examples of bioinspired solutions to cope with the challenges in chemical sensing [Huerta and Nowotny 40]. Practical difficulties arise from the part of the chemical sensors, for example, the turbulent nature of the odor information carrier, the long term instabilities like drift, lack of sensitivity, and slow response times of sensors required by mobile robots. The achieved classification results can be viewed as a proof of concept of odor classification with the AL model for the odor localization task. Further work is needed to explore different turbulence conditions and more complex configurations of the odor mixture with a larger number of trials.

It is worth noting that the array of 16 metal-oxide sensors was used in the experiment, since the large-scale array of conducting polymer sensors had not been fully tested on the robotic set up due to the lack of the resources. The special navigation data set collected for the two arrays – 16 metal-oxide sensors and 4096 conducting polymer sensors – is the first step towards future experiments with large-scale arrays with a possibility to exploit the high-dimensional and redundant chemosensor input data in application to the odor localization.

On the data simulations in machine olfaction

The series of three works, resulted in the final release of the data simulation tool, represent the main findings: the model of drift [Ziyatdinov et al. 109], the simplified sensor models estimated on the reference data set [Ziyatdinov et al. 108], and the basic parametrized examples of data simulation and data processing in machine olfaction [Ziyatdinov and Perera-Lluna 110].

The developed drift model extends the model based on a reference gas proposed by Artursson and colleagues [Artursson et al. 1]. The model is based on the common principal component analysis, which finds the drift variation in data jointly for several gas classes without a need for a reference gas. It is important to note that the proposed model can be applied to data collected from sensors based on other technologies, taking into account that a correspondent long-term and reliable reference data set is required.

The concept of parametrized simulated sensor array data was proposed for the first time in machine olfaction [Ziyatdinov and Perera-Lluna 110]. Three different use cases were highlighted in that work, and the user of the data simulation tool can define many more scenarios, which are basically limited by the basic properties of the reference data set (up to three analytes in mixtures and 17 different prototype sensor profiles).

On the laboratory experiments with simulation of the sniffing behavior

The experiments with modulation of the gas flow is still an on-going work, which attempts to emulate the sniffing behaviour in the olfactory system [Ziyatdinov et al. 106]. Sniffing, sampling odors actively, has been studied recently in neuroscience, and it has been suggested that the respiration frequency is an important parameter of the olfactory system. Animals can actively control different parameters of the respiration cycle such as frequency, amplitude and duration. For example, respiratory rates of rodents and other small mammals are in the range of 1–4 Hz when rest awake in a safety environment. This frequency increases rapidly (in only one respiration cycle) to 12 Hz when animals perform odor source localization, novel odorants exploration, or odor discrimination tasks [Verhagen et al. 96, Kay et al. 47]. In spite of many known examples related to the sniffing behavior, the computational advantages of high frequency sampling have not been yet elucidated [Wachowiak 99].

The proposed experimental set up features a mechanical ventilator to modulate the flow in the gas delivery system. The spectra of the modulated sensor signals contains two components: the low-frequency part, which demostrate a conventional response curve of a sensor in response to a gas pulse, and the high-frequency part, which have a clear principal harmonic at the frequency 0.08 Hz (the respiration frequency).

Samples of two pure analytes were collected in the first data set, and the explored question was whether any discrimination information in the modulated response exists.

Comparison between low-frequency and high-frequency components of the modulated response have been performed in the conference proceeding work [Ziyatdinov et al. 106], and the analyte-specific patterns in the high-frequency part of the signals have been qualitatively demonstrated. It was observed that the high-frequency features were likely to appear earlier in the course of the measurement.

A relatively broad combination of samples of binary mixtures and pure analytes were collected in the second data set. A quantitative analysis of these data will be accomplished in the future, in order to confirm the previously hinted early-detection performance of the high-frequency features. The future results are expected to motivate further investigation of the sensor system under different gas flow conditions, and to be valuable for system integration with the state-of-the-art of neuromorphic systems [Neftci et al. 62, Schmuker et al. 84].

5 CONCLUSIONS

1. The embedded computer has been assembled to conduct bioinspired experiments with a unique set of features: three types of chemical sensor arrays, the mobile platform with multimodal sensing capabilities; embedded computations in real time including the neuromorphic data processing; and the software development kit for virtualization of the set up.

2. The fully autonomous robotic set up has demonstrated stable performance on the odor localization and classification scenario in the controlled indoor wind tunnel environment. This robotic implementation is different from others in the implementation of embedded computations of the neural network model of the moth olfactory pathway.

3. The bioinspired experiments with gas flow modulation have shown an advantage of the modulated signals in terms of early availability of informative features. That is particularly beneficial in terms of the early detection scenario.

4. The novel multivariate model of drift estimates the long-term drift noise as a common variance among several classes. In the trivial case when the number of gas classes is one, the proposed model is exactly the same as the state-of-the-art model based on a reference gas. In general case when the number of components is greater than one, the model exploits the information from all classes and does not require a selection step of a reference class.

5. A set of simulation models have been designed to reproduce the reference long-term data set of 17 conducting polymer sensors. The proposed models developed under simplified assumptions are able to reproduce the reference data set. Virtual arrays are capable to generate sensor array data, which extend the reference data in terms of the number of sensors, up to three components in gas mixtures, and the structure and amount of the noise in data.

6. The developed data simulation tool has been introduced for the first time and fills the gap in conducting the synthetic experiments in machine olfaction. The tool is intended to be an alternative data source in the data analysis domain, since acquisition of a large, reliable and representative set of measurements is practically costly and time-consuming, especially when one is interested in biomimetic experiments at large scale, both in the number of sensors and the number of chemicals.

7. Ten scenarios for machine olfaction – classification, quantification, segmentation, habituation, event detection, novelty detection, drift compensation, and sensor replacement – have been designed and formalized in the framework of the data simulation tool. For three of these scenarios – classification, segmentation, and sensor replacement – synthetic benchmark data sets at different difficulty levels have been generated. The benchmarks are available to the community on the web page http://neurochem.sisbio.recerca.upc.edu/?page_id=257.

8. The full data set acquired under the gas flow modulation has been made available on the public repository with an objective to continue the joint effort in re-

search of biologically inspired systems. The data set is available on the web page https://archive.ics.uci.edu/ml/datasets/Gas+sensor+array+under+flow+modulation.

6 PUBLICATIONS

This chapter presents all publications derived in the course of the thesis.

Journal articles

1. **Ziyatdinov, A.**, Marco, S., Chaudry, A., Persaud, K., Caminal, P., & Perera, A. (2010). Drift compensation of gas sensor array data by common principal component analysis. Sensors and Actuators B: Chemical, 146(2), 460–465. doi:10.1016/j.snb.2009.11.034 [109]
2. **Ziyatdinov, A.**, Fernández Diaz, E., Chaudry, A., Marco, S., Persaud, K., & Perera, A. (2013). A software tool for large-scale synthetic experiments based on polymeric sensor arrays. Sensors and Actuators B: Chemical, 177, 596–604. doi:10.1016/j.snb.2012.09.093 [108]
3. **Ziyatdinov, A.**, & Perera-Lluna, A. (2014). Data Simulation in Machine Olfaction with the R Package Chemosensors. PLoS ONE, 9(2), e88839. doi:10.1371/journal.pone.0088839 [110]
4. Fernández-Albert, F., Llorach, R., Garcia-Aloy, M., **Ziyatdinov, A.**, Andrés-Lacueva, C., & Perera-Lluna, A. (2014). Intensity drift removal in LC/MS metabolomics by Common Variance Compensation. Bioinformatics, btu423. [19]

Book chapters

1. López, L., Vouloutsi, V., Chimeno Escudero, A., Marcos, E., Bermúdez i Badia, S., Mathews, Z., Verschure, P. F.M.J., **Ziyatdinov, A.** & Perera i Lluna, A. (2011). Moth-Like Chemo-Source Localization and Classification on an Indoor Autonomous Robot. In L. D. Pramatarova (Ed.), On Biomimetics. InTech. [53]
2. Vouloutsi, V., López, L., Mathews, Z., Chimeno Escudero, A., **Ziyatdinov, A.**, Perera i Lluna, A., Bermúdez i Badia, S. & Verschure, P. F. M. J. (2013). The Synthetic Moth: A Novel Neuromorphic Approach towards Artificial Olfaction in Robots. In K. C. Persaud, S. Marco, & A. Gutierrez-Galvez (Eds.), Neuromorphic Olfaction. CRC Press. [98]

Conference proceedings

1. **Ziyatdinov, A.**, Chaudry, A., Persaud, K., Caminal, P., & Perera, A. (2009). Common principal component analysis for drift compensation of gas sensor array data. In Olfaction and electronic nose: Proceedings of the 13th International Symposium on Olfaction and Electronic Nose (Vol. 1137, pp. 566–569). [105]
2. Perera, A., & **Ziyatdinov, A**. (2011). A Large Scale Virtual Array of Non-selective Odour Sensors for the Neurochem Project. In Neurochem's Workshop on Bioinspired Computation for Chemical Sensing. [68]

3. **Ziyatdinov, A.**, Calvo, J. M. B., Lechón, M., Bermúdez i Badia, S., Verschure, P. F. M. J., Marco, S., & Perera, A. (2011). Odour Mapping Under Strong Backgrounds With a Metal Oxide Sensor Array. In AIP Conf. Proc. (Vol. 232, pp. 232–233). doi:10.1063/1.3626371 [104]

