TECHNICAL REPORT
NATICK/TR-14/021

AD ________________

DATA-DRIVEN PROPERTY ESTIMATION FOR PROTECTIVE CLOTHING

by
Sree Srinivasan*
Joseph Lavoie
and
Ramanathan Nagarajan

***Oak Ridge Institute for Science and Education (ORISE)**
Belcamp, MD 21017

September 2014

Final Report
January 2012 – December 2013

Approved for public release; distribution is unlimited

U.S. Army Natick Soldier Research, Development and Engineering Center
Natick, Massachusetts 01760-5020

DISCLAIMERS

The findings contained in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of trade names in this report does not constitute an official endorsement or approval of the use of such items.

DESTRUCTION NOTICE

For Classified Documents:

Follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX.

For Unclassified/Limited Distribution Documents:

Destroy by any method that prevents disclosure of contents or reconstruction of the document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 30-09-2014	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* January 2012 – December 2013

4. TITLE AND SUBTITLE

DATA-DRIVEN PROPERTY ESTIMATION FOR PROTECTIVE CLOTHING

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Sree Srinivasan*, Joseph Lavoie, and Ramanathan Nagarajan

5d. PROJECT NUMBER
BA07PRO102

5e. TASK NUMBER
IV

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Natick Soldier Research, Development and Engineering Center
ATTN: RDNS-WSC-M
Kansas St., Natick, MA 01760-5020

8. PERFORMING ORGANIZATION REPORT NUMBER

NATICK/TR-14/021

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Threat Reduction Agency
8725 John J. Kingman Rd.
Stop 6201
Fort Belvoir, VA 22060-6201

10. SPONSOR/MONITOR'S ACRONYM(S)
DTRA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

*Oak Ridge Institute for Science and Education (ORISE); Oak Ridge Associated Universities (ORAU) Maryland, 4692 Millennium Drive , Suite 101, Belcamp, MD 21017

14. ABSTRACT

This report details a 2-year exploratory effort for making data-driven prediction of barrier properties, completed in December 2013 at the Natick Research, Development and Engineering Center (NSRDEC) as part of the Integrated Protective Fabric System (IPFS) Project sponsored by the Defense Threat Reduction Agency (DTRA). Desorption data for 23 organic solvents ("data chemicals") from butyl rubber were measured and comprehensively analyzed to estimate diffusion coefficients. Using commercial computational chemistry software, machine-readable structures were built and numerous molecular descriptors calculated for these solvents and several threat agents and simulants ("query chemicals"). Matlab® codes were developed and probed to implement a machine learning technique—Artificial Neural Networks. Trained using the descriptors and diffusion coefficients of the data chemicals, the network is able to make good predictions for query chemicals for which only the descriptors are known. Cheminformatics—demonstrated in this work for characterizing threat agents—has a broader scope, in computational toxicology.

15. SUBJECT TERMS

NTA PERMEATION CHEMINFORMATICS ARTIFICIAL INTELLIGENCE
MODELS PREDICTIONS NEURAL NETWORKS NON-TRADITIONAL AGENTS
BARRIERS DATA-DRIVEN PREDICTION MODELS COMPUTATIONAL CHEMISTRY
SOLVENTS NEURAL NETS THREAT EVALUATION CHEMICAL WARFARE AGENTS
NETWORKS BUTYL RUBBER PROTECTIVE CLOTHING CHEMICAL AGENT SIMULANTS
SIMULANTS CHEMICAL AGENTS DIFFUSION COEFFICIENT ARTIFICIAL NEURAL NETWORKS
ANN(ARTIFICIAL NEURAL NETWORKS)

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Ramanathan Nagarajan
U	U	U	UU	100	**19b. TELEPHONE NUMBER** *(include area code)* 508-233-6445

This page intentionally left blank

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This report details the methodology and findings of an auxiliary research effort aimed at "data-driven" prediction of fabric barrier properties conducted at the U.S. Army Natick Soldier Research, Development and Engineering Center (NSRDEC), as part of Task IV (Advanced Modeling for Fabric Systems) of the Integrated Protective Fabric System (IPFS) Project (BA07PRO102) being spearheaded by NSRDEC and sponsored by the Defense Threat Reduction Agency (DTRA). The work described was done from January 2011 to December 2013. It illustrates the promise of machine learning for property estimation and defines the scope for deploying "big data" tools upon small databases.

ACKNOWLEDGMENTS

The authors would like to thank:

- Integrated Protective Fabric System (IPFS) Project Manager Heidi Schreuder-Gibson for constant support of exploration
- Matlab wizards on the internet (Greg Heath, Jacek Kierzenka, Siyi et al.) for coding ruses
- Cheminformatics numerati worldwide (Subhash Basak, Achintya Das, Kas Subramanian, Vikrant Vijay, Junmei Wang, Matthias Scholz, Igor Tetko, Gennady Poda, Frank Burden, Antony Williams, Stan Young et al.) for pointers on QSAR theory and practice
- NSRDEC editors Robert L Gaumer and Julia E. Morris for excellent editorial advice

EXECUTIVE SUMMARY

Discerning patterns in “big data” by data mining is a hot topic: Google searches, credit scores, stock picking, election strategies, and mail sorting (recognition of hand-written zip codes) are but a few ubiquitous applications. The underlying use of computer algorithms to “learn” by exposure to raw data and make useful recognitions and reliable predictions falls under the rubric “machine learning”. Inspired by the applications of machine learning in pharmaceutical drug design and computational toxicology, the U.S. Army Natick Soldier Research, Development and Engineering Center (NSRDEC) conducted an exploratory research effort, to make data-driven prediction of barrier properties of chemical protective gear. This effort, completed in December 2013, was a part of Task IV (Advanced Modeling for Fabric Systems) of the Integrated Protective Fabric System (IPFS) Project—a multi-organization, multi-university, multi-year program spearheaded by NSRDEC, under the aegis of the Defense Threat Reduction Agency (DTRA).

Being an incubator for envisioned military technologies and a testing laboratory in the supply chain of military procurements, NSRDEC is often called upon to characterize and assess developmental and commercial concepts and materials. This task invariably involves the estimation of some property or another. In the case of chemical protective gear, a metric of interest is how effective the material is at stopping or slowing down the intake of chemical warfare agents (CWAs) or non-traditional threat agents (NTAs). Correspondingly, the property to estimate is the diffusivity or permeability of CWAs and NTAs in the barrier materials.

Property estimation is not a trivial task. Handling threat agents is harmful and hence is usually avoided by the use of simulants. Regardless of whether a chemical is an agent or a simulant, resorting to experimental measurements in each and every case would be impractical and costly. *Ab initio* prediction from first principles is not the answer either, since a suitable theory may not exist for every system.

An alternative to measurement or theory is data-driven property estimation: make use of old or new property databases (on common solvents, past agents, and simulants interacting with the barrier material under investigation or similar barriers) and, by generalizing from the known to the unknown, predict the performance of the query chemicals in that barrier material. Such methods are formally known as quantitative structure activity relationships (QSAR) or quantitative structure property relationships (QSPR): given the query chemical’s formula, structure, or a few key measured or calculated features, QSAR/QSPR would yield a property estimate.

Traditional QSAR employs linear regression of available data using correlations based on intuition. For example, the Potts-Guy correlation yields a chemical’s skin permeability, given its molecular weight and octanol/water partition coefficient. Lately, correlations have been supplemented or supplanted by techniques of pattern recognition via unstructured or structured learning such as Classification and Regression Trees (CART), Artificial Neural Networks (ANN), and Fuzzy Logic. Specific “big data” applications in data-driven property estimation abound in the field of cheminformatics—a combination of data mining and computational chemistry.

Cheminformatics is routine in pharmaceutical drug discovery by virtual screening to find synthesizable chemicals that may mimic the pharmacologic activity of a synthetic or natural lead drug or known active. Cheminformatics is also being increasingly used in computational toxicology—under a deluge of new industrial chemicals—to assess chemicals of unknown toxicity, by searching for analogs that have known toxicity profiles and making predictions based on the available data on the analogs to arrive at toxicity estimates for the query chemicals. It became of interest to probe whether the power of such machine learning techniques can be harnessed to the task of predicting barrier properties for military applications. A small exploratory effort was launched, tasked with applying such data-driven methods to address CWA/NTA behavior in conditioned polymer films. The sub-tasks were to develop:

- Experimental methods to measure chemical permeation through barrier materials.
- Computational-chemistry capabilities to calculate a variety of properties or “molecular descriptors” given only the formula of a chemical.
- Data-driven algorithms to predict the permeation of chemicals through barrier materials.

The products of these three sub-tasks were categorized as “data”, “descriptors”, and “predictions”, respectively.

The “data” comprise a database of target properties, mainly diffusion coefficients for 23 selected solvents in a commercial butyl rubber sheet, estimated from desorption transients measured using the immersion technique. The analysis underlying this parameter estimation was quite involved and made use of mass transfer models of varying complexity, including a novel nonlinear model that accounts for the de-swelling of the rubber sheet as desorption proceeds.

The target property was the dependent variable (commonly denoted as Y) for establishing or “training” the prediction technique. The solvents constitute the “data chemicals” for which the target property (here, the diffusion coefficient) is known from the above measurements. The other moiety of chemicals, for which the target property is unknown and has to be predicted, constitute the “query chemicals”.

Training also required independent variables (denoted as X), namely molecular properties or “descriptors” of the data chemicals. Descriptors—which can be thought of as the “personal identification numbers” of the chemicals—also needed to be calculated for the query chemicals. With only the molecular formulas as the input, the descriptors were calculated using commercial computational chemistry software. First, the molecules were rendered as structures in machine-readable formats using one type of software (Spartan®). Next, the structures were input to another type (Sarchitect®) that calculates all the descriptors.

Not all of the descriptors are pertinent to every target property. That is, not all of the 1,000 or so molecular descriptors were used while training the network or in using the trained network to make predictions. In order to avoid “the curse of dimensionality”, key descriptors had to be selected for each type of target, either manually (based on expert opinion) or automatically, via “filter” or “wrapper” algorithms. Filters use set selection criteria; e.g., that the chosen descriptors be correlated highly with the targets, but weakly with one another. Wrappers try various sets (or combinations) of descriptors and arrive at the optimal descriptor set iteratively based on feedback from the prediction errors that result from each set. Both filters and wrappers were pursued in this work initially, but eventually the culling of key descriptors was left to expert choice. With some straightforward coding—using genetic algorithms, for instance—descriptor selection can be automated as well.

A well-known structured learning technique—Artificial Neural Networks, or ANN—was implemented for data-driven prediction, yielding the "predictions". The network was trained using a select few of the molecular descriptors and the target properties of the data chemicals (in terms of a hidden function: Y = f(X)); the trained network was then used to make the "predictions" given the "descriptors" of the query chemicals.

The structure of the project is shown in Figure ES-1, with the outputs highlighted in yellow.

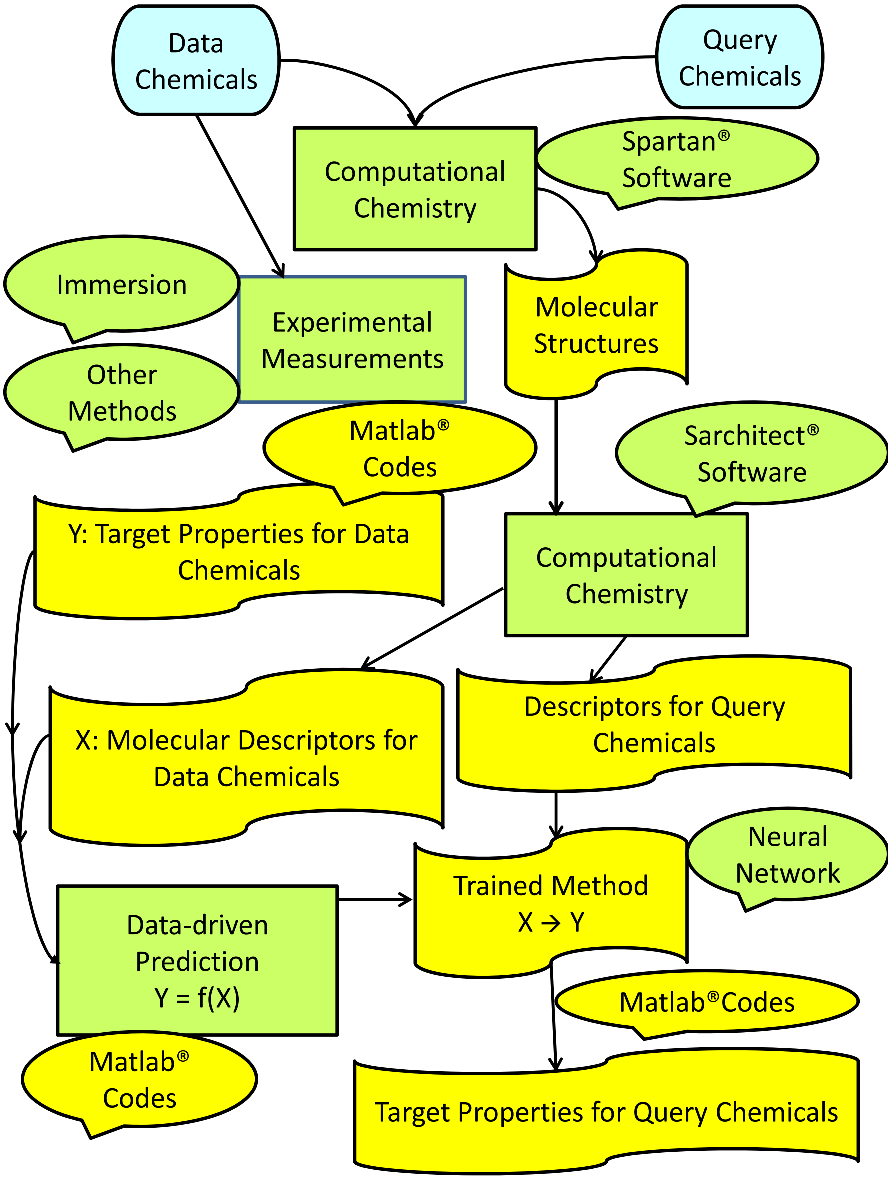


Figure ES-1 Data-Driven Property Estimation--Overview

The goals of the effort were met, with varying degrees of success. The results, in outline:

- Desorption transients of 23 organic solvents from a commercial butyl rubber sheet were measured using the immersion method. They were then comprehensively analyzed to obtain integral diffusion coefficients based on the linear constant-thickness, "Crank's Solution", as well as on a novel nonlinear model that accounts for swelling, and to obtain the parameters for solubilities using Flory-Huggins, Hildebrand, and Hansen theories.
- Using commercial computational chemistry software (Spartan®), machine-readable molecular structures in Structure-Data File (SDF) format were built for these organic solvents ("data chemicals"), as well as about 60 threat agents and simulants ("query chemicals").
- Using commercial computational chemistry software (Sarchitect®) and based on the molecular structures, about 1,000 constitutional, topological, and conformational properties ("molecular descriptors") were calculated for all the data and query chemicals.
- A machine learning technique—ANN, for regression—was implemented, by means of NSRDEC-developed Matlab® software and detailed probing studies, to provide data-driven prediction of target properties that characterize how chemical threat agents would permeate protective barrier materials. The predictions, while not perfect, are good enough to be on par with literature precedents for predicting aqueous solubility of pharmaceutical chemicals.
- Besides numerical data, descriptors, and predictions, this work generated:
    - Protocols (for using commercial computational-chemistry software to render the structures and calculate the descriptors)
    - Tutorials (on ANN applied to property estimation)
    - Matlab® codes (one set of codes for extracting diffusivity estimates from desorption transients and another set that, after some modifications, can be used for predicting a host of other properties of interest besides diffusion coefficients). The codes have many options and features and a coding-style aimed at error avoidance, extensive commentary, and displays and provisions for comprehensive record-keeping.

This work has brought out the power and promise of machine learning for property estimation and delineated the scope for deploying "big data" techniques on small databases:

- Bayesian regulation ANN (BRANN) is ideal for small datasets since it avoids, unlike other neural network algorithms, the need for setting aside a portion of the training data for internal validation, thus reducing the data demand.
- In QSAR or QSPR, prediction accuracy depends crucially on data clarity and coverage. That is, what the network is being fed as training target data must genuinely represent the phenomenon being investigated and not be obscured by spurious effects, imprecise measurements, or fitting-model inaccuracy, especially if the "data" are not directly measured quantities such as aqueous solubility, but are actually theory-based estimates of some parameter, e.g., diffusion coefficient, as in the present case.
- Because structure-activity relations invariably involve nonlinearities that cannot always be linearized by merely taking logarithms, it is better to use the raw descriptors directly as independent variables, instead of linear combinations of the descriptors such as statistical principal components.

- The data chemicals should cover a broad range of the descriptors that are significant to the underlying phenomenon; for example, if molecular weight is a key descriptor, the weights of the molecules chosen must include at least one high value and one comparatively low value as in a two-level factorial design of experiments.
- If the range of key descriptors in the training set is broad enough for the network to "learn" the dependence of the target property on the descriptors, similarity between data and query chemicals is less of a requirement.
- The minimum number of data chemicals should equal or exceed the number of effective parameters (in the target-descriptor relationship), which, incidentally, may not be more than a few in macroscopic phenomena like permeation. In fact, *a posteriori* estimates (that are an output of the algorithm used for training the network) put the number of effective parameters around five or six in the present cases.

The last point brings out an important difference between standard data mining and QSAR. Standard "big data" involves pattern recognition by sifting through very large but subjective databases—faces, hand-writing, key strokes, stock prices, voting, and such. Cheminformatics aimed at pharmaceutical drug discovery involves needle-in-the-haystack pattern recognition amidst elusive and complex biological phenomena. In contrast, data-driven QSAR involves a clinical analysis of carefully curated data on relatively well-defined physicochemical systems.

In sum, the training database for data-driven QSAR for physicochemical properties need not be big, in principle. In practice, however, more data chemicals would be needed if the data are noisy or display less variance in a key descriptor; the phenomenon of "regression to the mean" should result in successful training and prediction with increasingly large databases. That is, while a large database size does not guarantee prediction accuracy, with a large database the network may be able to avoid overfitting to the noise and detect the true patterns, i.e., capture the underlying generalities in the data without memorizing data idiosyncrasies. However, since cost considerations drive down the number of data chemicals, data quality remains paramount.

Data-driven methods can be used to predict not just permeation, but also many other physicochemical properties: solubilities, vapor pressures, partition coefficients, chemical degradation products, and, with additional effort, toxicity metrics. That is, the prediction codes developed in this work are not restricted to the diffusivity database, but can be used to make predictions of other physicochemical or toxicity properties, with some modifications, given the appropriate databases. Subject-matter expertise would be helpful in descriptor selection, however, i.e., in deciding which molecular features are important for the target property to be predicted. With some additional coding descriptor selection can be automated as well.

Despite the challenges of the original remit of the task, the effort has demonstrated and strengthened NSRDEC expertise in cheminformatics, a discipline that has much potential for addressing important DTRA objectives such as computational toxicology of NTAs.

DATA-DRIVEN PROPERTY ESTIMATION FOR PROTECTIVE CLOTHING

CHAPTER 1 INTRODUCTION

This report details an exploratory effort, completed in December 2013, for making data-driven prediction of barrier properties. This project was a part of Task IV (Advanced Modeling for Fabric Systems) of the Integrated Protective Fabric System (IPFS) Project–a multi-organization, multi-university, multi-year program being conducted by the U.S. Army Natick Soldier Research, Development and Engineering Center (NSRDEC) under the sponsorship of the Defense Threat Reduction Agency (DTRA).

As an incubator for envisioned military technologies and as a testing laboratory aiding military procurements, NSRDEC is often called upon to characterize and assess developmental and vendor-proffered materials. This task invariably involves property estimation. In the case of chemical protective gear, the property to estimate is the diffusivity or permeability of chemical warfare agents (CWAs) and non-traditional threat agents (NTAs) in the barrier materials.

Property estimation is not a trivial task. Handling threat agents is harmful and hence is usually avoided by the use of simulants. With agents or simulants, resorting to experimental measurements in each and every case would be impractical and costly. Prediction from first principles is not a panacea either, since a suitable theory may not exist for every system.

An alternative to measurement or theory is data-driven property estimation: make use of old or new property databases (on common solvents, past agents, and simulants interacting with the barrier material under investigation) and predict the performance of known and new query chemicals in that barrier material. Such methods are formally known as quantitative structure activity relationships (QSAR) or quantitative structure property relationships (QSPR). Given the query chemical's formula or structure, QSAR/QSPR would yield a property estimate.

Traditional QSAR employs linear regression of available data using correlations based on intuition. Lately, however, correlations have been supplemented or supplanted by computer science techniques of qualitative and quantitative pattern recognition via unstructured or structured learning such as Classification and Regression Trees (CART), Artificial Neural Networks (ANN), and Fuzzy Logic. Ubiquitous applications of such "big data" techniques include Google searches, credit scores, stock picking, election strategies, and recognition of (hand-written) zip codes. Specific applications in data-driven property estimation abound in the field of cheminformatics—a combination of data mining and computational chemistry.

Cheminformatics is routine in pharmaceutical drug discovery by virtual screening to find synthesizable chemicals that may mimic the pharmacologic activity of a synthetic or natural lead drug or known active. Cheminformatics is also being increasingly used in computational toxicology—under a deluge of new industrial chemicals—to assess chemicals of unknown toxicity by searching for analogs with known toxicity profiles and by making predictions based on the available data on the analogs to arrive at toxicity estimates for the query chemicals. It became of interest to probe whether the power of such machine-learning techniques can be harnessed to the task of predicting barrier properties for military applications. The main goal of this effort was to apply such data-driven methods to address CWA/NTA behavior in conditioned polymer films. The sub-tasks were to develop:

- Experimental methods to measure chemical permeation through barrier materials.
- Computational chemistry capabilities to calculate a variety of properties or “molecular descriptors” given only the formula of a chemical.
- Data-driven algorithms to predict the permeation of chemicals through barrier materials.

Correspondingly, this report is divided into three chapters in addition to the Introduction and Conclusions: Chapter 2, Data; Chapter 3, Descriptors; and Chapter 4, Predictions.

Chapter 2 details the development of a database of target properties—diffusion coefficients, polymer solubility parameters, and solvent volume fractions—for 23 selected solvents in a commercial butyl rubber sheet, from desorption transients measured using the immersion technique. (Future reports shall detail alternative measurement methods such as the drop volume technique.) Such target properties constitute the dependent variables (commonly denoted as Y) for establishing or “training” the prediction technique.

Chapter 3 details the development of the database of molecular descriptors for data and query chemicals. The molecular descriptors, namely molecular properties of the solvents which are the “data chemicals”, constitute the independent variables (denoted as X) required for training the prediction technique. Descriptors also had to be calculated for the “query chemicals” for which the target properties were predicted using the trained method. With only the molecular formulas as the input, the descriptors were calculated using commercial computational chemistry software.

Chapter 4 details how a well-known structured learning technique—ANN—was implemented, by means of in-house developed software and detailed probing studies, to provide data-driven prediction of target properties that characterize how CWAs would permeate in protective barrier materials. The network is trained using data chemicals for which molecular descriptors as well as target properties are known (in terms of a hidden function: Y = f(X)); the trained network is then used to make predictions for query chemicals for which only the descriptors are known.

The structure of the project is shown in Figure 1, with the outputs or deliverables highlighted in yellow. Besides numerical results (namely, diffusivity data, and predictions for a small set of data and query chemicals), the project’s outputs also included useful databases (of molecular structures in machine-readable formats and about 1,000 molecular descriptors for each chemical), protocols (for using commercial computational-chemistry software to render the structures and calculate the descriptors), a tutorial (on ANN), and Matlab® codes for estimating solubility and diffusivity parameters from desorption transients and, for predicting diffusion coefficients (and, after some modifications, a host of other properties of interest).

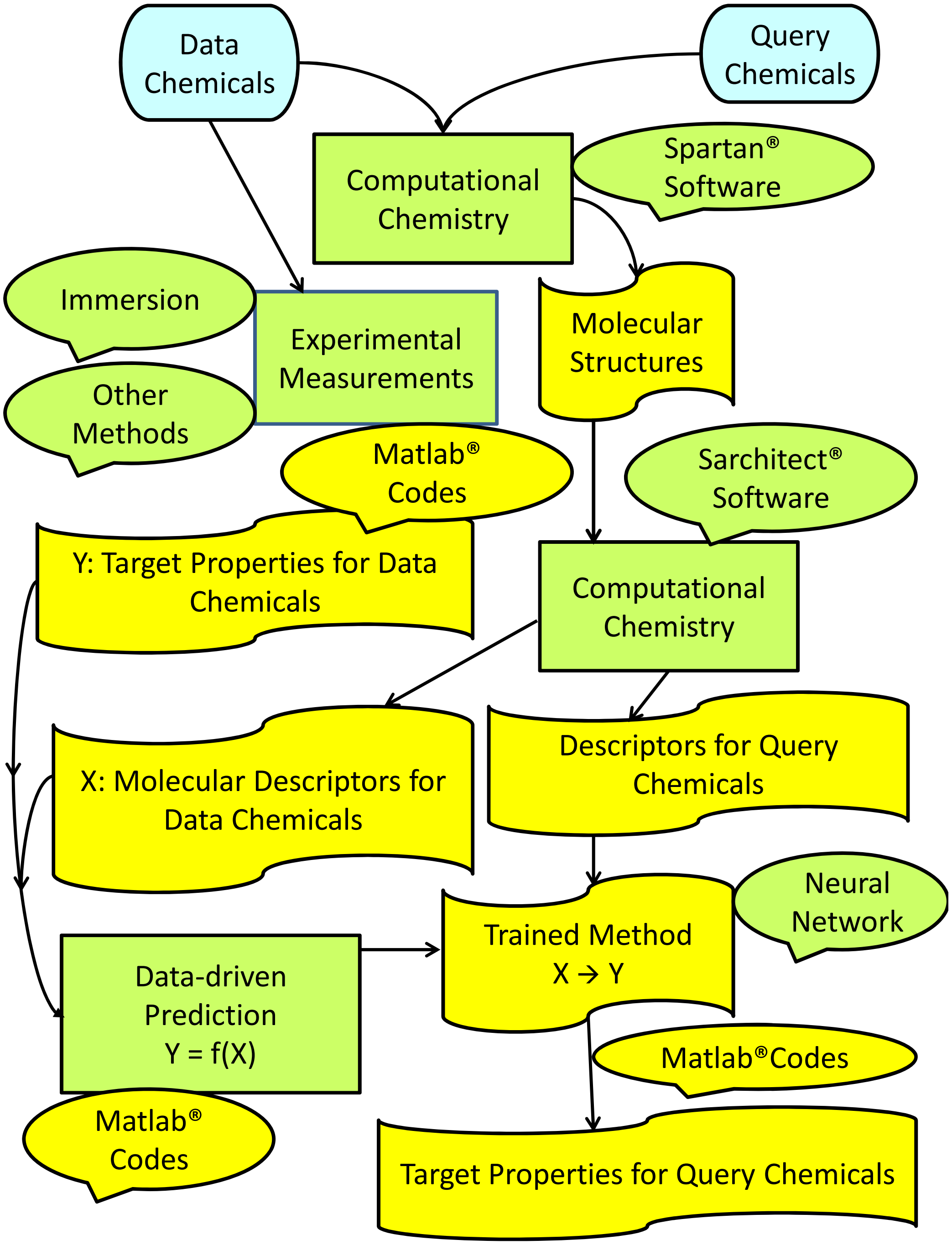


Figure 1 Data-Driven Property Estimation—Overview

CHAPTER 2 DATA

2.1 BACKGROUND

In order to evaluate the level of chemical protection provided by a material, it is necessary to predict the rate of permeation through it. Because of the hazards of working with CWAs even in a controlled laboratory environment, studies are usually conducted not directly on CWAs, but on less toxic chemicals similar to the CWAs in structure and physicochemical properties. With a large group of simulants or ordinary chemicals covering a wide range of functionalities and properties, a database can be established that will enable prediction of permeability of CWAs, using advanced statistics and computational chemistry. Towards this goal, solubility and diffusion coefficients were estimated from the rates at which solvents desorb from butyl rubber. The underlying analyses and coding were invested with a great deal of intensity and attention– since the resulting estimates constitute the “data” in data-driven prediction with the predictions being acutely sensitive to data quality, especially when the database is small. Amidst the vast literature on diffusion in polymers (reviewed in Refs. 1-3), the literature search was focused on a few key topics: butyl rubber [4-8], parameter estimation [9-11], and swelling [12]. This chapter details development of a database of diffusion coefficients for 23 solvents in a commercial butyl rubber sheet, from desorption transients measured using the immersion technique.