4. **Ziyatdinov, A.**, Eduard Fernández-Diaz, Chaudry, A., Marco, S., Persaud, K., & Perera, A. (2011). A Large Scale Virtual Gas Sensor Array. In AIP Conf. Proc. 1362 (pp. 151–152). doi:http://link.aip.org/link/doi/10.1063/1.3626339 [107]

5. Kanaan-Izquierdo, S., **Ziyatdinov, A.**, Massanet, R., & Perera, A. (2012). Multiview approach to spectral clustering. In 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 1254–1257). IEEE. doi:10.1109/EMBC.2012.6346 [44]

6. **Ziyatdinov, A.**, Fernandez, L., Gutierrez-Galvez, A., Fonollosa, J., Marco, S., & Perera, A. (2013). A qualitative study of a bioinspired sensor system based on gas flow modulation by an artificial lung apparatus. In 15th International Symposium on Olfaction and electronic nose. [106]

Other publications

1. Kanaan-Izquierdo, S., **Ziyatdinov**, A., Massanet, R., & Perera, A. (2013). Patent Proposal: A computer spectral clustering improved method and uses thereof. Spain: Oficina Española de Patentes y Marcas. [45]

A DATA SETS

Three data sets presented in the thesis are a part of a collection of data sets measured in the Neurochem project. Measurement of these three data sets are results of joint work mainly distributed between four partners of the Neurochem project:

- Polytechnic University of Catalonia (UPC), Barcelona, Spain;
- Pompeu Fabra University (UPF), Barcelona, Spain;
- University of Barcelona (UB), Barcelona, Spain;
- Consiglio Nazionale delle Ricerche (CNR) at Institute for Microelectronics and Microsystems, Rome, Italy.
- The University of Manchester (UNIMAN), Manchester, United Kingdom

Table A.1 shows the contribution of each partner in collecting the data sets. UPC led the work on the three data sets, and the contributions assigned to UPC are mostly matched with the contributions derived in the thesis.

Contribution	UPC	UPF	UB	CNR+UNIMAN
Assembling embedded computer	S2, S3			
Assembling robotic infrastructure		S3		
Design of sensor array	S1		S2	S3
Data acquisition electronics	S2, S3		S2, S3	S3
Data acquisition software	S1, S2, S3			
Conducting experiments	S1 S2, S3	S3	S2, S3	

Table A.1: Contributions by the UPC, UPF, UB and CNR partners of the Neurochem project in collecting the data sets S1, S2 and S3. The data sets are shown on the Figure 3.1 as orange circles with the letter S.

A.1 S1 BENCHMARKS

Synthetic benchmarks were an alternative to the real measurements at the middle stage of the Neurochem project, in order to run simulations of the olfactory system. The realization of the synthetic experiments required a model of an array of gas sensors. That model needed to capture the main features shown by polymer sensors (the reference data set was measured with an array of conducting polymer sensors) and be simple enough so that it could be included in the system software. The model was implemented in the data simulation tool (the R package *chemosensors*) designed in the course of the thesis.

The benchmark data were generated on May 28, 2011. The main web page with the summary and the links is on http://neurochem.sisbio.recerca.upc.edu/?page_id=257.

The distributed data include the raw data in the two formats of *RData* and *csv* and the documentation files in *PDF* format.

Scenario	Letter	Date	Difficulty	Classes (T)	Classes (V)
Classification	S1	28 March 2011	1	A, C	A, C
			2	A17C83, A83C17	A17C83, A83C17
			3	A33C67, A67C33	A33C67, A67C33
			4	A40C60, A60C40	A40C60, A60C40
			5	A45C55, A55C45	A45C55, A55C45
Segmentation	S1	28 March 2011	1	A, C	A50C50
			2		A45C55
			3		A60C40
			4		A67C33
			5		A83C17
Sensor Damage	S1	28 March 2011	1 (6.25%)	A33C67, A67C33	A33C67, A67C33
			2 (12.5%)		
			3 (18.75%)		
			4 (25%)		
			5 (31.25%)		

Table A.2: Description of three benchmark data sets produced in the course of the thesis. The sensor array consisted of 1020 virtual sensors in the framework of the data simulation tool under version 0.4.3. The number of samples per class for either training and validation sets was 30.

The benchmarks were generated under the version 0.4.3 of the data generation tool (the R package *chemosensors*). The main features of the simulation models were implemented in the given early version of the tool:

- 17 types of sensors based on the profiles of the 17 CP sensors from the reference data set;
- the steady-state sensor model consisted of two parts:
  - the linear model based on partial least squares (PLS) regression;
  - the non-linear model based on the EL;
- the transient sensor model based on the auto-regressive (AR) filter of the 2nd order;
- the three noise models
  - the drift model (additive noise, multi-component);
  - the sensor aging model (multiplicative noise);
  - the concentration noise model (gas camera).

Three scenarios were included in the final release of the benchmarks: classification, segmentation and sensor damage. Table A.2 contains the basic summary for these scenarios.

The complete list of 10 scenarios designed in the scope of the Neurochem project is available in the Supporting Information, File S1 [Ziyatdinov and Perera-Lluna 110], which copy is presented in Section 3.4.

The scenarios are parametrized by difficulty levels (from 1 to 5). For most of the scenarios the difficulty is defined in terms of a similarity among gas classes or components in gas mixtures, for example, classification and segmentation in the Table A.2. The class labels consist of two parts: the gas letter A, B or C and the concentration value in percents of the maximum concentration value, at which the simulated sensors are in the saturation regime and their response does not change at higher concentrations. The difficulty of sensor damage scenario is defined by the proportion of damaged sensors in the array that were simulated to not respond in the validation set.

A.2 S2 LABORATORY DATA SETS

The data sets were measured on the static set up in the laboratory, and the robotic equipment was not used. The array consisted of 16 metal-oxide sensors of 5 different TGS models by Figaro Inc. under one of two constant heating voltages (3.3 V and 5.5 V). The sensors were selected heuristically so that the sensor transients were able to follow the flow dynamics. A custom printed circuit board was designed to read out signals from the sensors. The sensors were placed in a 70 ml chamber connected to a mechanical ventilator. The odor delivery system was combined with an external mechanical ventilator to simulate the biological respiration cycle. The purpose of the experiment was to emulate the sniffing behavior, sampling odors actively, in the olfactory system. More information on the experiments can be found in the conference proceeding [Ziyatdinov et al. 106].

Data set	Letter	Date	N samples	N classes	Classes (repetitions)
pulmon5	S2	21 Apr 2011	33	5	ace-1 (10), eth-1 (10), ace-0.2-eth-1 (3), ace-0.5-eth-1 (3), ace-1-eth-1 (3), air (4)
pulmon6	S2	17-20 May 2011	58	12	ace-0.1 (6), ace-0.3 (6), ace-1 (3), eth-0.1 (6), eth-0.3 (4), eth-1 (5), ace-0.1-eth-0.1 (4), ace-0.1-eth-0.3 (5), ace-0.3-eth-0.1 (5), ace-0.1-eth-1 (3), ace-1-eth-0.1 (3), air (8)

Table A.3: Description of two laboratory data sets acquired in the course of the thesis. The sensor array consisted of 16 metal-oxide sensors, and the gas flow was modulated by the artificial lung apparatus.

Two data sets, referred to as pulmon5 and pulmon6, have been collected. The first data set pulmon5 mainly contains samples of two pure analytes (dilutions), acetone at 1% vol. and ethanol at 1% vol., that allows to conduct a data exploratory analysis for searching data patterns that are analyte-specific in the two-class problem. The second data set pulmon6 is targeted to a more complex discrimination task than a separation of two gas classes. The measured classes, 12 in total, form a relatively broad combination of acetone and ethanol analytes in binary mixtures. Each analyte takes three concentrations values 0.1% vol., 0.3% vol. and 1% vol. that marks the change of at most one order of the magnitude. Table A.3 presents the summary of the two collected data sets pulmon5 and pulmon6.

A.3 S3 NAVIGATION DATA SETS

The navigation experiment was conducted in the wind tunnel space under controlled conditions. One or two sources of vapourized analyte dilutions were placed on one side of the tunnel, while several mechanical ventilators were placed on the other side of the tunnel, in order to produce the air flow. The robotic set up explored the chemical space by following a trajectory of 15 spot points evenly distributed in the tunnel space. The robotic set up was stopped for 60 s at each spot point to perform the measurements from the sensor array. More information on the experimental set up is available in [López et al. 53], which copy is presented in Section 3.5.

Two sensor arrays were used in the set up in turns: the array of 16 metal-oxide sensors and the array of 4096 conducting polymer sensors. In addition to the two arrays, the ion-mobility spectrometry (IMS) device performed the measurements simultaneously with each array. The IMS has a higher resolution to quantify the chemicals, and, thus, IMS data served as a reference to capture the chemical odor map. The description of the data acquired from three devices is given in Table A.4.

Device	Letter	Date	N samples	N classes	Classes (repetitions)
16 MOX	S3	11-13 Jul 2011	11	8	ace-0.01 (1), ace-0.1 (1), eth-0.03 (1), eth-0.1 (1), eth-1 (1), ace-0.01-eth-0.01 (2), ace-0.1-eth-0.1 (2), ace-0.1-eth-0.1 (2)
4096 CP	S3	13 Jul 2011	6	5	ace-0.01 (1), eth-0.01 (1), eth-0.03 (1) ace-0.01-eth-0.01 (2), ace-0.1-eth-0.1 (1)
IMS	S3	11-13 Jul 2011	15	9	ace-0.01 (1), ace-0.1 (4), eth-0.01 (1), eth-0.03 (1), eth-0.1 (1), eth-1 (1), ace-0.01-eth-0.01 (2), ace-0.1-eth-0.1 (2), ace-0.1-eth-1 (2)

Table A.4: Description of the navigation data sets acquired in the thesis.