2.2 EXPERIMENTAL METHOD

2.2.1 Materials

The number of solvents had to be small to cut costs, but an attempt was made to choose solvents that are diverse in structure. Butyl rubber was obtained from Midwest Rubber Sales as a large 1/32-in thick sheet, which was cut into disks for testing approximately 5 cm in diameter and 0.7 mm thick. (If necessary, the same measurements can be done on commercial glove materials.) In this report, as in the literature, the terms “disk”, “membrane”, and “film” are used interchangeably. The measurements were performed in screw-top jars with a layer of glass beads at the bottom to prevent the disk of butyl rubber from lying flat on the bottom of the jar, as shown in Figure 2, in order to maintain a uniform environment on both sides of the disk.

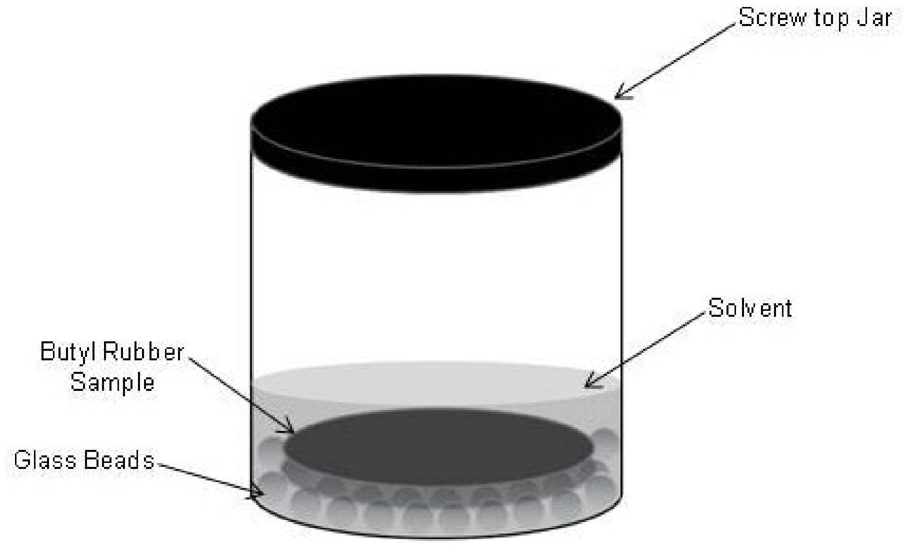


Figure 2 Immersion Test Schematic

2.2.2 Measurements

The jars were cleaned with isopropyl alcohol; the disks were rinsed in water, air-dried, weighed, and submerged in the solvent under study. In order to avoid the errors associated with additives leaching out, measurements were made during desorption instead of absorption:

1. First, a disk was submerged in the solvent and taken out daily to be weighed. This helped establish the duration required to reach saturation (2 to 20 days), indicated by the sample mass not increasing by more than 1% in a 24-h period.
2. Second was the preparatory stage where multiple butyl rubber samples were submerged in the solvent inside individual jars for the duration determined in the first step.
3. In the third and final stage of data collection, the samples were removed from the solvent and placed upright in short vials, exposing both sides of the sample to air. In this desorption stage, which typically lasted several days, and in some cases weeks, periodic measurements of the sample mass (or, loosely, "weight") were made.

A few other weights are also significant: W_F (the weight of the pristine disk, before any sorption or desorption), W_0 (after the soaking step), and W_∞ (at the end of desorption). Ideally, the ultimate weight at the end of desorption should be the same as the weight of the pristine disk. In reality W_∞ is invariably smaller than W_F because of additives leaching out during the soaking step. Further, W_∞ is also subject to uncertainty when diffusion is very slow or the measurements are stopped prematurely; hence, W_∞ may be an adjustable parameter besides D.

2.3 DATA ANALYSIS AND THEORY

2.3.1 Weights

The amount of additives is set as the difference between W_F and the measured or fitted W_∞.

$$W_A = W_F - W_\infty \tag{1a}$$

The initial solvent loading is obtained from the weights after soaking and after complete desorption.

$$W_S = W_0 - W_\infty \tag{1b}$$

Correspondingly, the weight of the polymer moiety is given by

$$W_P = W_F - W_A = W_\infty \tag{1c}$$

The solvent-to-polymer weight ratio or "Swelling Ratio" [5] is calculated as

$$w_{SP} = \frac{W_s}{W_P} = \frac{W_0 - W_\infty}{W_\infty} \tag{1d}$$

Another ratio of interest: the solvent loading to the amount of additives.

$$w_{SA} = \frac{W_s}{W_A} = \frac{W_0 - W_\infty}{W_F - W_\infty} \tag{1e}$$

2.3.2 Sorption

Based on the solvent loading ratio w_{SP} , the polymer volume fraction is calculated as

$$\phi_p = \frac{1/\rho_p}{1/\rho_p + w_{SP}/\rho_s} \tag{2}$$

The relevant solubility theory draws from two theoretical canons: polymer-solvent interactions originally developed by Flory and Huggins, and solubility parameters originally proposed by Hildebrand and developed further by Hansen. Defining the energy parameter in terms of solubility parameters was an innovation due to Zellers, Hardy, et al. [6-8]. The sorption theory is amply detailed in Ref. 8. Here, summarizing the final equation should suffice.

$$-[\ln(1-\phi_p)+\phi_p] = \chi\phi_p{}^2 \tag{3}$$

The left-hand side is calculated entirely based on the measured polymer fraction, ϕ_p . The right-hand side involves many fixed and adjustable parameters. The coefficient χ is calculated using the properties of the solvent (which can be found in handbooks) and the properties of the polymer (treated as adjustable parameters).

$$\chi = \chi_H + \chi_S \tag{4}$$

The Flory-Huggins interaction parameter χ comprises an energy part χ_H that is a function of several adjustable parameters and an entropy part χ_S, which is by itself an adjustable parameter.

$$\chi_H = \frac{V_S A}{RT} \tag{5}$$

V_S is the molar volume of the solvent, and A is the square of the "distance" between solvent and polymer in the space of solubility parameters.

$$A = a\,(\delta_{dS} - \delta_{dP})^2 + b\left\{(\delta_{pS} - \delta_{pP})^2 + (\delta_{hS} - \delta_{hP})^2\right\} \tag{6}$$

The deltas are the solubility parameters. The lower-case subscripts d, p, and h denote dispersion, polar, and hydrogen-bonding, respectively. The subscripts S and P denote solvent and polymer. It is important to note that, in accordance with Ref. 8, "a" is set to 1 and "b" is fixed at 0.25.

In summary, for each solvent, four input parameters (the three deltas and the molar volume) are looked up in a handbook [9]. These, along with the measured polymer fraction, are input to an interactive nonlinear least squares regression program (using the Levenberg-Marquardt algorithm) in Matlab®. The program finds the best-fit estimates for the four adjustable parameters: the three deltas for the polymer and the entropy parameter.

A pertinent discussion of units (of solubility, diffusivity, and permeability) is given in Appendix A.

2.3.3 Diffusion

For analyzing the desorption transients to extract diffusion coefficients, the weights are normalized to yield the fractional uptakes as:

$$\frac{M_t}{M_0} = \frac{W_t - W_\infty}{W_0 - W_\infty} \tag{7}$$

In desorption, the fractional uptake starts at unity and declines to zero with time. Desorption transient data were first probed for diffusion anomalies, next fitted to a classic linear theory, and finally fitted to a numerical solution of a nonlinear model to account for (de-)swelling.

2.3.4 Diffusion Anomalies

In a standard classification [10 (Section 11.1)], diffusion in polymers falls into three types: Case I (Fickian), in which diffusion is slower than the relaxation of the membrane structure; Case II, where diffusion is fast and structure-relaxation is the rate determining step; and non-Fickian, with comparable diffusion and relaxation rates. In a power-law fit (eq. 8), Case I and Case II systems are characterized by n = 0.5 and n = 1.0, respectively, and the non-Fickian by intermediate values of n. The present data were subjected to this test, after modifying the power law to suit desorption, and restricting the fit to the initial stages of desorption (i.e., small times):

$$1 - \frac{M_t}{M_0} = Kt^n \tag{8}$$

2.3.5 Liner Diffusion Theory

Regardless of diffusion anomalies, all desorption transients were first analyzed using the classic “Crank’s Solution”— for of 1-dimensional Fick’s law diffusion and linear Henry’s law sorption, omitting swelling and concentration dependences [10 (Eq. 4.23)], modified for desorption:

$$\frac{M_t}{M_0} = \frac{8}{\pi^2} \sum_{n=0}^{\infty} \frac{1}{(2n+1)^2} \exp\left(-\frac{(2n+1)^2 \pi^2 Dt}{d^2} \right) \tag{9}$$

Extracting the diffusion coefficient using this series in full or approximations thereof has evolved into an art form, comprehensively summarized by Balik [11]. Consult Ref. 11 for details of the various approximate and exact methods. A précis is given in eq. 10 to 15:

Half-Time:

$$D = \frac{0.04919\ d^2}{t @ \frac{M_t}{M_0} = 0.5} \tag{10}$$

Short-Times:

$$\frac{M_t}{M_0} = 1 - \frac{4}{d}\sqrt{\frac{Dt}{\pi}} \tag{11}$$

Short-Times Series:

$$\frac{M_t}{M_0}=1-\left\{\frac{4}{d}\sqrt{\frac{Dt}{\pi}}+\frac{8}{d}\sqrt{Dt}\sum_{n=1}^{\infty}(-1)^n \mathrm{ierfc}\left(-\frac{nd}{2\sqrt{Dt}}\right)\right\} \tag{12}$$

Long-Times:

$$\ln\left(\frac{M_t}{M_0}\right)=\ln\left(\frac{8}{\pi^2}\right)-\frac{\pi^2 Dt}{d^2} \tag{13}$$

First Moment*:*

$$D=\frac{d^2}{12\int_0^{\infty}(M_t/M_0)dt} \tag{14}$$

Balik [11] introduced two effective combinations of the short- and long-times solutions. Both combinations have the same form (Eq. 15) and the same dimensionless time (Eq. 16), and both are based on the same short-times solution (Eq. 17) and the same long-times solution (Eq. 18).

Balik Solution for the Full Transient:

$$\frac{M_t}{M_0}=1-\{\phi(x)f(x)+[1-\phi(x)]g(x)\} \tag{15}$$

Dimensionless Time**:** $x=\frac{Dt}{d^2}$ (16)

Small-Times Solution**:** $f(x)=4\sqrt{\frac{x}{\pi}}$ (17)

Long-Times Solution**:** $g(x)=1-\frac{8}{\pi^2}\exp\left(-\pi^2 x\right)$ (18)

The Balik solutions differ in how the weighting term $\phi(x)$ is defined—using a Step function (Eq. 19a) or a Fermi function (Eq. 19b).

Weighting Term Using a Step Function To Combine f(x) and g(x)***:***

$$\phi(x)=\begin{cases}1, x\le 0.05326\\ 0, x>0.05326\end{cases} \tag{19a}$$

Weighting Term Using a Fermi Function To Combine f(x) and g(x)***:***

$$\phi(x) = \frac{1}{1+\exp\left(\frac{x-a}{b}\right)} \tag{19b}$$

2.3.6 Evaluating Film Thickness

It is important to re-emphasize that Eq. 9 to 19—normally presented for sorption—have been modified to suit *de*sorption. Also, all the fits yield not "D" directly, but D/d^2. The D estimate is then calculated by multiplication with d^2. The "d" in these equations denotes the constant, full thickness of the disk (as opposed to half-thickness "h" used in a subsequent model that accounts for swelling). Based on the insights gained from the latter model, this work incorporates an improvement over the literature in how "d", even in the "constant-thickness" equations, is evaluated: Traditionally, the Crank's Solution and all the other estimates would be based on d_{dry}, the thickness of the pristine membrane (consistent with ignoring the effects of swelling). When swelling is significant, this approach can confound the D estimate significantly; e.g., by a factor of 12, when the thickness of the swollen membrane at the beginning of the desorption step can be 3.5 times as large as d_{dry}.

Here, for each D/d^2 estimate, a suitable average value of d is used instead of d_{dry}.

The thickness "d" is expressed as a function of the fractional uptake, assuming that swelling is solely a function of the solvent volume fraction (i.e., neglecting any volume change of mixing and omitting the polymer's free volume). The justification for this equation will become apparent in the context of the final result. At this point, it is sufficient to note that ϕ_{s0} is the solvent volume fraction at the beginning of desorption.

$$d = \frac{d_{dry}}{\left[1-\phi_{s0}\frac{M_t}{M_0}\right]} \tag{20}$$

Instead of using the same thickness in every equation, a novel alternative was derived for keeping track of the change in thickness during the time-range of the desorption transient: First, the terminal values of the thickness (at the start and at the end of the time-range that is covered by the fit) are evaluated by substituting the corresponding fractional uptakes in Eq. 20. Next, the effective thickness is set to be the average of the two terminal values, as in Eq. 21.

$$\langle d \rangle = \frac{d_{dry}}{2}\left\{\frac{1}{\left[1-\phi_{s0}\left.\frac{M_t}{M_0}\right|_{start}\right]}+\frac{1}{\left[1-\phi_{s0}\left.\frac{M_t}{M_0}\right|_{end}\right]}\right\} \tag{21}$$

While the authors arrived at this ruse (Eq. 21) by intuition and trial and error, there may be a theoretical justification for it based on using the proper frame of reference when diffusion is accompanied by swelling, as noted rather cryptically in the context of Eq. 10.161 in Ref. 10.

2.3.7 Evaluating End Weight

As noted previously, W_∞ is subject to uncertainty, for instance when diffusion is very slow or the measurements are stopped prematurely. Accordingly, the desorption transients are first fitted using Eq. 9, with W_∞ as an adjustable parameter besides D; the W_∞ to be used in subsequent calculations is then interactively selected by the user from the following choices: $W_{\infty\text{fit}}$, W_F, and the last two measured weights in the desorption transient. In all the present cases, the experimentally measured weights were chosen, either the ultimate weight or the penultimate one (when the time gap between the last two data points far exceeded that between the others). With the chosen W_∞, the data were re-fitted to the various models, with only the diffusion coefficient as the adjustable parameter.