B DEMONSTRATIONS

Four demonstrations have been released in the course of the thesis. The demonstrations are shown on Figure 3.1 as blue circles with the letter *D*: D1, D2, D3 and D4.

The demonstrations represent the results on the development of the biomimetic set up at the middle and final stages of the Neurochem project. The first two demonstrations D1 and D2 were completed at the middle stage of the Neurochem project, when real sensor devices were under development and the only array available was an array of virtual sensors. Demonstrations D1 and D2 were presented on the second review meeting of the project in Brussels, on February 2010. The last two demonstrations D3 and D4 reported the ultimate results achieved at the final stage of the Neurochem project. Demonstrations D3 and D4 were presented on the final review meeting of the project in Barcelona, on June 2011.

The demonstrations are results of the joint work mainly distributed between UPC and UPF partners of the Neurochem project. Table B.1 shows the contribution of each partner in development of the demonstrations. The contributions assigned to UPC are mostly matched with the contributions derived in the thesis.

Contribution	UPC	UPF
Assembling embedded computer	D2, D3, D4	
Assembling robotic infrastructure		D2, D4
Chemosensor IQR modules	D1, D3, D4	
Robotic IQR modules		D2, D4
Neuromorphic IQR modules		D1, D4
Conducting experiments	D1, D2, D3	D1, D2, D4

Table B.1: Contributions by the UPC and UPF partners of the Neurochem project in development of the demonstrations D1, D2, D3 and D4.

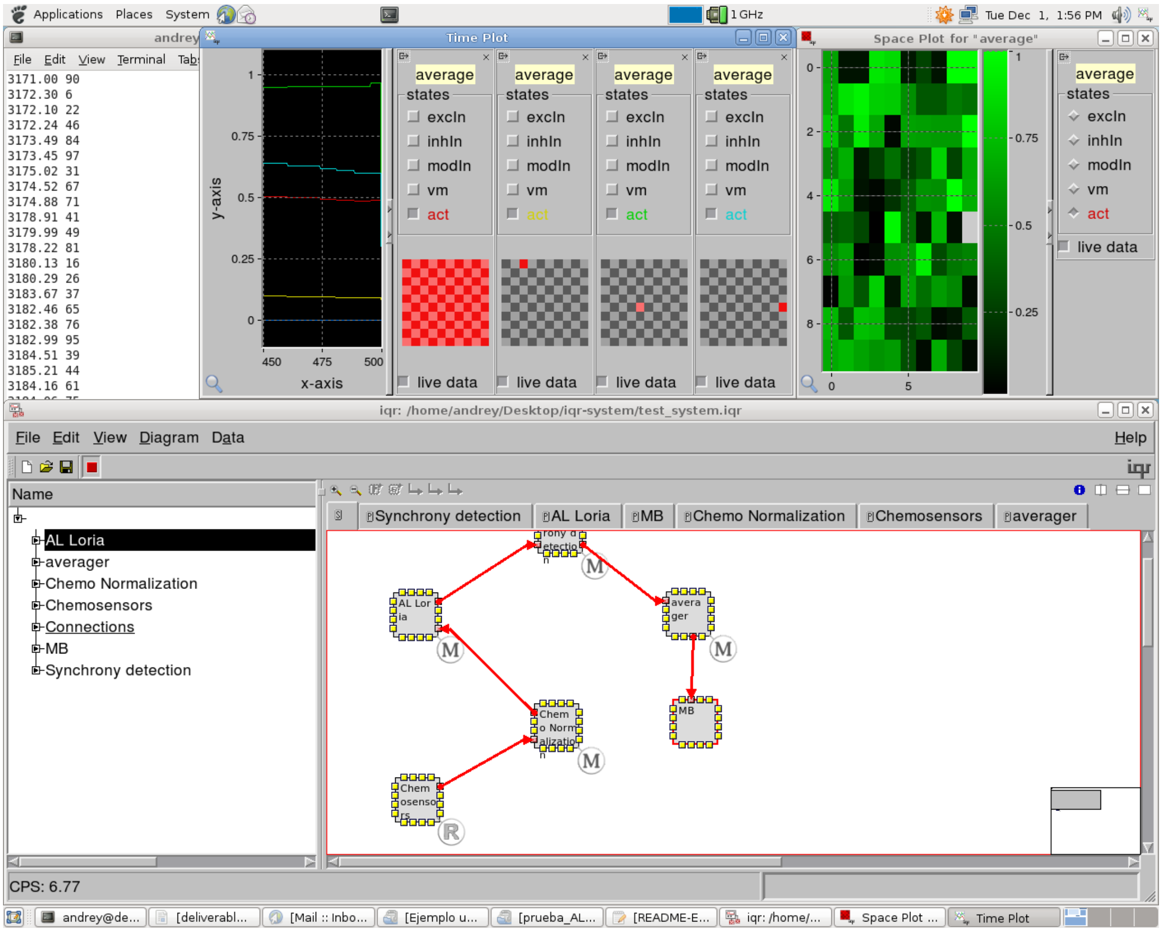
B.1 D1 IQR

This demonstration presents the preliminary models of the insect olfactory system implemented in the framework of the IQR neuronal simulator [Bernardet et al. 7]. Two main system models of the Antennal Lobe (AL) and Mushroom Body (MB) have been adapted to process dynamic and analog input data from simulated chemical sensors, displaying some of the system properties such as the classification of presented chemical stimulus. Both the models and the IQR software are included in the Neurochem OS image described in Appendix C.

Chemical sensors have been simulated at the early development stage of the data simulation tool of the thesis, when the tool was a list of R scripts, rather than an R package. The chemical sensor simulator takes as input an arbitrary number of chemical stimuli and their concentrations at every time step of the simulation. Then it returns the simulated response of a user-defined number of sensor elements. In the simulation of the demonstration D1, the actual computation of the responses of the sensors is carried out offline (the sensor responses are pre-computed), and, thus, it does not induce any computational limitations for the rest of the integrated system.

The AL module takes as input sensory information of up to one hundred simulated sensors and returns the spiking activity generated by each one of its simulated Projection Neurons (PNs) and the Local Field Potential (LFP) of the whole population of PNs. It was shown that the information provided by the implemented AL model is sufficient to decode the PNs that show a phase locked activity with the LFP, which in turn encodes the stimulus properties and represents the information needed by the Mushroom Body (MB) module.

The MB module provides the interface between IQR and the simplified MB model (the complete model of the Neurochem project). It essentially takes the binary synchrony pattern provided by the AL module as input and associates it to the closest of a set of pre-trained patterns.



```
3171.00 90
3172.30 6
3172.10 22
3172.24 46
3173.49 84
3173.45 97
3175.02 31
3174.52 67
3174.88 71
3178.91 41
3179.99 49
3178.22 81
3180.13 16
3180.29 26
3183.67 37
3182.46 65
3182.38 76
3182.99 95
3184.51 39
3185.21 44
3184.16 61
```

CPS: 6.77

Figure B.1: The screenshot of the desktop when the demonstration D1 was executed on the set up.

Figure B.1 (the bottom panel) shows the IQR system of the integrated insect olfactory pathway. The chemosensor simulator module of 100 sensors generates simulated transient signals in response to two gases A and C (no mixtures) (Time Plot on the upper panel). Then the signals go through an initial process of normalization in order to adapt them to the amplitudes used by the AL model. The AL processes a high dimensional input to encode the concentration and amplitude of the stimuli by means of the AL's PN action potentials. Once the synchronous PNs with the LFP are identified, an averaging process uses the information of several LFP oscillations to improve the signal-to-noise ratio (Space

Plot on the upper panel). Finally, it is the task of the MB to find the closest matching of the AL responses to its training set.

The results of the demonstration can be assessed by opening Space Plot of the MB module, where three class-specific patterns (no gas, gas A or gas C) can be observed.

B.2 D2 ROBOTIC PLATFORM

This demonstration shows the assembled robotic set up of the Neurochem project in action. The robot consists of two parts: the embedded computer developed by UPC and the robotic platform developed by UPF. The full description of the robotic platform is given in [López et al. 53].

The robot explored the space in the wind tunnel by receiving the commands from the joystick device via bluetooth. The IQR simulator was the top-level environment for processing the data coming from various sensors, running the behavioral models and controlling the hardware devices including the motor. Figure B.2 shows the screenshot of the desktop when the demonstration D2 was executed on the set up.