2.3.8 Accounting for Swelling

The membrane thickness is assumed to be a constant in the Crank's Solution and its variants described above. This assumption is tenuous, especially in cases where the initial solvent loading is high. Based on a model of Altinkaya et al [12], treating swelling as a moving-boundary problem and omitting external mass transfer resistances, a numerical solution was developed in this work, specifically aimed at isothermal desorption under large solvent-activity differentials, which is detailed in the following subsections. The present model is a simplified version of the one by Altinkaya. Both models treat swelling as a moving-boundary problem and omit external mass transfer resistances. Key differences are:

(1) Here, heat transfer is assumed to be fast enough to keep the system isothermal, which is reasonable for liquid diffusants (unlike for gases/vapors).
(2) Only one side of the film is open to mass transfer in the cited model; the other is impermeable. Here, both sides are open; by symmetry, the problem needs to be solved for only half the film.
(3) This model is tailor-made for desorption (but can be easily modified for sorption).

2.3.8.1 Solvent Mass Balance

$$\frac{\partial C_s}{\partial t} = \frac{\partial}{\partial x}\left(D_s \frac{\partial C_s}{\partial x} \right) \qquad (22)$$

2.3.8.2 Moving Boundary or Evolution of Film Thickness

The following equation can be derived from the polymer- and solvent-fluxes and a jump boundary condition. In essence, any change in film thickness is due to the polymer flux (equal and opposite to the solvent flux).

$$\frac{dh}{dt} = \underbrace{D_s V_s \left.\frac{\partial C_s}{\partial x}\right|_{x=\pm h}}_{\text{diffusion}} + \underbrace{\frac{dh}{dt} V_s C_s\Big|_{x=\pm h}}_{\text{convection}} \qquad (23)$$

Upon rearrangement, this yields:

$$\frac{dh}{dt} = \frac{D_s\, V_s}{\left(1 - V_s\, C_s\big|_{x=\pm h}\right)} \frac{\partial C_s}{\partial x}\bigg|_{x=\pm h} = \frac{D_s\, V_s}{\left(1 - V_s\, C_{se}\right)} \frac{\partial C_s}{\partial x}\bigg|_{x=\pm h} \qquad (24)$$

2.3.8.3 Boundary Conditions

Sorption Equilibrium at the Film-Solvent Interfaces:

$$x = \pm h : \qquad C_s = C_{se} \qquad (25)$$

C_{se} is specified indirectly through the Flory-Huggins equation:

$$\ln \phi_{se} + \left(1 - \phi_{se}\right) + \chi\left(1 - \phi_{se}\right)^2 = \ln a_{se}\text{, where,} \qquad (26)$$

$$\phi_{se} = C_{se} V_s \quad \text{and,} \qquad (27)$$

$$\phi_s = C_s V_s\,. \qquad (28)$$

The solvent concentration/activity in the film at the interfaces corresponds to the external solvent-vapor partial pressure. For a pure liquid or saturated vapor, $a_{se} = 1$. The proper experimental procedure is differential desorption, decreasing the solvent activity in many small steps from unit- to zero-activity, e.g., using a vacuum microbalance. For integral single-step desorption, however, the external activity is reduced precipitously from $a_{se} = 1$ to $a_{se} = 0$ (and $\phi_{se} = C_{se} = 0$). While this poses theoretical difficulties (concentration dependent diffusion, violation of the thermodynamic reversibility condition that underlies Fick's law, stiff differential equations, etc.) single-step desorption is the norm in practical situations such as drying.

Symmetry condition at the Film Center:

$$x = 0 : \qquad \frac{\partial C_s}{\partial x} = 0 \qquad (29)$$

2.3.8.4 Initial Conditions

Solvent Loading (Uniform):

$$t = 0 : \qquad C_s = C_0 = \frac{1}{V_s}\left(\frac{w_{SP}/\rho_s}{1/\rho_p + w_{SP}/\rho_s}\right) \text{ where, } w_{SP} = \frac{W_0 - W_\infty}{W_\infty} \qquad (30)$$

Film (Half-) Thickness:

$$t = 0 : \qquad h = h_0 \qquad (31)$$

2.3.8.5 Switching to Volume Fractions

Equation 28 $\left(\phi_s = C_s V_s\right)$, verifiable by a units-check, permits a condensation of eq. 22, 24, 25, 29, 30, and 31 into eq. 32, 33, 34, 35, 36, and 37, respectively:

$$\frac{\partial \phi_s}{\partial t} = \frac{\partial}{\partial x}\left(D_s \frac{\partial \phi_s}{\partial x}\right) \tag{32}$$

$$\frac{dh}{dt} = \frac{D_s}{\left(1-\phi_{se}\right)} \frac{\partial \phi_s}{\partial x}\bigg|_{x=\pm h} = \frac{D_s}{\phi_{pe}} \frac{\partial \phi_s}{\partial x}\bigg|_{x=\pm h} \tag{33}$$

$$x = \pm h: \qquad \phi_s = \phi_{se} \ . \tag{34}$$

$$x = 0: \qquad \frac{\partial \phi_s}{\partial x} = 0 \tag{35}$$

$$t = 0: \qquad \phi_s = \phi_{s0} = \frac{w_{SP}/\rho_s}{1/\rho_p + w_{SP}/\rho_s} \text{ where, } w_{SP} = \frac{W_0 - W_\infty}{W_\infty} \tag{36}$$

$$t = 0: \qquad h = h_0 \tag{37}$$

Before solving, it is advisable to transform the moving-boundary into a pseudo fixed-boundary and to non-dimensionalize the equations.

2.3.8.6 Moving-Boundary Transformation

The "Landau Transformation"—of the moving-boundary to a fixed-boundary—consists of normalizing the distance coordinate "x" by the time-dependent film thickness.

$$y \equiv \frac{x}{h} \quad \text{or,} \qquad x = h * y \tag{38}$$

This entails a change of the differentiation operators with respect to distance and time [13].

$$\frac{\partial}{\partial x} = \frac{1}{h}\frac{\partial}{\partial y} \quad \text{and} \quad \frac{\partial}{\partial t}\bigg|_x = \frac{\partial}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial}{\partial t}\bigg|_y = \frac{-x\frac{dh}{dt}}{h^2}\frac{\partial}{\partial y} + \frac{\partial}{\partial t}\bigg|_y = -\frac{y}{h}\frac{dh}{dt}\frac{\partial}{\partial y} + \frac{\partial}{\partial t}\bigg|_y \tag{39}$$

The governing equations and the external boundary condition (Eqs. 32 to 35) change accordingly:

$$\frac{\partial \phi_s}{\partial t} = \frac{1}{h^2}\frac{\partial}{\partial y}\left(D_s \frac{\partial \phi_s}{\partial y}\right) + \frac{y}{h}\frac{dh}{dt}\frac{\partial \phi_s}{\partial y} \tag{40}$$

$$h\frac{dh}{dt} = \frac{D_s}{\phi_{pe}}\frac{\partial \phi_s}{\partial y}\bigg|_{y=\pm 1} \tag{41}$$

$$y=\pm 1: \qquad \phi_s = \phi_{se} \,. \tag{42}$$

$$y=0: \qquad \frac{\partial \phi_s}{\partial y} = 0 \tag{43}$$

2.3.8.7 Nondimensionalization

Scaling can transform the actual variables (which differ by orders of magnitude) into dimensionless variables that are comparable in magnitude, thus avoiding ill-conditioned matrices. Defining:

$$\xi \equiv \frac{h}{h_0}, \tag{44}$$

$$D \equiv \frac{D_s}{D_{ref}}, \tag{45}$$

$$\tau \equiv \frac{D_{ref}\, t}{(h_0)^2}, \text{ and} \tag{46}$$

$$\theta \equiv \frac{\phi_s - \phi_{se}}{\phi_{s0} - \phi_{se}}, \tag{47}$$

the final equations can be listed as a partial differential equation (PDE) coupled with an ordinary differential equation (ODE)—Eq. 48c and Eq. 49, respectively—which have independent variables y and τ and dependent variables θ and ξ:

$$\frac{\partial \theta}{\partial \tau} = \frac{1}{\xi^2}\frac{\partial}{\partial y}\left(D\frac{\partial \theta}{\partial y}\right) + \frac{y}{\xi}\frac{d\xi}{d\tau}\frac{\partial \theta}{\partial y} \tag{48a}$$

or

$$\xi^2 \frac{\partial \theta}{\partial \tau} = \frac{\partial}{\partial y}\left(D\frac{\partial \theta}{\partial y}\right) + y\xi\frac{d\xi}{d\tau}\frac{\partial \theta}{\partial y} \quad \text{or} \tag{48b}$$

$$\xi^2 \frac{\partial \theta}{\partial \tau} = \frac{\partial}{\partial y}\left(D\frac{\partial \theta}{\partial y}\right) + y\frac{\partial \theta}{\partial y} D\left(\frac{\phi_{s0} - \phi_{se}}{1 - \phi_{se}}\right)\frac{\partial \theta}{\partial y}\bigg|_{y=\pm 1} \tag{48c}$$

$$\xi\frac{d\xi}{d\tau} = D\left(\frac{\phi_{s0} - \phi_{se}}{1 - \phi_{se}}\right)\frac{\partial \theta}{\partial y}\bigg|_{y=\pm 1} \tag{49}$$

Boundary Conditions:

$$y=0: \qquad \frac{\partial\theta}{\partial y}=0 \tag{50}$$

$$y=\pm1: \qquad \theta = 0\ . \tag{51}$$

Initial Conditions:

$$t=0: \qquad \theta = 1 \tag{52}$$

$$t=0: \qquad \xi=1 \tag{53}$$

2.3.8.8 .Metrics Derived from Solutions

For a desorption step that goes from time zero to time infinity, the film will have a constant solvent volume fraction throughout, both at the beginning and at the end. The initial value is ϕ_{s0} and the final value is ϕ_{se} . At times in between, there will be a solvent volume fraction *profile*: $\phi_{s0} = \phi_s(y)$.

The dimensionless uptake (which will go from 1 to 0 as "t" spans 0 to ∞) can be expressed as:

$$\frac{M_t}{M_0}=\frac{\left[\int_0^1 \phi_s dy\right]-\phi_{se}}{\phi_{s0}-\phi_{se}}=\int_0^1 \theta dy \tag{54}$$

The film thickness transient is simply:

$$h=h_0\xi \tag{55}$$

2.3.8.9 Expectations

The equations are now ready to be solved, but it is instructive to pause and establish bounds on the anticipated results by means of an approximate analysis, beginning with the moving-boundary equation. It is appropriate to note that these particular bounds are novel, not disclosed in the prior art to the best of the authors' knowledge. For the sake of clarity, the bounds are developed by going back to the original equations (specifically, Eq. 23).

$$\frac{dh}{dt}=\frac{D_s}{\phi_{pe}}\frac{\partial\phi_s}{\partial x}\bigg|_{x=\pm h}=D_s\frac{\partial\phi_s}{\partial x}\bigg|_{x=\pm h}+\frac{dh}{dt}\phi_s\big|_{x=\pm h} \tag{56}$$

Integrating this equation and applying the initial condition for the film thickness results in:

$$h=h_0+\int_0^t\left[D_s\frac{\partial\phi_s}{\partial x}\bigg|_{x=\pm h}+\frac{dh}{dt}\phi_s\big|_{x=\pm h}\right]dt \tag{57}$$

The integral on the right-hand side can be recognized as the cumulative polymer flux (i.e., the negative of the solvent flux), which should equal the cumulative reduction in solvent loading, which in turn is related to the uptake and film-thickness transients.

$$\int_0^t \left[D_s \frac{\partial \phi_s}{\partial x}\bigg|_{x=\pm h} + \frac{dh}{dt}\phi_s\Big|_{x=\pm h} \right] dt = \int_0^h \phi_s(t)\,dx - h_0\,\phi_{s0} = h\int_0^1 \phi_s\,dy - h_0\,\phi_{s0} \qquad (58)$$

Rearranging eq. 54, $\dfrac{M_t}{M_0} = \dfrac{\left[\int_0^1 \phi_s\,dy\right] - \phi_{se}}{\phi_{s0} - \phi_{se}}$,

$$\int_0^1 \phi_s\,dy = (\phi_{s0} - \phi_{se})\frac{M_t}{M_0} + \phi_{se} \qquad (59)$$

Since $h = h_0 + h\left[(\phi_{s0} - \phi_{se})\dfrac{M_t}{M_0} + \phi_{se}\right] - h_0\,\phi_{s0}$, (60)

$$\frac{h}{h_0} = \frac{(1-\phi_{s0})}{\left[(1-\phi_{se}) - (\phi_{s0} - \phi_{se})\dfrac{M_t}{M_0}\right]} \qquad (61)$$

This result has the correct limits, namely, since:

$$\lim_{t\to 0}\frac{M_t}{M_0} = 1 \qquad (62)$$

Substitution of this limit in Eq. 61 yields:

$$\lim_{t\to 0}\frac{h}{h_0} = 1 \qquad (63)$$

Similarly,

$$\lim_{t\to\infty}\frac{M_t}{M_0} = 0 \qquad (64)$$

$$\lim_{t\to\infty}\frac{h}{h_0} = \frac{(1-\phi_{s0})}{(1-\phi_{se})} \qquad (65)$$

The latter limit correctly implies that the polymer moiety of the film thickness stays constant.

$$h_\infty(1-\phi_{se}) = h_0(1-\phi_{s0}) \qquad (66)$$

This is an important benchmark, or a "conservation test", that the numerical solution must meet; that is, the thickness transient from the numerical solution should equal the thickness transient calculated algebraically by substituting the desorption-transient from the numerical solution into Eq. 61. This operation is detailed in Eq. 67:

$$\left.\frac{h}{h_0}\right|_{\text{numerical}} = \left.\frac{h}{h_0}\right|_{\text{algebraic}} = \frac{(1-\phi_{s0})}{\left[(1-\phi_{se})-(\phi_{s0}-\phi_{se})\left.\frac{M_t}{M_0}\right|_{\text{numerical}}\right]} \qquad (67)$$

Conversely, the desorption-transient from the numerical solution should equal the result of substituting the thickness transient from that solution into the converse of Equation 67:

$$\left.\frac{M_t}{M_0}\right|_{\text{numerical}} = \left.\frac{M_t}{M_0}\right|_{\text{algebraic}} = \frac{\left[(1-\phi_{se})-(1-\phi_{s0})\div\left.\frac{h}{h_0}\right|_{\text{numerical}}\right]}{(\phi_{s0}-\phi_{se})} \qquad (68)$$

In the present case of single-step desorption with $\phi_{se} = 0$,

$$\frac{h}{h_0} = \frac{(1-\phi_{s0})}{\left[1-\phi_{s0}\frac{M_t}{M_0}\right]} \quad \text{and} \qquad (69)$$

$$\lim_{t\to\infty}\frac{h}{h_0} = 1-\phi_{s0}\,. \qquad (70)$$

After all the solvent desorbs, the film shrinkage equals the initial solvent volume fraction.