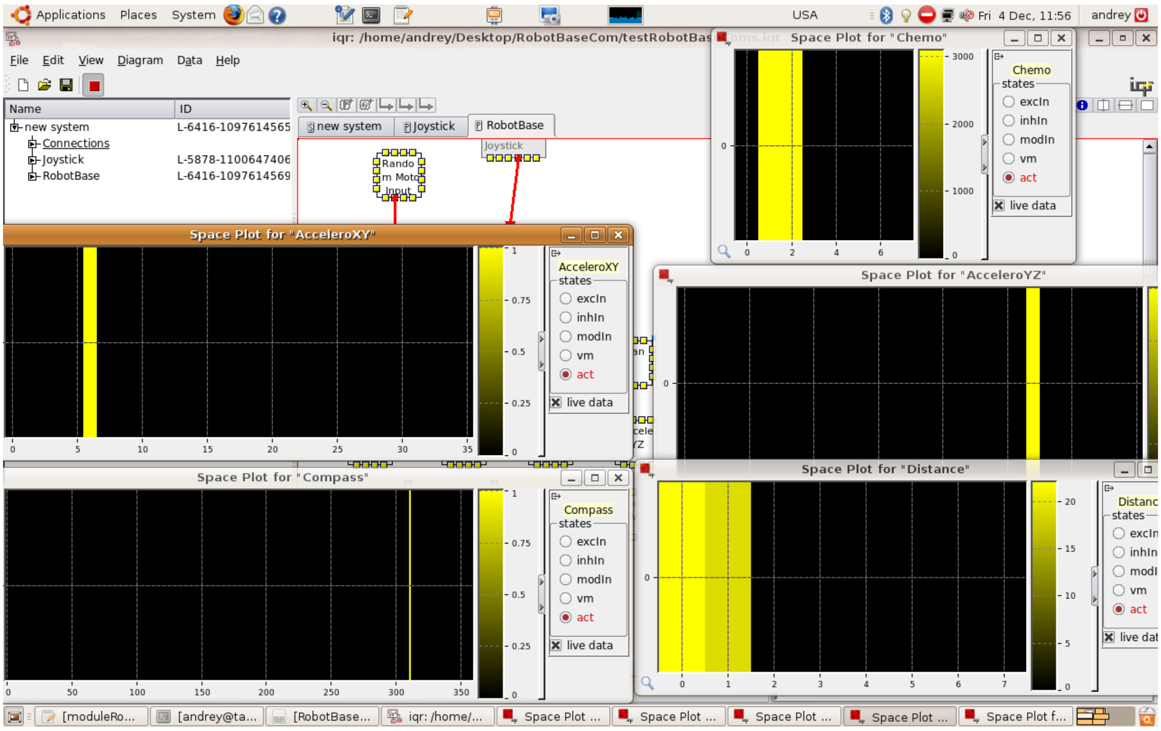

Applications Places System
USA
Fri 4 Dec, 11:56
andrey
iqr: /home/andrey/Desktop/RobotBaseCom/testRobotBas
File Edit View Diagram Data Help
Name ID
new system L-6416-1097614565
Connections
Joystick L-5878-1100647406
RobotBase L-6416-1097614569
new system
Joystick
RobotBase
Random Motor Input
Space Plot for "Chemo"
Chemo
states
excIn
inhIn
modIn
vm
act
live data
Space Plot for "AcceleroXY"
AcceleroXY
Space Plot for "AcceleroYZ"
Space Plot for "Compass"
Compass
Space Plot for "Distance"
Distanc
[moduleRo... [andrey@ta... [RobotBase... iqr: /home/... Space Plot ... Space Plot ... Space Plot ... Space Plot ... Space Plot f...

Figure B.2: The screenshot of the desktop when the demonstration D2 was executed on the set up.

The video of the demonstration is available on http://neurochem.sisbio.recerca.upc.edu/?p=19. The video was created by the SPECS laboratory, UPF, Barcelona, on November 2009.

B.3 D3 LARGE-SCALE ARRAY

This demonstration was designed to show the real-time acquisition from the large-scale CP array of 4096 sensors. The design of the sensors was performed by UNIMAN and CNR partners in the Neurochem project, and its detailed description is available in the internal documentation of the project. The D3 demonstration shows the results of the work performed by UPC partner of the project related to the thesis. In particular, these results include the data acquisition on the side of the embedded computer (interfaced

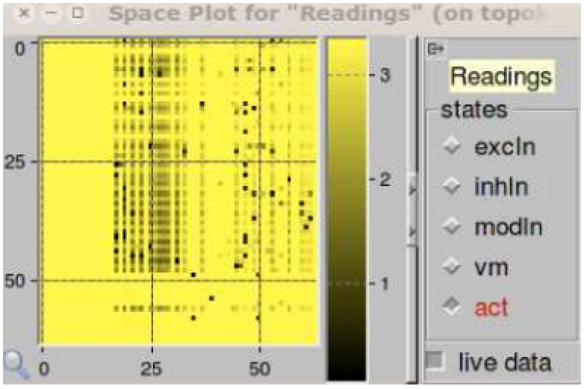


Figure B.3: The screenshot of the Space Plot window in the IQR simulator when the demonstration D3 was executed on the set up. The heatmap shows the raw readout in volts from the CP array of 4096 sensors.

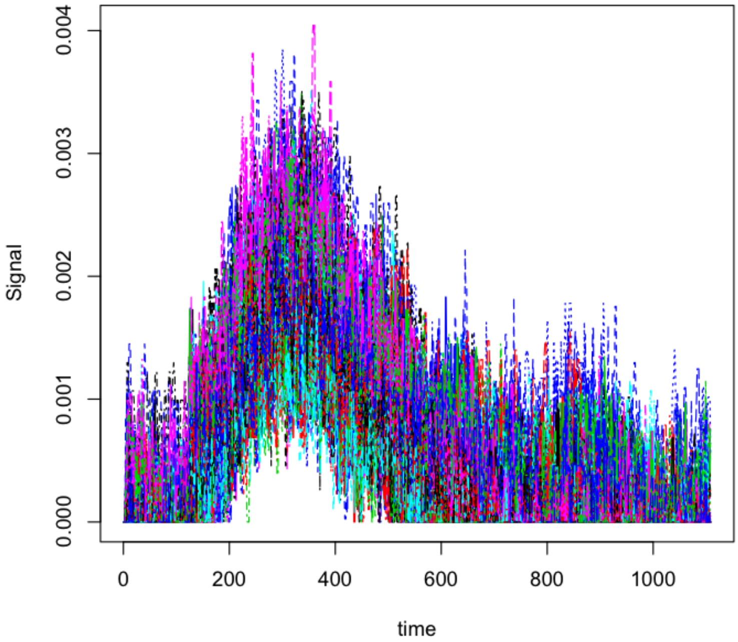


Figure B.4: The response to a gas pulse of the ethanol analyte at a certain concentration shown for 4096 sensors in the CP array.

with the scanning boards of the array produced by CNR partner) and integration of the input sensor array data on the top-level in IQR environment.

Figure B.3 shows the heatmap of the acquired large-scale data visualized in the real time by the IQR simulator on the Space Plot window. The measured signal is the voltage on the sensors in the range from 0 to 3.3 V. The data acquisition frame rate is 1 Hz, and it has been required to conduct the neuromorphic simulations in the real time. In this demonstration only one out of 16 boards with the sensor array is connected to the embedded computer, but this acquisition rate is possible for the whole modular sensor array, since the acquisition board of the embedded computer is able to acquire data from 16 channels in parallel.

One of the unique features of the large-scale array is its diversity. The total number of 31 polymer materials were used.The resistance of the sensors ranges from less than 1KΩ to several MΩ. A large group of sensors have very the high resistances (more than 1MΩ), as shown on Figure B.3. Figure B.4 shows the response to a gas pulse of a small subset from 4096 sensors of the CP array.

B.4 D4 NAVIGATION

The robotic platform was re-designed after the D2 demonstration due to mechanical issues and a excessive robot speed. The robotic platform is equipped with multimodal sensing capabilities (olfaction, compass, wind sensing, ultrasound, collision detection and vision). The robot is able to navigate autonomously in the wind tunnel to find an odor source and classify it. Figure B.5 shows the new version of the robot.

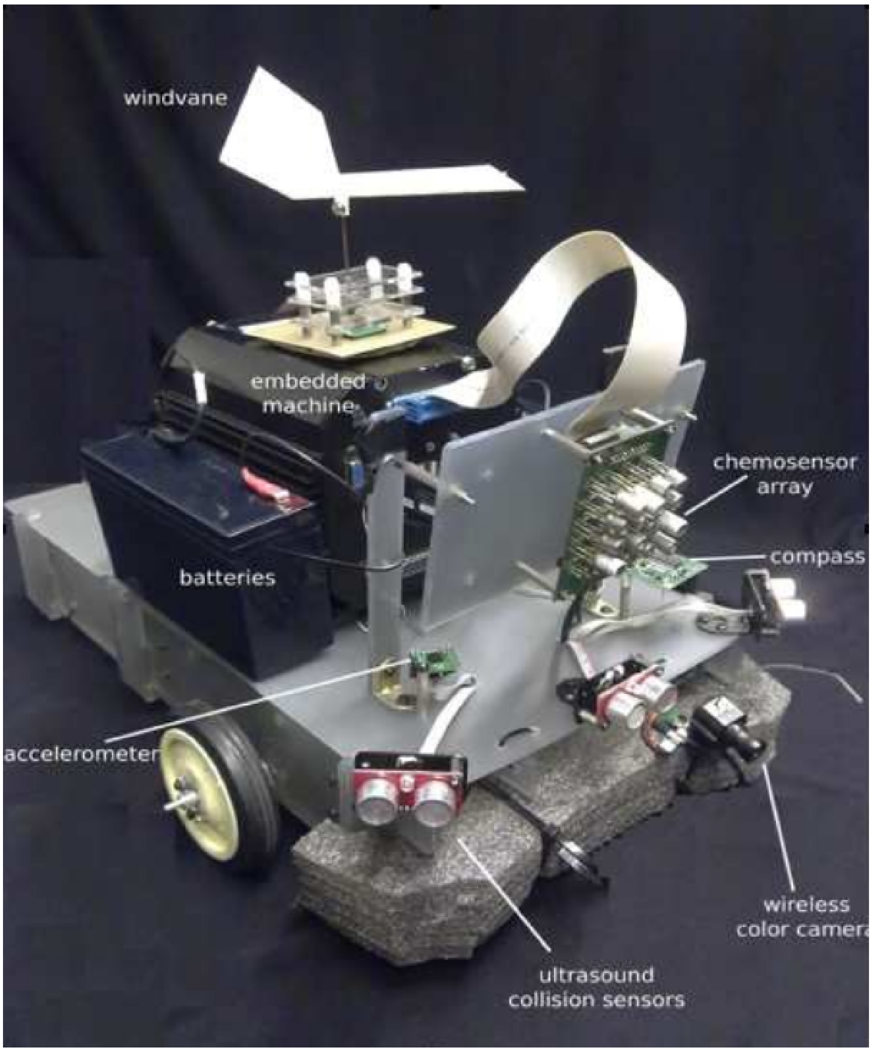


Figure B.5: The final robot with the new mobile platform used in the demonstration D4. Figure source: [Marco et al. 56].