This exercise also provides an approximate swelling model that can be used in parameter estimation: use the Crank's Solution (Eq. 9), but update the film thickness as a function of time based on the changing uptake. This approximation is outlined below.

The linear case involves a series summation any given "t", with a constant "h":

$$\frac{M_t}{M_0} = \frac{8}{\pi^2}\sum_{n=0}^{\infty}\frac{1}{(2n+1)^2}\exp\left(-\frac{(2n+1)^2\pi^2 Dt}{4(h_0)^2}\right) \qquad (71)$$

(Note: Eq. 71 is the same as Eq. 9, with the difference that h_0 here is the half-thickness whereas d in Eq. 9 is the full thickness.) When swelling is included, the series is still summed, but over a fine time-grid, and the thickness is altered by substituting $\frac{M_t}{M_0}$ (from the previous time point) in Eq. 72 and substituting the resulting thickness in Eq. 73:

$$h\big|_t = \frac{h_0(1-\phi_{s0})}{\left[(1-\phi_{se})-(\phi_{s0}-\phi_{se})\left.\frac{M_t}{M_0}\right|_{t-\Delta t}\right]} \tag{72}$$

Swelling Approximation:

$$\frac{M_t}{M_0} = \frac{8}{\pi^2}\sum_{n=0}^{\infty}\frac{1}{(2n+1)^2}\exp\left(-\frac{(2n+1)^2\pi^2 Dt}{4\left(h\big|_t\right)^2}\right) \tag{73}$$

2.3.8.10 Numerical Solution

The coupled partial and ordinary differential equations were solved using the method of lines, as implemented in the Matlab® code "pdepe", which is based on an algorithm put forward by Skeel and Berzins [14, 15]. The spatial derivatives were discretized using finite differences, and the resulting ODEs were solved using the Matlab® ODE solver ODE15s (which is designed to handle "stiff problems" that involve disparate scales in time or space).

Significant work was involved in customizing the standard Matlab® solver to suit the peculiar source terms. Also nonstandard, but apparently effective, was the innovative use of a routine for solving coupled PDEs to solve the present PDE-ODE couple.

2.3.8.11 Using Numerical Solution for Parameter Estimation

As is discussed later, the simple Crank's Solution with a constant thickness (Eq. 9) cannot be expected to fit the data for high solvent/polymer ratio cases. The ad hoc modification of the Crank's Solution (adjusting thickness as a function of loading by Eq. 72, assuming that swelling is related to solvent volume fraction) can offer an excellent fit, but is theoretically unsound, yielding a D-estimate that is not physically meaningful. It would be best to fit the data to the Altinkaya model that accounts for swelling exactly, but the numerical solution is too computer-time intensive to be called numerous times by the parameter-estimation algorithm. The breakthrough out of this impasse came as the following ruse: solve the numerical solution only once, and tabulate the results in dimensionless form—fractional uptake versus dimensionless time:

$$\frac{M_t}{M_0} \quad \text{vs.} \quad \tau \equiv \frac{D_{ref}\, t}{(h_0)^2} \tag{74}$$

The function-evaluation routine (which gets called by the parameter estimation routine many times, with different values for the parameter D at each iteration) first converts the experimental real times into dimensionless times and evaluates the corresponding fractional uptake by interpolating the tabulated numerical solution. These uptakes then constitute the "theoretical" desorption transient, which is compared with the experimental one, and the sum-of-squares of errors (SSE) is calculated; D is altered using the Levenberg-Marquardt algorithm until SSE is minimal. This fitting-via-interpolation was first validated by applying it to the Crank's Solution and the swelling approximation cases; it yielded D estimates that were identical to the estimates in which the fractional uptakes were calculated afresh at each iteration while converging to the best fit using the model in question.

2.4 RESULTS

2.4.1 Sorption

The solvent-to-additive and solvent-to-polymer ratios are summarized in Table 1.

**Table 1 Sorption Loading Ratios:
Solvent to Additives and Solvent to Polymer**

SOLVENT	S/A	S/P
Acetonitrile	0.96	0.05
1Butanol	1.10	0.05
2-Ethoxyethanol	1.36	0.04
N,N-Dimethylformamide	4.48	0.06
N,N-Dimethylacetamide	2.85	0.13
Ethylene Glycol Butyl Ether	2.17	0.12
1-Methyl-2-Pyrrolidinone	3.69	0.15
Benzonitrile	2.78	0.19
Benzaldehyde	45.10	0.50
Ethyl Acetate	4.05	0.17
1,2-Dichloroethane	4.54	0.24
Butylamine	6.79	0.48
Dichloromethane	13.96	0.86
Benzene	10.14	0.78
Hexane	15.88	0.78
Heptane	17.40	0.80
Tetrahydrofuran	19.76	1.26
Triethylamine	23.49	1.01
para Xylene	22.06	1.31
Mesitylene	36.34	1.19
Chloroform	94.55	2.37
Trichloroethylene	43.60	2.93
Tetrachloroethylene	90.26	3.30

S/A = Solvent to additives

S/P = Solvent to polymer

The range of ϕ_p values for several solvents in the present butyl rubber sample is shown in Table 2, along with the solubility parameters for the solvents.

Table 2 Sorption Equilibrium Properties

SOLVENT	φ_P	δ_{dS} (MPa$^{0.5}$)	δ_{pS} (MPa$^{0.5}$)	δ_{hS} (MPa$^{0.5}$)	V_S (cm^3/gmole)
Acetonitrile	0.932	15.3	18.0	6.10	52.2
1-Butanol	0.926	16.0	5.7	15.80	91.5
2-Ethoxyethanol	0.949	16.2	9.2	14.30	96.9
N,N-Dimethylformamide	0.936	17.4	13.7	11.30	73.1
N,N-Dimethylacetamide	0.855	16.8	11.5	10.20	92.7
Ethylene Glycol Butyl Ether	0.858	16.0	5.1	12.30	131.3
1-Methyl-2-Pyrollidinone	0.850	18.0	12.3	7.20	96.4
Benzonitrile	0.808	17.4	9.0	3.30	103.0
Benzaldehyde	0.645	19.4	7.4	5.30	101.8
Ethyl Acetate	0.813	15.8	5.3	7.20	97.7
1,2-Dichloroethane	0.786	19.0	7.4	4.10	78.8
Butylamine	0.520	16.2	4.5	8.00	98.8
Dichloromethane	0.537	18.2	6.3	6.10	64.1
Benzene	0.450	18.4	0.0	2.00	88.8
Hexane	0.391	14.9	0.0	0.00	130.8
Heptane	0.410	15.3	0.0	0.00	146.5
Tetrahydrofuran	0.357	16.8	5.7	8.00	81.1
Triethylamine	0.362	17.8	0.4	1.00	138.6
Para Xylene	0.343	17.8	0.0	2.66	122.6
Mesitylene	0.338	18.0	0.0	0.60	138.5
Chloroform	0.331	17.8	3.1	5.70	80.0
Trichloroethylene	0.283	18.0	3.1	5.30	90.0
Tetrachloroethylene	0.283	19.0	6.5	2.90	102.0

The results of solubility-data regression are displayed in Figure 3. The same analysis procedure was also applied to the data in Refs. 7 and 8. The coefficients are compared in Table 3.

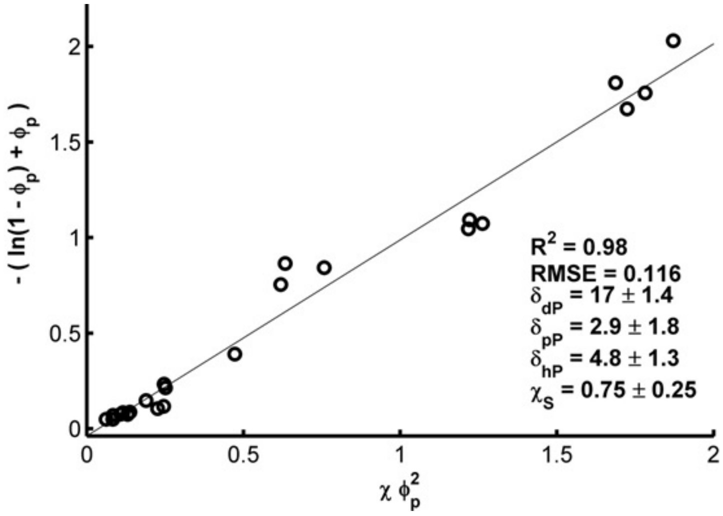


Figure 3 Regression of Sorption Data

Table 3 Comparison of Solubility Parameters

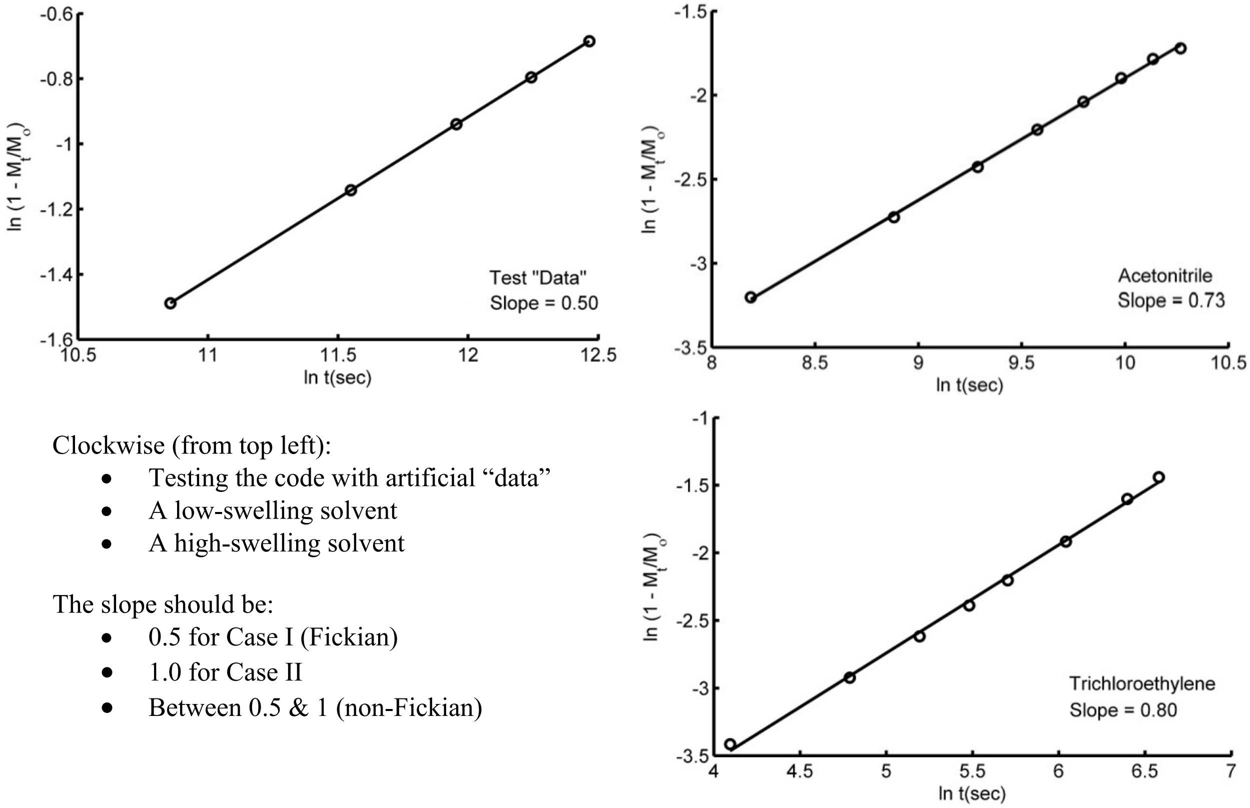
DATA SOURCE	ρ_p (g/cm^3)	δ_{dP} (MPa)$^{0.5}$	δ_{pP} (MPa)$^{0.5}$	δ_{hP} (MPa)$^{0.5}$	χ_S	R^2	RMSE
This work	1.225	17 ± 1.4	2.9 ± 1.8	4.8 ± 1.3	0.75 ± 0.25	0.976	0.116
Butyl [7]	1.263	19 ± 1.6 (17.3)	4.2 ± 2.5 (4.3)	3.8 ± 1.9 (3.4)	1.20 ± 0.45	0.925	0.336
"Best" [8]	1.074	18 ± 1.5 (18.13)	2.7 ± 2.2 (2.71)	7.5 ± 1.8 (7.55)	1.20 ± 0.32 (1.23)	0.961 (0.961)	0.217

Note: The numbers in parentheses are the coefficients reported in Refs.. 7 and 8. The numbers outside parentheses are estimates based on the raw data reported in these references.

2.4.2 Diffusion

2.4.2.1 Diffusion Anomalies

Examples of the power-law fit are shown in Figure 4. The slopes are listed in Table 4.



Clockwise (from top left):

- Testing the code with artificial "data"
- A low-swelling solvent
- A high-swelling solvent

The slope should be:

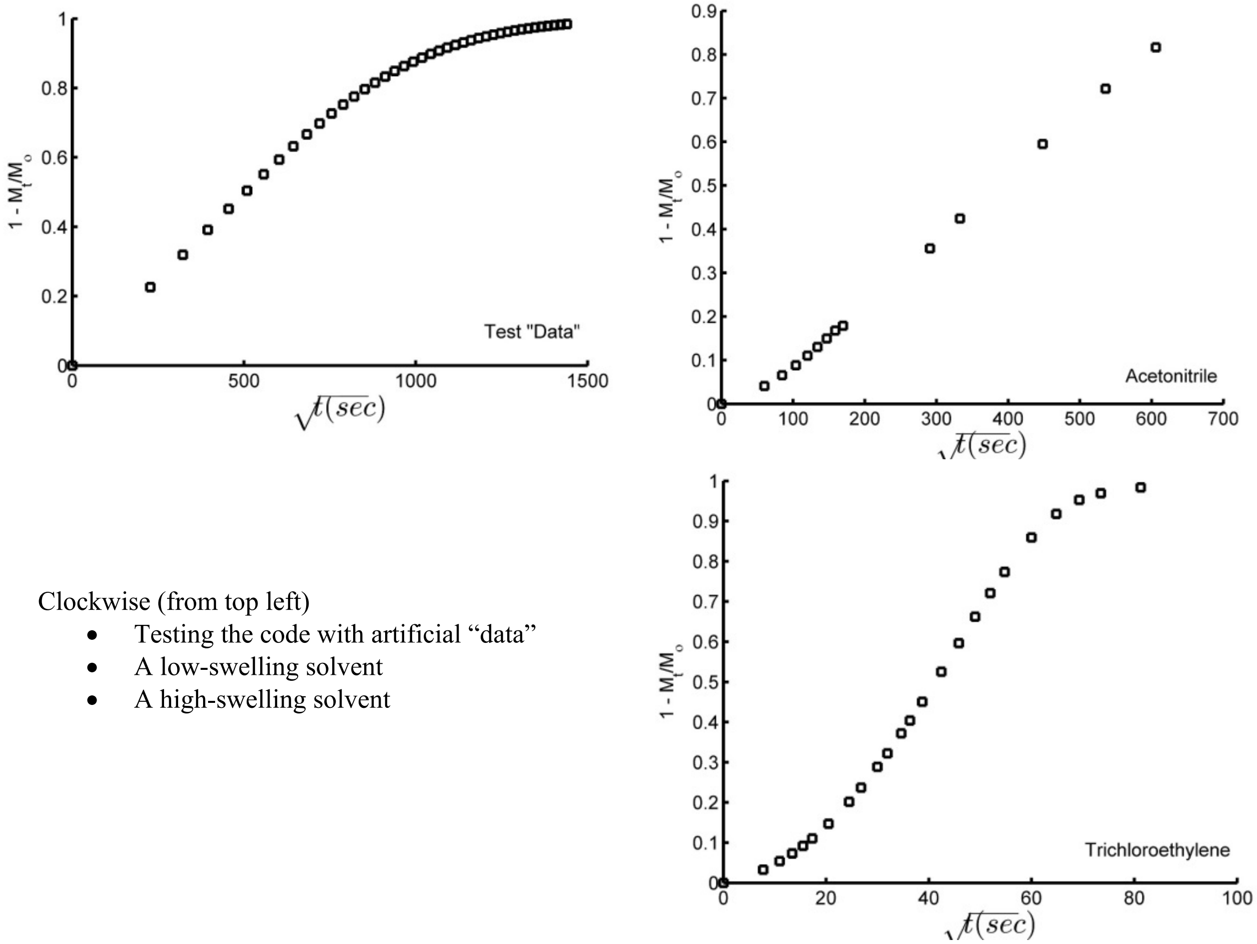
- 0.5 for Case I (Fickian)
- 1.0 for Case II
- Between 0.5 & 1 (non-Fickian)

Figure 4 Probing for Anomalous Diffusion

Table 4 Power-Law Coefficients

SOLVENT	n (Eq. 8)
Acetonitrile	0.74
1Butanol	0.41
2-Ethoxyethanol	0.48
N,N-Dimethylformamide	0.45
N,N-Dimethylacetamide	0.42
Ethylene Glycol Butyl Ether	0.44
1-Methyl-2-Pyrrolidinone	1.32
Benzonitrile	0.58
Benzaldehyde	0.72
Ethyl Acetate	0.81
1,2-Dichloroethane	0.70
Butylamine	0.68
Dichloromethane	0.60
Benzene	0.76
Hexane	0.67
Heptane	0.74
Tetrahydrofuran	0.49
Triethylamine	0.83
para Xylene	0.88
Mesitylene	0.84
Chloroform	0.76
Trichloroethylene	0.80
Tetrachloroethylene	0.75

Towards sorting the sorption transients by visual inspection into pseudo-Fickian, sigmoid, and other shapes, examples from the present data are given in Figure 5.



Clockwise (from top left)

- Testing the code with artificial "data"
- A low-swelling solvent
- A high-swelling solvent

Figure 5 Another Probe of Anomalous Diffusion

2.4.2.2 Film Thickness

By way of keeping track of the changes in film thickness during the course of desorption, the thickness is calculated as a function of the fractional uptake (using Eq. 21). The average "d" values corresponding to the different sections of the desorption transient, normalized by the dry membrane thickness, are listed in Table 5.

Table 5 Estimated Thicknesses for Different Time Ranges

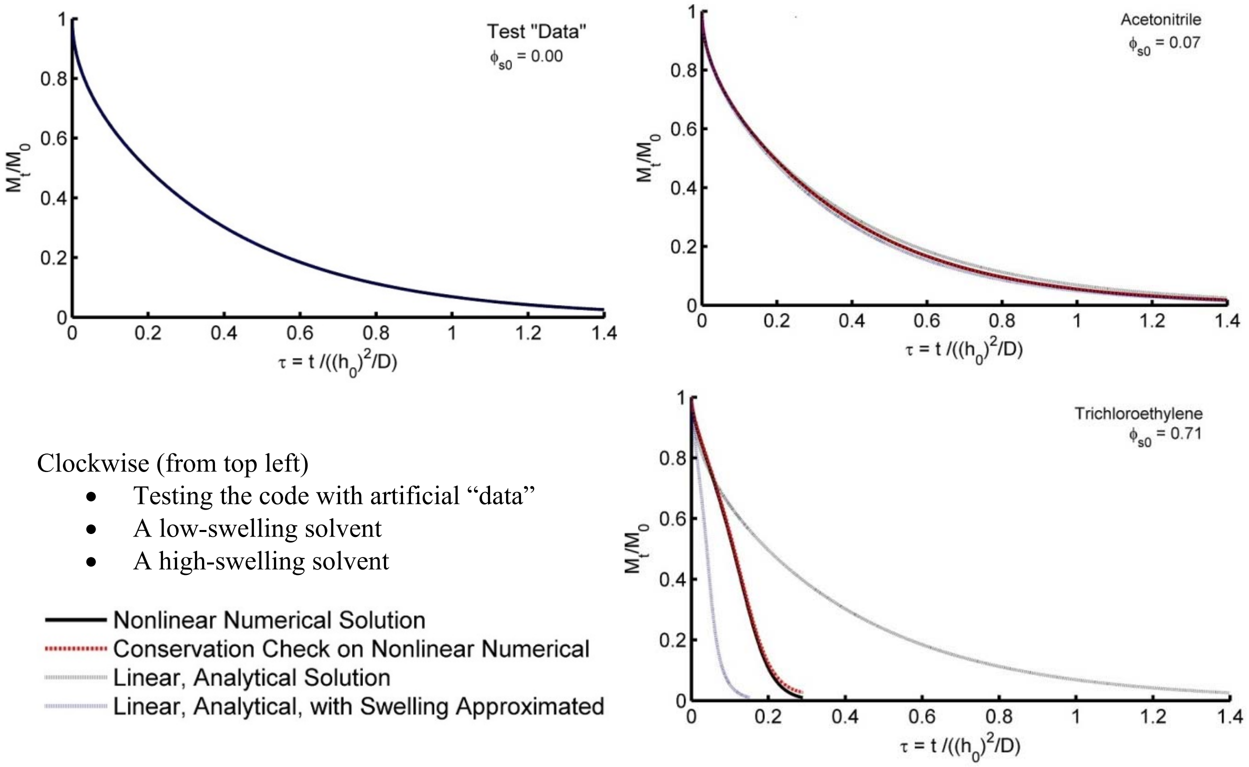
SOLVENT	d_{dry}	$d_{\mathrm{initial}}/d_{\mathrm{dry}}$	$d_{\mathrm{short}}/d_{\mathrm{dry}}$	$d_{\mathrm{half}}/d_{\mathrm{dry}}$	$d_{\mathrm{long}}/d_{\mathrm{dry}}$
Acetonitrile	0.72	1.1	1.1	1.0	1.0
1Butanol	0.69	1.1	1.1	1.0	1.0
2-Ethoxyethanol	0.71	1.1	1.0	1.0	1.0
N,N-Dimethylformamide	0.70	1.1	1.1	1.0	1.0
N,N-Dimethylacetamide	0.69	1.2	1.2	1.1	1.0
Ethylene Glycol Butyl Ether	0.69	1.2	1.1	1.1	1.0
1-Methyl-2-Pyrrolidinone	0.71	1.2	1.2	1.1	1.0
Benzonitrile	0.69	1.2	1.2	1.1	1.0
Benzaldehyde	0.67	1.6	1.5	1.3	1.0
Ethyl Acetate	0.76	1.2	1.2	1.2	1.1
1,2-Dichloroethane	0.77	1.2	1.2	1.1	1.0
Butylamine	0.71	1.8	1.5	1.4	1.1
Dichloromethane	0.76	1.8	1.5	1.4	1.1
Benzene	0.74	2.1	1.8	1.6	1.1
Hexane	0.75	2.5	1.9	1.7	1.1
Heptane	0.74	2.4	2.2	1.7	1.1
Tetrahydrofuran	0.73	2.7	1.9	1.9	1.0
Triethylamine	0.71	2.7	2.2	1.9	1.1
para Xylene	0.74	2.9	2.7	1.9	1.3
Mesitylene	0.72	2.7	2.6	1.9	1.4
Chloroform	0.71	3.0	2.3	2.0	1.1
Trichloroethylene	0.73	3.5	3.0	2.2	1.1
Tetrachloroethylene	0.72	3.5	3.2	2.3	1.2

Note: The values are based on the terminal uptakes (Eq. 21) of the time-range.

2.4.2.3 Diffusion Coefficients

Just as in the case of Figures 4 and 5, shown first in Figures 6 to 11 are demonstrations of the curve fitting, using test “data” artificially generated using the Crank’s Solution, Eq. 9. Shown in the subsequent two plots in each figure are samples of the curve fitting using actual data, one for a low-swelling solvent and another for a high-swelling solvent.

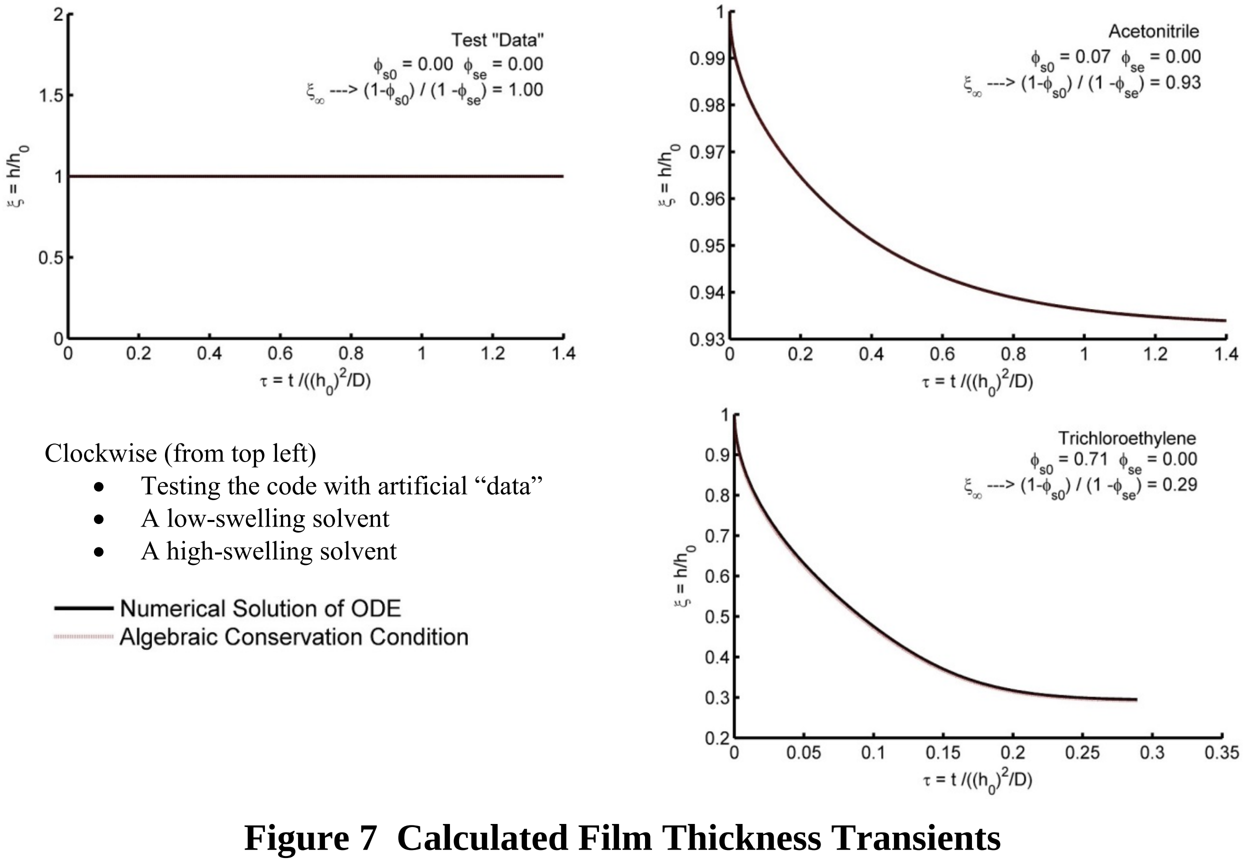
In Figures 6 and 7 are displayed the sorption and film-thickness transients, respectively, as a function of the dimensionless time. As noted previously, these results are relied upon later in diffusivity estimation.