The robot is driven by the model of the moth *cast-and-surge* strategy. The AL model previously developed for the demonstration D1 combined with temporal population coding is used to classify the odors. The visual landmarks are combined with the chemical cues and movement vector to learn landmark-to-landmark trajectories. The robot operates in the wind tunnel containing multiple odor sources. Thereby, the robot starts at the down-wind end of the tunnel and search for odor cues in the upwind direction. Simultaneously, it will run the odor classification model and also the multimodal-landmark learning mechanism. Based on the classified odor source, the robot is able to initiate a particular behavior, for example, attraction or aversion.

C NEUROCHEM IMAGE

The Neurochem software of the set up includes the drivers of the hardware components, the Neurochem library to interface the electronics for acquisition of the sensor signals, the IQR neuronal simulator, the neuromorphic IQR modules, the IQR modules for control of the robot, and the data simulation tool. The Neurochem software was especially designed for the Debian Linux operating system. Hence, the design of a custom Debian-based operating system image (the Neurochem Image) was a natural step in the software development.

The motivation to use the Neurochem Image is to automate the process to set-up, to configure and to release the operating system within the Neurochem software. The usage of the image is oriented towards the end-users of the software, who are free from routine on resolving technical issues related to compatibility of the Neurochem software and the operating system. The users are supposed to run the system via the Neurochem Image *neurochem.img* and get started to use the software right away.

Debian is a free operating system based on the Linux kernel and distributed with a broad variety of software packages. The Neurochem Image is a medium that stores a custom Debian operating system especially designed for the Neurochem project.

The main characteristics of the Neurochem Image:

- The original image format is *IMG*, that is an archive format for digital storage, transmission and replication of different storage media. The file name of the image is *neurochem.img*.
- The file size is 563 MB and fits to a typical size of CD.
- Supported media for the image are CDs/DVDs, Hard drives, USB drives and Flash drives.

The main characteristics of the operating system Debian used in the image:

- The distribution is Debian GNU/Linux 5.0.1.
- The Kernel Linux is 2.6.26-2-686 (i686).
- The C Library is GNU C Library version 2.7 (stable).

One of the main work directions in the thesis was the virtualization of the set up, and the Neurochem Image is the final product of this work. Let us imagine that the user is running any operating system Windows/Macintosh/Linux on his/her PC. If this computer supports the emulator program such as Virtual Box, the user will be able to access the Neurochem software by loading the image as an independent virtual operating system.

An example of the user session with the Neurochem image may be the following.

1. The user runs the program Virtual Box and pushes the button ”Start” to initiate the virtual system.

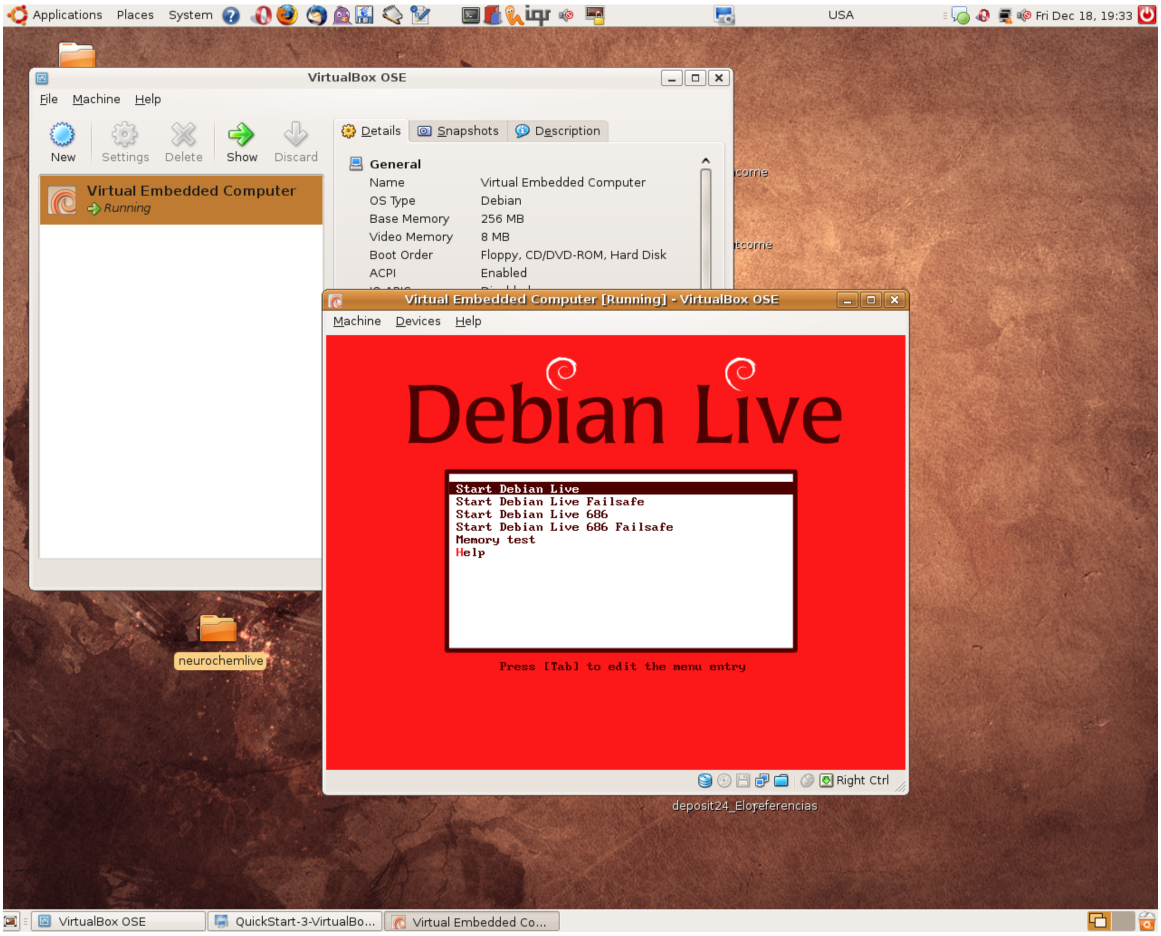
2. The user should check if the Neurochem image *neurochem.vdi* is specified as a primary hard drive for the virtual system.

3. In a new window, the user sees the boot menu and is asked to select the option *Start Debian Live* in the menu (Figure C.1).

4. The virtual system is booting, and the user will be automatically login with the username *neurochem*. Now the user sees the Desktop of the virtual system.

5. The user runs the IQR simulator by executing a special command on the Unix shell: iqr -r -f data/modelRM1/AcqDevice.iqr -c data/modelRM1/readings.conf (Figure C.2). Another option to run the IQR simulator might be executing any of the Desktop shortcut prepared in the Neurochem image (one of these shortcuts contains the IQR command given before).

6. The results of the execution of the command given on the previous step will be the main window of the IQR simulator with the system that simulates a virtual array of 17 sensors (specified in the second parameter of the command -f). In addition, the user will observe Space Plot for all sensors and Time Plot for 3 selected sensors (specified in the third parameter of the command -c) (Figure C.3).



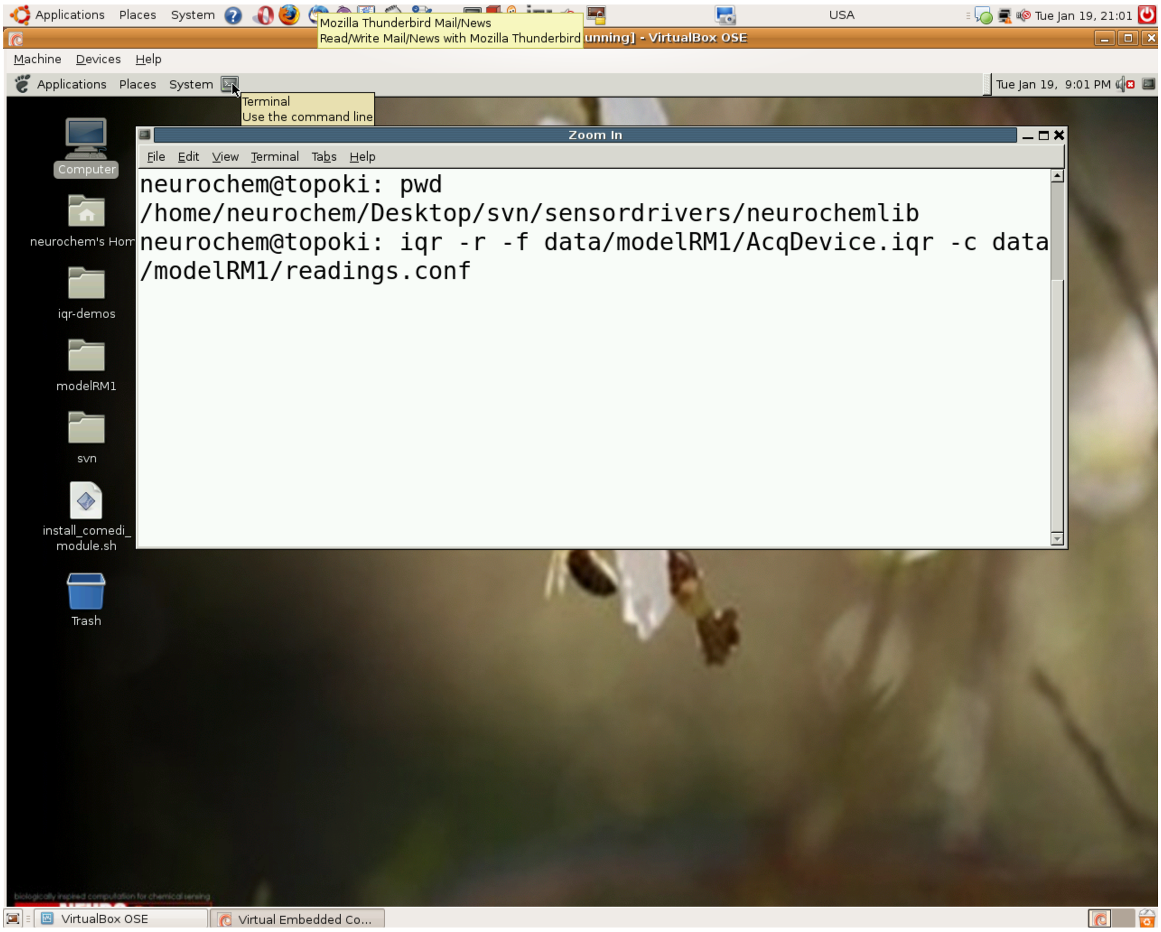
General

Name	Virtual Embedded Computer
OS Type	Debian
Base Memory	256 MB
Video Memory	8 MB
Boot Order	Floppy, CD/DVD-ROM, Hard Disk
ACPI	Enabled

```
Start Debian Live
Start Debian Live Failsafe
Start Debian Live 686
Start Debian Live 686 Failsafe
Memory test
Help

Press [Tab] to edit the menu entry
```

Figure C.1: The boot menu for the Neurochem Image of the virtual system via the Virtual Box program.



```
neurochem@topoki: pwd
/home/neurochem/Desktop/svn/sensordrivers/neurochemlib
neurochem@topoki: iqr -r -f data/modelRM1/AcqDevice.iqr -c data
/modelRM1/readings.conf
```

Figure C.2: The Unix shell to run the IQR simulator on the virtual system of the Neurochem Image.

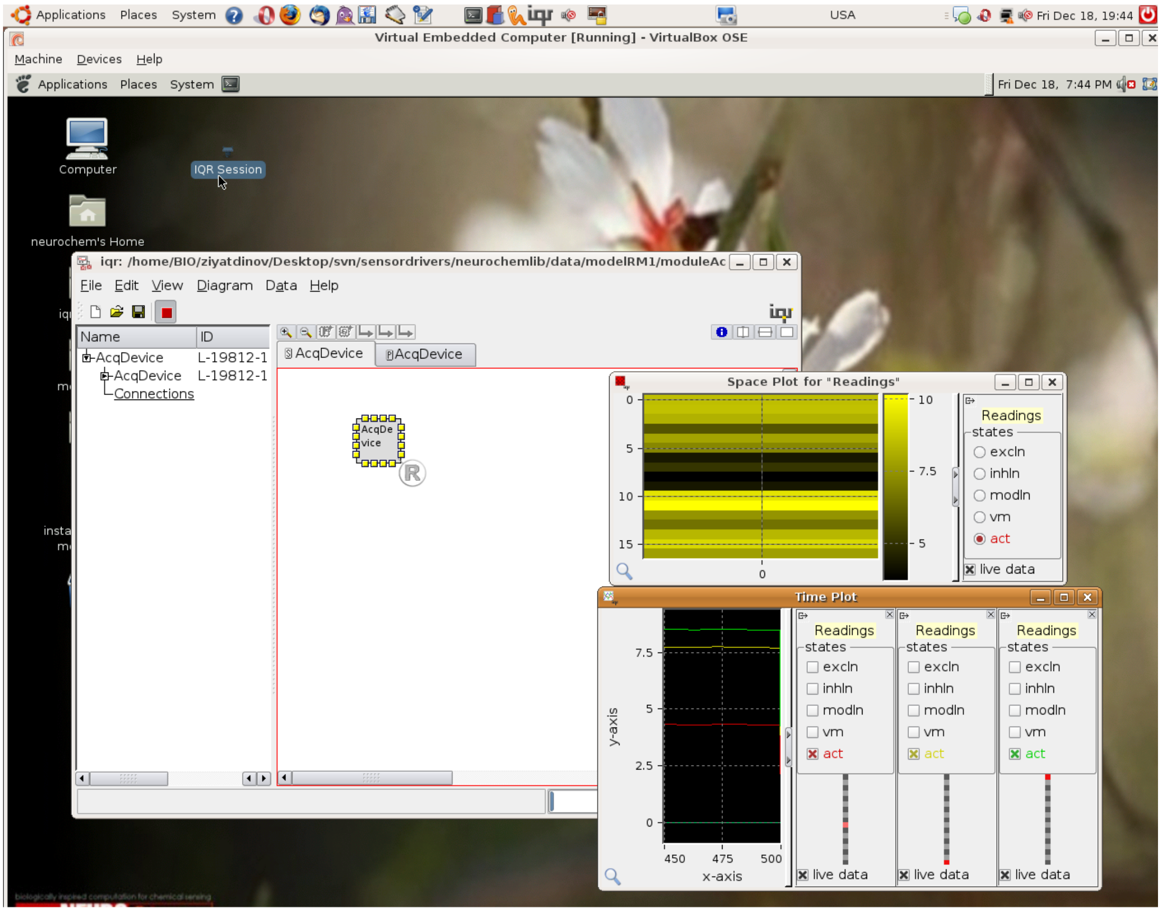


Figure C.3: An example of IQR simulation on the the virtual system of the Neurochem Image.

BIBLIOGRAPHY

[1] Artursson, T., Eklov, T., Lundstrom, I., Martensson, P., Sjostrom, M., and Holmberg, M. (2000). Drift correction for gas sensors using multivariate methods. *Journal of Chemometrics*, 14(5-6):711–723.

[2] Axel, R. (1995). The molecular logic of smell. *Scientific American*, 273(4):154–159.

[3] Bache, K. and Lichman, M. (2013). UCI machine learning repository.

[4] Bai, R. and Yang, R. T. (2001). A thermodynamically consistent langmuir model for mixed gas adsorption. *Journal of Colloid and Interface Science*, 239(2):296 – 302.

[5] Balaban, M. O., Korel, F., Odabasi, A. Z., and Folkes, G. (2000). Transportability of data between electronic noses : mathematical methods. *Sensors And Actuators*, 71.

[6] Bermak, A., Belhouari, S., Shi, M., Martinez, D., et al. (2005). Pattern recognition techniques for odor discrimination in gas sensor array. *The Encyclopedia of Sensors*.

[7] Bernardet, U., Blanchard, M., and Verschure, P. F. (2002). Iqr: a distributed system for real-time real-world neuronal simulation. *Neurocomputing*, 44:1043–1048.

[8] Beyeler, M., Stefanini, F., Proske, H., Galizia, G., and Chicca, E. (2010). Exploring olfactory sensory networks: simulations and hardware emulation. In *2010 Biomedical Circuits and Systems Conference (BioCAS)*, pages 270–273. IEEE.

[9] Bissell, R. A., Persaud, K. C., and Travers, P. (2002). The influence of non-specific molecular partitioning of analytes on the electrical responses of conducting organic polymer gas sensors. *Physical Chemistry Chemical Physics*, 4(14):3482–3490.

[10] Buck, L. and Axel, R. (1991). A novel multigene family may encode odorant receptors: A molecular basis for odor recognition. *Cell*, 65(1):175 – 187.

[11] Carmel, L. (2003). A feature extraction method for chemical sensors in electronic noses. *Sensors and Actuators B: Chemical*, 93(1-3):67–76.

[12] Cerrato Oliveros, M. (2002). Electronic nose based on metal oxide semiconductor sensors as a fast alternative for the detection of adulteration of virgin olive oils. *Analytica Chimica Acta*, 459(2):219–228.

[13] Clifford, P. and Tuma, D. (1983a). Characteristics of semiconductor gas sensors I. Steady state gas response. *Sensors & Actuators*, pages 233–254.

[14] Clifford, P. and Tuma, D. (1983b). Characteristics of semiconductor gas sensors II. Transient response to temperature change. *Sensors & Actuators*, pages 255–281.

[15] Davide, F. A. M., Natale, C. D., and Amico, A. D. (1994). Self-organizing multisensor systems for odour classification: internal categorization, adaptation and drift rejection. *Sensors and Actuators B: Chemical*.

[16] Distante, C., Leo, M., Siciliano, P., and Persaud, K. C. (2002a). On the study of feature extraction methods for an electronic nose. *Sensors And Actuators*, 87:274–288.

[17] Distante, C., Siciliano, P., and Persaud, K. C. (2002b). Self-Organising Maps. *Pattern Analysis & Applications*, pages 306–315.

[18] Eklöv, T., Martensson, P., and Lundstrom, I. (1999). Selection of variables for interpreting multivariate gas sensor data. *Analytica Chimica Acta*, 381(2-3):221–232.

[19] Fernández-Albert, F., Llorach, R., Garcia-Aloy, M., Ziyatdinov, A., Andres-Lacueva, C., and Perera, A. (2014). Intensity drift removal in lc/ms metabolomics by common variance compensation. *Bioinformatics*, page btu423.