Clockwise (from top left)

- Testing the code with artificial “data”
- A low-swelling solvent
- A high-swelling solvent

Figure 6 Calculated Dimensionless Desorption Transients

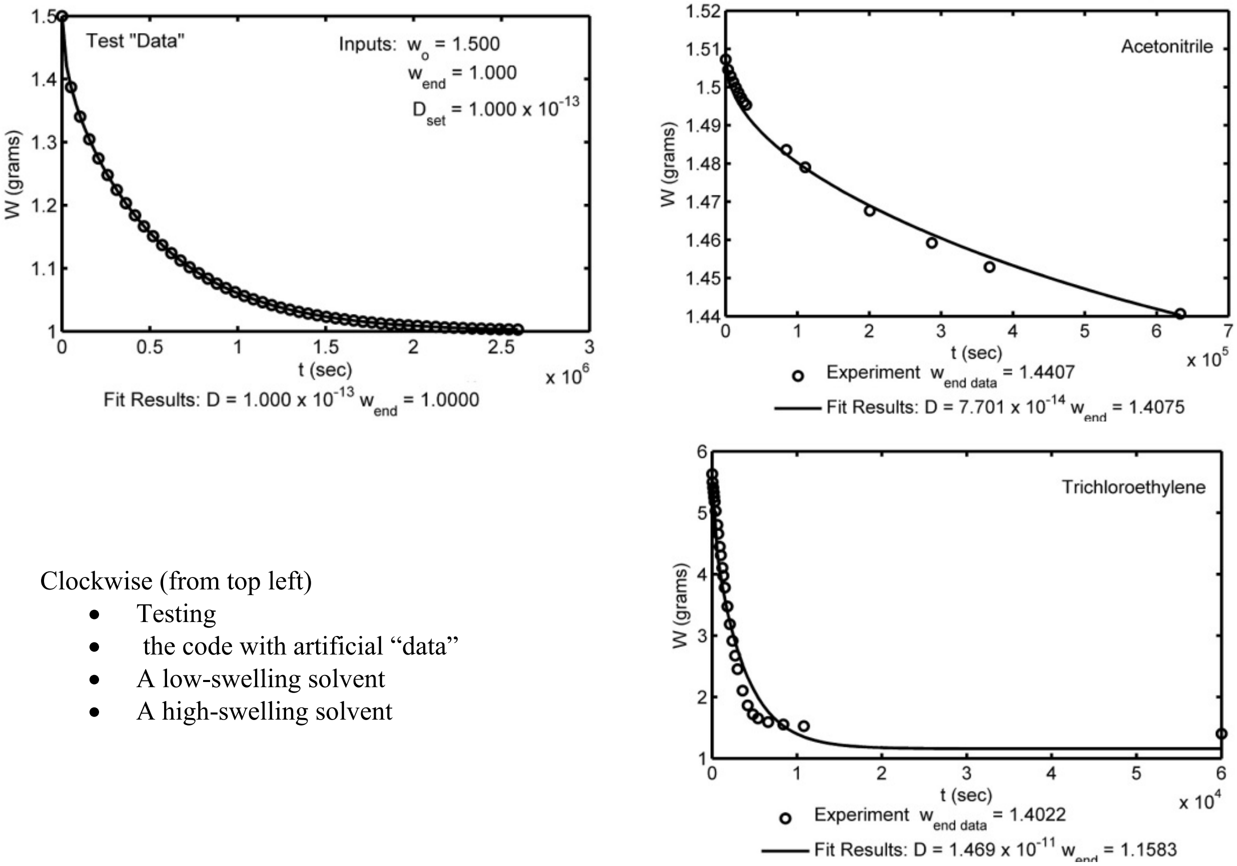


Clockwise (from top left)

- Testing the code with artificial “data”
- A low-swelling solvent
- A high-swelling solvent

Figure 7 Calculated Film Thickness Transients

Shown in Figure 8 are generalized fits where the end-weight W_∞ is also treated as an adjustable parameter, as well as the diffusion coefficient. The resulting theoretical estimate offers an additional choice for the end-weight and serves as a check on the measured value.

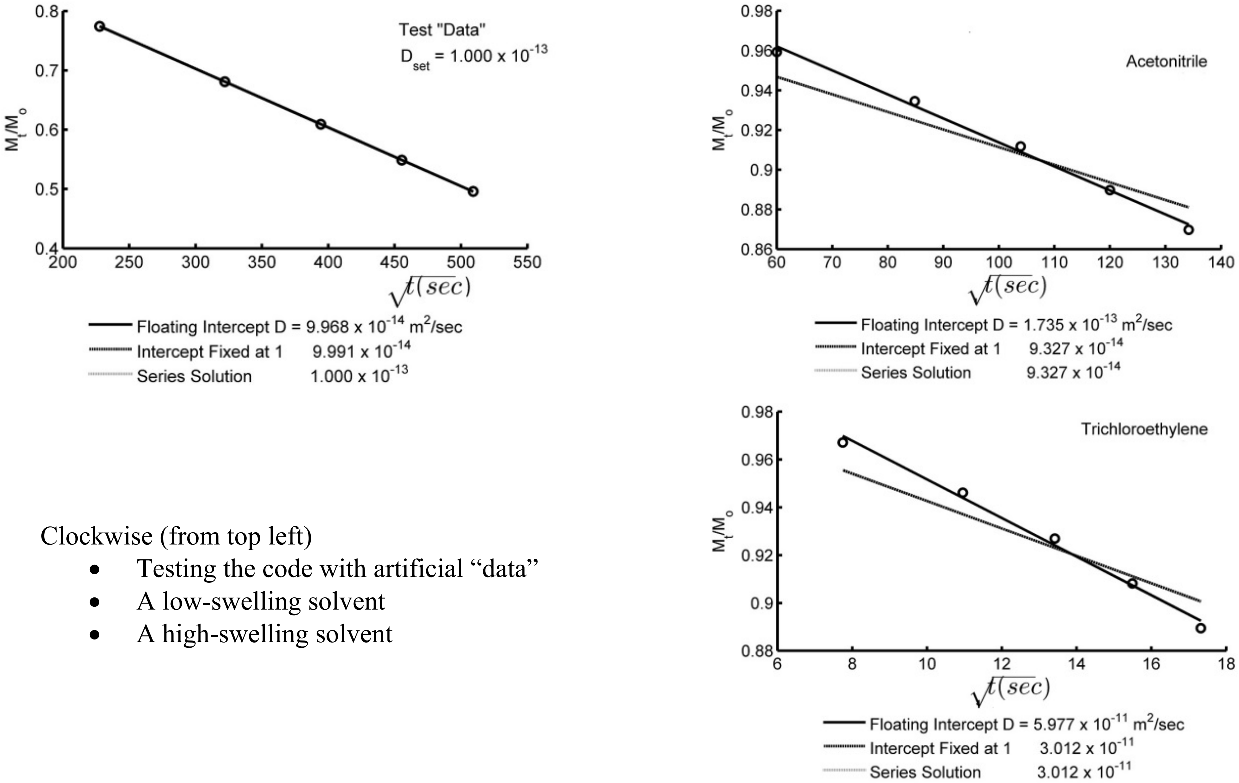


Clockwise (from top left)

- Testing
- the code with artificial “data”
- A low-swelling solvent
- A high-swelling solvent

Figure 8 Treating End-Weight as a Parameter

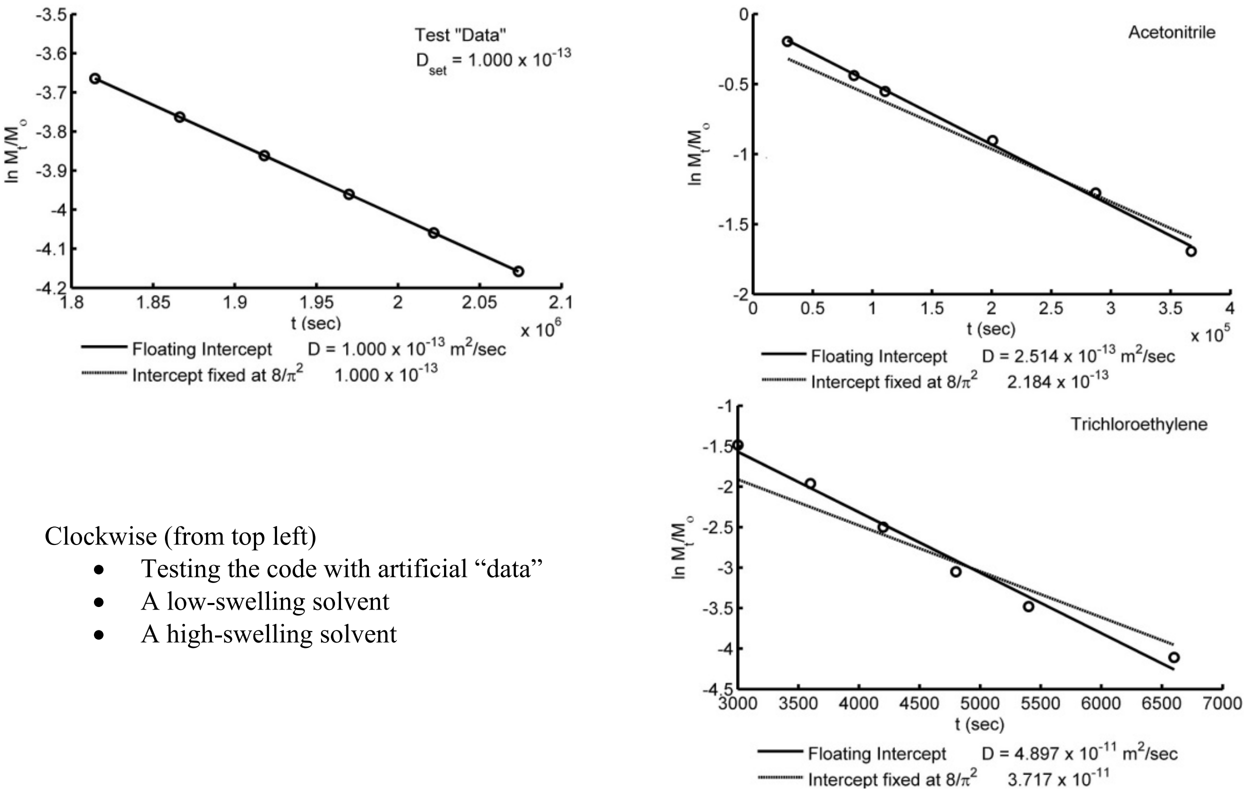
Shown in Figures 9 and 10 are the fits of selected time-ranges of the data—short and long, respectively. These linear fits include the variations of keeping the intercepts floating or fixed. The corresponding diffusivity estimates are listed in Table 6. Finally, shown in Figure 11 are the all-times fits (of the entire desorption transient) to the linear Crank’s Solution, as well as the nonlinear swelling model. The fits to the two Balik approximations are indistinguishable from the Crank’s Solution fit and hence are not shown, for the sake of clarity. The resulting diffusivity estimates are summarized in Table 7.



Clockwise (from top left)

- Testing the code with artificial "data"
- A low-swelling solvent
- A high-swelling solvent

Figure 9 Short-Times Fits



Clockwise (from top left)

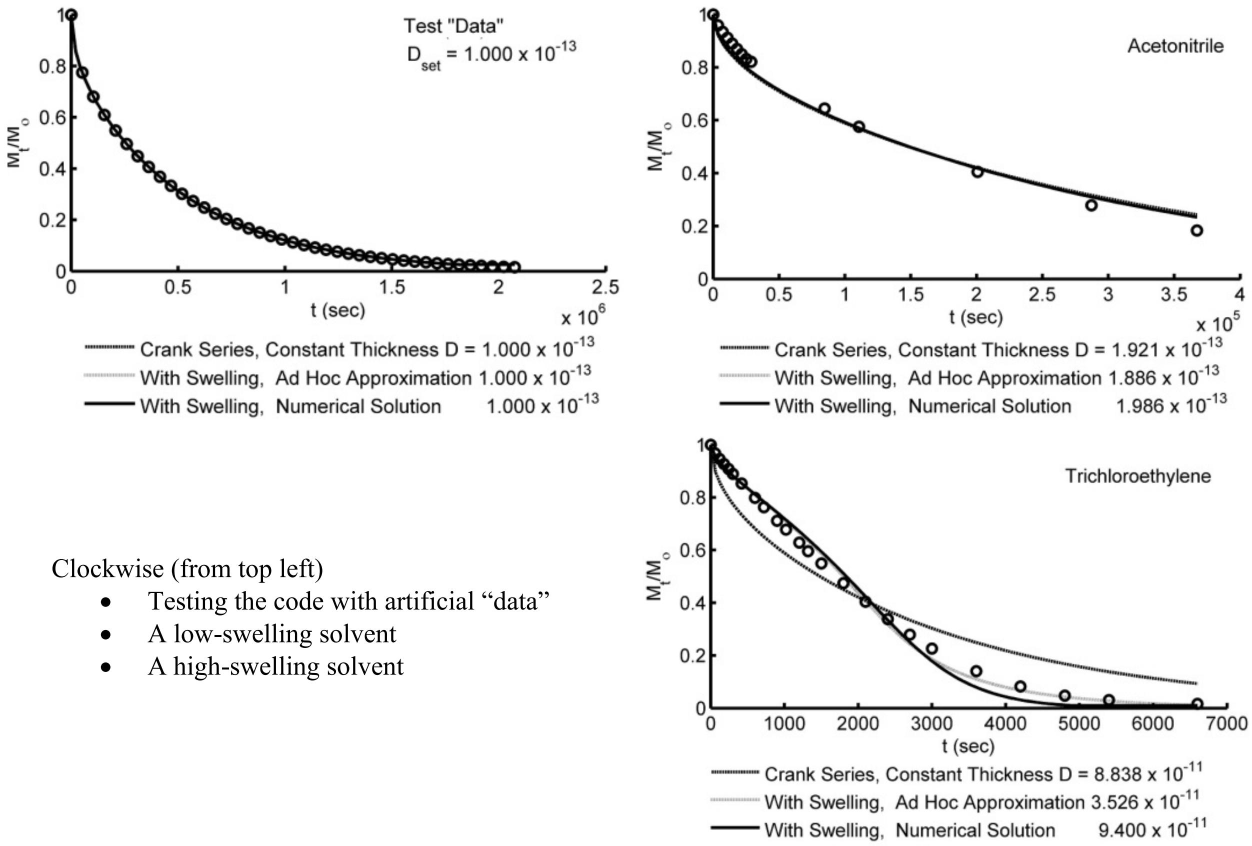
- Testing the code with artificial "data"
- A low-swelling solvent
- A high-swelling solvent