[20] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

[21] Fort, A., Gregorkiewitz, M., Machetti, N., Rocchi, S., Serrano, B., Tondi, L., and Ulivieri, N. (2002). Selectivity enhancement of SnO 2 sensors by means of operating temperature modulation. *Thin Solid Films*, 418:2–8.

[22] Freund, Y. and Mansour, Y. (1997). Learning under persistent drift. In *Computational Learning Theory*, pages 109–118. Springer.

[23] Fryder, M., Holmberg, M., Winquist, F., and Lundstrom, I. (1995). A calibration technique for an electronic nose. In *Solid-State Sensors and Actuators, 1995 and Eurosensors IX.. Transducers' 95. The 8th International Conference on*, volume 1, pages 683–686. IEEE.

[24] Gardner, J., Bartlett, P., and Pratt, K. (1995). Modelling of gas-sensitive conducting polymer devices. *IEE Proceedings - Circuits, Devices and Systems*, 142(5):321.

[25] Geng, Z., Yang, F., and Wu, N. (2011). Optimum design of sensor arrays via simulation-based multivariate calibration. *Sensors and Actuators B: Chemical*, 156(2):854–862.

[26] Gibson, T. (1997). Detection and simultaneous identification of microorganisms from headspace samples using an electronic nose. *Sensors and Actuators B: Chemical*, 44(1-3):413–422.

[27] Gutierrez-Galvez, A. (2006). *Coding and learning of chemosensor array patterns in a neurodynamic model of the olfactory system*. PhD thesis, Texas A&M University. Texas A&M University.

[28] Gutierrez-Galvez, A. and Gutierrez-Osuna, R. (2006). Increasing the separability of chemosensor array patterns with Hebbian/anti-Hebbian learning. *Sensors and Actuators B: Chemical*, 116(1-2):29–35.

[29] Gutierrez-Osuna, R. (1998). *Signal Processing and Pattern Recognition for an Electronic Nose*. PhD thesis, North Carolina State University.

[30] Gutierrez-Osuna, R. (2000). Drift Reduction For Metal-Oxide Sensor Arrays Using Canonical Correlation Regression And Partial Least Squares. *Analysis*, pages 1–7.

[31] Gutierrez-Osuna, R. (2002a). A self-organizing model of chemotopic convergence for olfactory coding. In *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, number October, pages 236––237. IEEE.

[32] Gutierrez-Osuna, R. (2002b). Pattern analysis for machine olfaction: a review. *IEEE Sensors Journal*, 2(3):189–202.

[33] Gutierrez-Osuna, R. (2003). Transient response analysis for temperature-modulated chemoresistors. *Sensors and Actuators B: Chemical*, 93(1-3):57–66.

[34] Gutierrez-Osuna, R. and Nagle, H. T. (1999). A method for evaluating data-preprocessing techniques for odour classification with an array of gas sensors. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 29(5):626–32.

[35] Haugen, J., Tomic, O., and Kvaal, K. (2000). A calibration method for handling the temporal drift of solid state gas-sensors. *Analytica chimica acta*, 407(1-2):23–39.

[36] Heilig, A. and Ba, N. (1997). Gas identification by modulating temperatures of SnO 2 -based thick film sensors. *Sensors And Actuators*, 43:45 – 51.

[37] Hildebrand, J. G. and Shepherd, G. M. (1997). Mechanisms of olfactory discrimination:converging evidence for common principles across phyla. *Annual Review of Neuroscience*, 20:595–631.

[38] Holmberg, M. (1996). Drift counteraction for an electronic nose. *Sensors and Actuators B: Chemical*, 36(1-3):528–535.

[39] Holmberg, M., Davide, F. A. M., Natale, C. D., Amico, A. D., Winquist, F., and Lundstrm, I. (1997). Drift counteraction in odour recognition applications: lifelong calibration method. *Sensors and Actuators B: Chemical*, 42(3):185–194.

[40] Huerta, R. and Nowotny, T. (2012). Bio-inspired solutions to the challenges of chemical sensing. *Frontiers in neuroengineering*, 5(October):24.

[41] Imam, N., Cleland, T. a., Manohar, R., Merolla, P. a., Arthur, J. V., Akopyan, F., and Modha, D. S. (2012). Implementation of olfactory bulb glomerular-layer computations in a digital neurosynaptic core. *Frontiers in neuroscience*, 6(June):83.

[42] Janata, J. and Josowicz, M. (2003). Conducting polymers in electronic chemical sensors. *Nature Materials*, 2:19 – 24.

[43] Jurs, P. C., Bakken, G. A., and McClelland, H. E. (2000). Computational methods for the analysis of chemical sensor array data from volatile analytes. *Chemical reviews*, 100(7):2649–78.

[44] Kanaan-Izquierdo, S., Ziyatdinov, A., Massanet, R., and Perera, A. (2012). Multiview approach to spectral clustering. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1254–1257. IEEE.

[45] Kanaan-Izquierdo, S., Ziyatdinov, A., Massanet, R., and Perera, A. (2013). Patent Proposal: A computer spectral clustering improved method and uses thereof.

[46] Karvanen, J. and Koivunen, V. (2002). Blind separation methods based on Pearson system and its extensions. *Signal Processing*, 82:663 – 673.

[47] Kay, L. M., Beshel, J., Brea, J., Martin, C., Rojas-Líbano, D., and Kopell, N. (2009). Olfactory oscillations: the what, how and what for. *Trends in neurosciences*, 32(4):207–14.

[48] Kermit, M. and Tomic, O. (2003). Sensor Array Measurement Data. *Sensors (Peterborough, NH)*, 3(2):218–228.

[49] Khan, A. G., Parthasarathy, K., and Bhalla, U. S. (2010). Odor representations in the mammalian olfactory bulb. *Wiley interdisciplinary reviews. Systems biology and medicine*, 2(5):603–11.

[50] Koickal, T., Hamilton, A., Pearce, T., Tan, S., Covington, J., and Gardner, J. (2006). Analog VLSI design of an adaptive neuromorphic chip for olfactory systems. In *2006 IEEE International Symposium on Circuits and Systems*, volume 1, pages 4547–4550. IEEE.

[51] Lei, H., Pitt, W. G., Mcgrath, L. K., and Ho, C. K. (2007). Modeling carbon black / polymer composite sensors. *Sensors And Actuators*, 125:396–407.

[52] Liu, Q., Li, X., Ye, M., Ge, S., and Du, X. (2014). Drift compensation for electronic nose by semi-supervised domain adaption.

[53] López, L., Vouloutsi, V., Chimeno Escudero, A., Marcos, E., Bermúdez i Badia, S., Mathews, Z., Verschure, P. F., Ziyatdinov, A., and Perera i Lluna, A. (2011). Moth-Like Chemo-Source Localization and Classification on an Indoor Autonomous Robot. In Pramatarova, L. D., editor, *On Biomimetics*. InTech.

[54] Ma, M. and Shepherd, G. M. (2000). Functional mosaic organization of mouse olfactory receptor neurons. *Proceedings of the National Academy of Sciences*, 97(23):12869–12874.

[55] Marco, S. and Gutierrez-Galvez, A. (2012). Signal and Data Processing for Machine Olfaction and Chemical Sensing : A Review. *IEEE Sensors Journal*, 12(11):3189–3214.

[56] Marco, S., Gutierrez-Galvez, A., Lansner, A., Martinez, D., Rospars, J. P., Beccherelli, R., Perera, A., Pearce, T. C., Verschure, P. F. M. J., and Persaud, K. (2013). A biomimetic approach to machine olfaction, featuring a very large-scale chemical sensor array and embedded neuro-bio-inspired computation. *Microsystem Technologies*, pages 1–14.

[57] Marco, S., Ortega, A., Pardo, A., and Samitier, J. (1998). Gas identification with tin oxide sensor array and self-organizing maps: adaptive correction of sensor drifts. *IEEE Transactions on Instrumentation and Measurement*, 47:316.

[58] Marco, S., Pardo, A., and Ortega, A. (1997). Gas identification with tin oxide sensor array and. *Self*, pages 904–907.

[59] Martinetz, T., Schulten, K., et al. (1991). *A" neural-gas" network learns topologies*. University of Illinois at Urbana-Champaign.

[60] Martinez, G. M. and Basmadjian, D. (1996). Towards a general gas adsorption isotherm. *Chemical Engineering Science*, 51(7):1043 – 1054.

[61] Mountcastle, V. B. (1998). *Perceptual neuroscience: The cerebral cortex*. Harvard University Press.

[62] Neftci, E., Binas, J., Rutishauser, U., Chicca, E., Indiveri, G., and Douglas, R. J. (2013). Synthesizing cognition in neuromorphic electronic systems. *Proceedings of the National Academy of Sciences*, 110(37):E3468—-E3476.

[63] Nitta, T., Shigetomi, T., Kuro-Oka, M., and Katayama, T. (1984). An adsorption isotherm of multi-site occupancy model for homogeneous surface. *Journal of chemical engineering of Japan*, 17(1):39–45.

[64] Pearce, T., Schiffman, S., Nagle, H., and Gardner, J. (2003). *Handbook of Machine Olfaction - Electronic Nose Technology*. John Wiley & Sons.

[65] Pearce, T., Verschure, P., White, J., and Kauer, J. (2001). *Robust Stimulus Encoding in Olfactory Processing : Hyperacuity and Efficient Signal Transmission*, pages 461–479. Springer.

[66] Pearce, T. C., Karout, S., Rácz, Z., Capurro, A., Gardner, J. W., and Cole, M. (2013). Rapid processing of chemosensor transients in a neuromorphic implementation of the insect macroglomerular complex. *Frontiers in neuroscience*, 7(July):119.

[67] Perera, A., Yamanaka, T., Gutierrez Galvez, A., Raman, B., and Gutierrez-Osuna, R. (2006). A dimensionality-reduction technique inspired by receptor convergence in the olfactory system. *Sensors and Actuators B: Chemical*, 116(1-2):17–22.

[68] Perera, A. and Ziyatdinov, A. (2011). A Large Scale Virtual Array of Non-selective Odour Sensors for the Neurochem Project. In *Neurochem's Workshop on Bioinspired Computation for Chemical Sensing*.

[69] Persaud, K. and Dodd, G. (1982). Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose. *Nature*, 299(5881):352–355.

[70] Phaisangittisagul, E. (2008). Transient feature extraction for machine olfaction based on wavelet decomposition. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on*, volume 1, pages 457–460. IEEE.

[71] Principe, J. C., Tavares, V. G., Harris, J. G., and Freeman, W. J. (2001). Design and implementation of a biologically realistic olfactory cortex in analog vlsi. volume 89, pages 1030–1051. IEEE.

[72] Raman, B. (2005). *Sensor-based machine olfaction with neuromorphic models of the olfactory system*. PhD thesis, Texas A&M University.

[73] Raman, B. and Gutierrez-Osuna, R. (2006). Concentration normalization with a model of gain control in the olfactory bulb. *Sensors and Actuators B: Chemical*, 116(1-2):36–42.

[74] Raman, B., Stopfer, M., and Semancik, S. (2011). Mimicking biological design and computing principles in artificial olfaction. *ACS chemical neuroscience*, 2(9):487–499.

[75] Raman, B., Sun, P., Gutierrez-Galvez, A., and Gutierrez-Osuna, R. (2006). Processing of chemical sensor arrays with a biologically inspired model of olfactory coding. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 17(4):1015–24.

[76] Ressler, K. J., Sullivan, S. L., and Buck, L. B. (1994). Information coding in the olfactory system: evidence for a stereotyped and highly organized epitope map in the olfactory bulb. *Cell*, 79(7):1245–1255.

[77] Rhodes, P. a. and Anderson, T. O. (2012). Evolving a neural olfactorimotor system in virtual and real olfactory environments. *Frontiers in neuroengineering*, 5(October):22.

[78] Röck, F., Barsan, N., and Weimar, U. (2008). Electronic nose: current status and future trends. *Chemical reviews*, 108(2):705–25.

[79] Rodriguez-Lujan, I., Fonollosa, J., Vergara, A., Homer, M., and Huerta, R. (2014). On the calibration of sensor arrays for pattern recognition using the minimal number of experiments. *Chemometrics and Intelligent Laboratory Systems*, 130:123–134.

[80] Roth, M., Hartinger, R., Faul, R., and E, H. (1996). Drift reduction of organic coated gas-sensors by temperature modulation. *Sensors (Peterborough, NH)*, 36:358–362.

[81] Roussel, S. (1998). Optimisation of Electronic Nose Measurements. Part I: Methodology of Output Feature Selection. *Journal of Food Engineering*, 37(2):207–222.

[82] S.-Y. Choi, Takahashi, K., Esashi, M., and Matsuo, T. (1985). Stability and sensitivity of MISFET hydrogen sensors. *Int. Conf. Solid-State Sensors and Actuators*.

[83] Samitier, J., López-Villegas, J., Marco, S., Cámara, L., Pardo, A., Ruiz, O., and Morante, J. (2002). A new method to analyse signal transients in chemical sensors. *Sensors and Actuators B: Chemical*.

[84] Schmuker, M., Pfeil, T., and Nawrot, M. P. (2014). A neuromorphic network for generic multivariate data classification. *Proceedings of the National Academy of Sciences of the United States of America*, 111(6):2081–6.

[85] Shepherd, G. M. (1987). A molecular vocabulary for olfaction. *Annals of the New York Academy of Sciences*, 510(1):98–103.

[86] Shepherd, G. M. (1994). Discrimination of molecular signals by the olfactory receptor. *Neuron*, 13:771–790.

[87] Shepherd, G. M. et al. (2004). *The synaptic organization of the brain*, volume 3. Oxford University Press New York.

[88] Sircar, S. (1995). Influence of adsorbate size and adsorbent heterogeneity of iast. *AIChE Journal*, 41(5):1135–1145.

[89] Tang, L., Peng, S., Bi, Y., Shan, P., and Hu, X. (2014). A new method combining lda and pls for dimension reduction. *PloS one*, 9(5):e96944.

[90] Tomic, O., Ulmer, H., and Haugen, J.-e. (2002). Standardization methods for handling instrument related signal shift in gas-sensor array measurement data. *Analytica Chimica Acta*, 472:99–111.

[91] Topart, P. and Josowicz, M. (1992). Transient Effects In the Interaction between Polypyrrole and Methanol Vapor. *The Journal of Physical Chemistry*, 96(21):8662–8666.

[92] Trevor Hastie, Robert Tibshirani, J. F. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, second edi edition.

[93] Vassar, R., Chao, S. K., Sitcheran, R., Vosshall, L. B., Axel, R., et al. (1994). Topographic organization of sensory projections to the olfactory bulb. *Cell*, 79(6):981–991.

[94] Vergara, A., Fonollosa, J., Mahiques, J., Trincavelli, M., Rulkov, N., and Huerta, R. (2013). On the performance of gas sensor arrays in open sampling systems using Inhibitory Support Vector Machines. *Sensors and Actuators B: Chemical*, 185:462—-477.

[95] Vergara, A., Vembu, S., Ayhan, T., Ryan, M. a., Homer, M. L., and Huerta, R. (2012). Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, pages 1–10.

[96] Verhagen, J. V., Wesson, D. W., Netoff, T. I., White, J. a., and Wachowiak, M. (2007). Sniffing controls an adaptive filter of sensory input to the olfactory bulb. *Nature neuroscience*, 10(5):631–9.

[97] Vilanova, X. (1996). Analysis of the conductance transient in thick-film tin oxide gas sensors. *Sensors and Actuators B: Chemical*, 31(3):175–180.

[98] Vouloutsi, V., López, L., Mathews, Z., Chimeno Escudero, A., Ziyatdinov, A., Perera i Lluna, A., Bermúdez i Badia, S., and Verschure, P. F. (2013). The Synthetic Moth: A Novel Neuromorphic Approach towards Artificial Olfaction in Robots. In Persaud, K. C., Marco, S., and Gutierrez-Galvez, A., editors, *Neuromorphic Olfaction*. CRC Press.

[99] Wachowiak, M. (2010). Active Sensing in Olfaction. In Menini A., editor, *The Neurobiology of Olfaction*. CRC Press, boca raton edition.

[100] Wang, X. R., Lizier, J. T., Nowotny, T., Berna, A. Z., Prokopenko, M., and Trowell, S. C. (2014). Feature selection for chemical sensor arrays using mutual information. *PloS one*, 9(3):e89840.

[101] Wilson, D. and DeWeerth, S. (1995). Odor discrimination using steady-state and transient characteristics of tin-oxide sensors. *Sensors and Actuators B: Chemical*, 28(2):123–128.

[102] Wold, S. (1998). Orthogonal signal correction of near-infrared spectra. *Chemometrics and Intelligent Laboratory Systems*, pages 175–185.

[103] Yang, R. T. (2003). *Adsorbents: fundamentals and applications*. John Wiley and Sons, Inc., Hoboken, New Jersey.

[104] Ziyatdinov, A., Calvo, J. M. B., Lechón, M., Badia, S. B. i., Verschure, P. F. M. J., Marco, S., and Perera, A. (2011a). Odour Mapping Under Strong Backgrounds With a Metal Oxide Sensor Array. In *AIP Conf. Proc. 1362*, pages 232–233.

[105] Ziyatdinov, A., Chaudry, A., Persaud, K., Caminal, P., and Perera, A. (2009). Common principal component analysis for drift compensation of gas sensor array data. In *Olfaction and electronic nose: Proceedings of the 13th International Symposium on Olfaction and Electronic Nose*, volume 1137, pages 566–569. AIP Publishing.

[106] Ziyatdinov, A., Fernández, L., Gutierrez-Galvez, A., Fonollosa, J., Marco, S., and Perera, A. (2013a). A qualitative study of a bioinspired sensor system based on gas flow modulation by an artificial lung apparatus. In *15th International Symposium on Olfaction and electronic nose*.

[107] Ziyatdinov, A., Fernández Diaz, E., Chaudry, A., Marco, S., Persaud, K., and Perera, A. (2011b). A Large Scale Virtual Gas Sensor Array. In *AIP Conf. Proc. 1362*, pages 151–152.

[108] Ziyatdinov, A., Fernández Diaz, E., Chaudry, A., Marco, S., Persaud, K., and Perera, A. (2013b). A software tool for large-scale synthetic experiments based on polymeric sensor arrays. *Sensors and Actuators B: Chemical*, 177:596–604.

[109] Ziyatdinov, A., Marco, S., Chaudry, A., Persaud, K., Caminal, P., and Perera, A. (2010). Drift compensation of gas sensor array data by common principal component analysis. *Sensors and Actuators B: Chemical*, 146(2):460–465.

[110] Ziyatdinov, A. and Perera-Lluna, A. (2014). Data Simulation in Machine Olfaction with the R Package Chemosensors. *PLoS ONE*, 9(2):e88839.

[111] Zou Z, L. F. and LB, B. (2005). Odor maps in the olfactory cortex. *PNAS*, 102:7724–7729.