Figure 10 Long-Times Fits

Table 6 Diffusivity Estimates from Limited Time Ranges of the Data

Units: $m^2/s * 10^{12}$

SOLVENT	HALF-TIME	SHORT-TIMES			LONG-TIMES		MOMENT
	Eq. 10	Floating Intercept Eq. 11	Intercept = 1 Eq. 11	Series Eq. 12	Floating Intercept Eq. 13	Intercept = $\ln(8/\pi^2)$ Eq. 13	Eq. 14
Acetonitrile	0.186	0.175	0.091	0.091	0.249	0.216	0.272
1Butanol	0.074	0.047	0.078	0.078	0.138	0.090	0.090
2-Ethoxyethanol	0.103	0.093	0.098	0.099	0.178	0.119	0.119
N,N-Dimethylformamide	0.091	0.108	0.113	0.113	0.129	0.090	0.100
N,N-Dimethylacetamide	0.088	0.035	0.048	0.048	0.094	0.079	0.093
Ethylene Glycol Butyl Ether	0.113	0.095	0.129	0.130	0.112	0.101	0.122
1-Methyl-2-Pyrrolidinone	0.093	0.007	0.002	0.002	0.093	0.077	0.094
Benzonitrile	0.178	0.227	0.189	0.191	0.189	0.188	0.237
Benzaldehyde	0.424	0.578	0.365	0.365	0.101	0.149	0.342
Ethyl Acetate	8.340	6.670	3.300	3.300	9.880	8.080	14.950
1,2-Dichloroethane	22.100	26.600	16.000	16.000	19.300	18.700	22.500
Butylamine	76.900	122.400	72.300	72.300	32.900	35.900	64.100
Dichloromethane	112.700	143.500	111.000	111.000	50.900	61.300	117.100
Benzene	87.000	119.100	60.000	60.000	57.400	53.500	101.300
Hexane	132.800	234.600	132.500	132.500	42.900	50.800	142.600
Heptane	49.100	48.000	26.400	26.400	27.400	26.000	61.800
Tetrahydrofuran	189.600	177.500	175.400	175.400	26.400	40.400	171.200
Triethylamine	97.700	121.400	54.800	54.800	20.800	31.800	107.000
para Xylene	14.200	3.300	1.300	1.300	17.400	12.200	20.600
Mesitylene	8.940	1.040	0.480	0.480	14.810	9.560	14.320
Chloroform	130.800	174.000	88.900	88.900	42.500	49.900	162.500
Trichloroethylene	77.100	59.800	30.100	30.100	49.000	37.200	112.900
Tetrachloroethylene	26.700	11.600	4.200	4.200	27.400	14.400	41.300



Clockwise (from top left)

- Testing the code with artificial “data”
- A low-swelling solvent
- A high-swelling solvent

Figure 11 All-Times Fits

Table 7 Diffusivity Estimates Based on Entire Desorption Transient

Units: m^2/s * 10^{12}

SOLVENT	BALIK'S APPROXIMATIONS		CRANK'S SOLUTION		NUMERICAL SOLUTION	D_{chosen}	
	Step, Eq. 19a	Fermi, Eq. 19b	Swelling, Approximate	Linear, Eq. 9	With Swelling, Exact	Estimate	± %
Acetonitrile	0.189	0.189	0.186	0.189	0.196	0.196	12.26
1Butanol	0.080	0.080	0.078	0.081	0.083	0.083	2.86
2-Ethoxyethanol	0.113	0.113	0.111	0.113	0.115	0.115	2.10
N,N-Dimethylformamide	0.089	0.089	0.087	0.089	0.092	0.089	2.32
N,N-Dimethylacetamide	0.086	0.086	0.079	0.086	0.088	0.086	2.14
Ethylene Glycol Butyl Ether	0.120	0.120	0.111	0.120	0.125	0.121	1.45
1-Methyl-2-Pyrrolidinone	0.086	0.086	0.080	0.086	0.089	0.086	2.35
Benzonitrile	0.223	0.223	0.197	0.223	0.230	0.230	2.46
Benzaldehyde	0.387	0.387	0.319	0.387	0.447	0.387	5.23
Ethyl Acetate	7.360	7.360	7.090	7.360	8.260	8.260	7.44
1,2-Dichloroethane	20.800	20.800	19.900	20.900	23.000	23.000	3.96
Butylamine	68.200	68.200	53.100	68.200	82.000	68.200	6.42
Dichloromethane	114.100	114.100	83.700	114.100	129.400	114.100	3.13
Benzene	89.400	89.400	60.600	89.500	105.000	105.000	5.94
Hexane	138.300	138.300	82.000	138.300	161.800	138.300	5.70
Heptane	50.700	50.700	29.900	50.700	59.300	59.300	4.67
Tetrahydrofuran	190.200	190.200	100.100	190.300	213.200	190.300	3.06
Triethylamine	103.300	103.300	52.700	103.400	111.900	111.900	7.82
para Xylene	14.100	14.100	7.700	14.100	17.300	17.300	6.37
Mesitylene	8.860	8.860	5.350	8.870	11.530	11.530	6.82
Chloroform	148.400	148.400	66.700	148.500	153.300	153.300	5.60
Trichloroethylene	88.300	88.300	35.400	88.400	94.000	94.000	3.83
Tetrachloroethylene	29.700	29.700	12.200	29.800	32.700	32.700	3.19
Column numbers →	1	2	3	4	5	6	7

2.5 DISCUSSION

2.5.1 General Observations

The solvent-to-additive and solvent-to-polymer ratios listed in Table 1 show that roughly half of the solvents tested are much more strongly absorbed by butyl rubber than the other half. (Incidentally, butyl rubber may not be a good protective barrier against these highly absorbing solvents.) Also striking is the observation that, for the weakly absorbing solvents, the solvent uptake is comparable to the amount of additives. This observation supports the choice of measuring desorption (instead of sorption) transients; presumably, the additives would have leached out during the soaking step that precedes desorption, thus not confounding the measurements.

As is apparent from the diffusion results discussed later, this division of the solvents (into low and high loadings) is matched by a division into whether the data are fitted well by constant-thickness models. Ideally, desorption measurements should be done in a differential manner, decreasing the external solvent activity in many small steps from unity to zero (e.g., by decreasing the external solvent vapor pressure gradually in a vacuum microbalance). As noted previously, the precipitous reduction in the solvent activity at the surfaces poses theoretical difficulties such as concentration dependent diffusion and changes in thickness due to swelling. Hence, it renders questionable the use of the Crank's Solution (based on Fick's law diffusion with a constant diffusion coefficient and linear Henry's law sorption). The results are still useful, however, since integral single-step desorption is the norm in practical situations. Also, much past work on sorption butyl rubber (e.g., Refs. 7 and 8) was also integral, with direct exposure of the membrane to liquid solvents (i.e., zero to unit activity).

The average thicknesses (for various sections of the desorption transient) listed in Table 5 are quite different from the dry membrane thickness, especially for the high-swelling cases. This highlights the perils of constant-thickness models and supports the use of average thicknesses.

2.5.2 Solubility

Shown in Table 3 are two separate comparisons of solubility parameter estimates:

(1) The coefficients estimated using the present data analysis algorithm applied to the raw data from three different sources: this project, Ref. 7, and Ref. 8 (first, second, and third rows, respectively).

(2) The coefficients estimated using the present algorithm and those reported in the literature based on alternative procedures: graphical [7] and least-squares [8] (second and third rows, respectively).

Both comparisons are satisfactory, although the standard deviations of the parameters are troublingly large, pointing to the need for a better solubility theory. Incidentally, the diffusivity estimates to be discussed next are totally independent of these solubility-parameter estimates.

2.5.3 Diffusion Anomalies

As noted previously, diffusion in polymers is Case I (Fickian), in which diffusion is slower than the relaxation of the membrane structure; Case II, where diffusion is fast and structure-relaxation is the rate determining step; or non-Fickian, with comparable diffusion and relaxation rates. Describing the amount absorbed at time t by the power law Kt^n, Case I and Case II systems are characterized by n = 0.5 and n = 1.0, respectively. Examples of the fit for the present study are shown in Figure 4. The slopes for all cases are listed in Table 4. Most of the cases are non-Fickian. Also, there is no clear example for Case II transport (n = 1), which is not surprising since such transport occurs in glassy polymers and the system here is rubbery. With the other probe (Figure 5) of diffusion anomalies (namely, visual inspection of the sorption transients), some cases are pseudo-Fickian, and some are noticeably sigmoid.

2.5.4 Diffusion Coefficients

As can be seen in the first plot in of each of Figures 9, 10, and 11, the curve-fitting algorithms per se "pass the test", since all the estimation methods yield nearly identical estimates (as they should) when the underlying "data" are simulated using an exact solution. Results for the experimental data (shown in the second and third plots in each of Figures 9, 10, and 11), however, display considerable variability depending on the time-region of the transient included in the fit; some comments are in order. The curve-fitting algorithms are not causing the variations in the estimates, as demonstrated with the test "data." So, any variability has to do with data precision and model accuracy, within each sample and across solvents. Also, even within one desorption transient for any solvent, not all data points can be expected to have the same precision or accuracy. The long-time data suffer from diminishing marginal returns (i.e., less and less change in mass as desorption time increases), which manifest as an increase in the relative error of weighing as the weights become smaller. Similarly, the small-time data will be sensitive to errors in time measurements. In sum, solutions focusing on different time ranges will yield different estimates because some data points are more susceptible to experimental error than others. These susceptibilities are magnified when the fitting (at short-times and long-times) is done without fixing the intercepts in strict accordance with the corresponding approximate solutions. These problems of imprecision in the data are further compounded when nonlinearities are manifest: swelling, change in sample dimensions, etc.

For cases of low solvent-to-polymer ratio, based on the ability of the Crank's Solution to fit the data, it is reasonable to conclude that the physics is quite linear. For cases of high solvent-to-polymer ratio, the Altinkaya model of accounting for (de-)swelling fits the data quite well. Unfortunately, the thickness was not measured along with the weight, during desorption. Hence, other causes (for departures from the Crank's Solution) may not be ruled out: e.g., concentration-dependent diffusion coefficients, nonisothermality, or slow evaporation [10 (Section 4.3.6)].

It is remarkable that, even when the fits are drastically different (as can be seen in the Figures 9, 10, and 11), the D estimates from the constant-thickness Crank's Solution are quite similar to the estimates based on the numerical solution of the Altinkaya model of swelling. This agreement probably is due to the use (Eq. 21) of the average thickness (instead of the dry membrane thickness) to extract D from D/d^2. As alluded to previously in the context of Eq. 21, this robustness of the D estimates may have a theoretical justification, based on using the proper frame of reference when diffusion is accompanied by swelling [10 (Eq. 10.161)]. This point deserves further study.

The plethora of approximate estimates is included for completeness; one of the all-times fits is chosen as the best diffusivity estimate from the Crank's Solution or from the numerical solution of the Altinkaya model of swelling, whichever has the smaller root-mean-square (RMS) error. These D_{chosen} estimates and the corresponding 95% confidence limits are shown in the last two columns of Table 7.

2.5.5 Data Diversity

As is described in Chapter 4, these diffusivity estimates were used as the target data (dependent variable) for training the neural network, along with the molecular descriptors of these solvents as the independent variables; the network was then used to make data-driven target predictions for CWA and NTA molecules for which only structural information is available. In this context, it is of interest to ask: How diverse (or informative) is the diffusivity database? A qualitative answer is displayed in Figure 12.

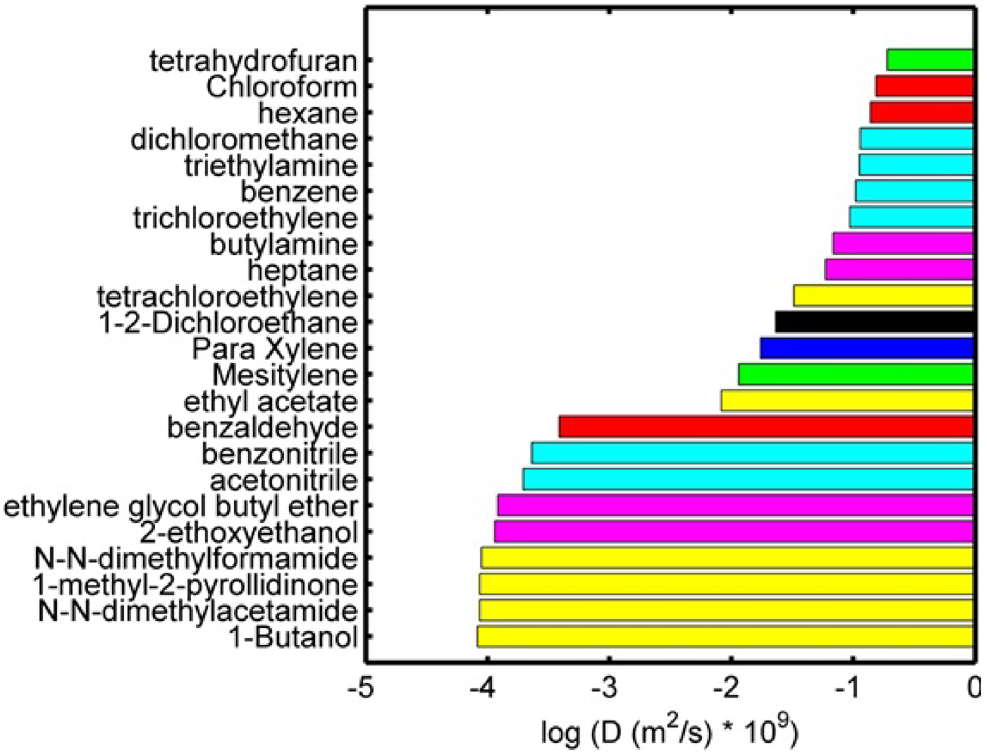


Figure 12 Data Diversity in Perspective

Adjacent bars of similar magnitude are given the same color in Figure 12, to highlight data "degeneracy". For example, the bottom four bars in yellow are nearly identical in size; in effect, four molecules condense into one. In sum, even though the number of solvents is 23 and the diffusivity values range over a factor of 500, the number of distinct target data points shrinks to about 13. Also, there is a sizeable gap in the database, in going from ethyl acetate to benzaldehyde. Such handicaps may detract from the accuracy of data-driven predictions and need to be mitigated with more comprehensive data.

CHAPTER 3 DESCRIPTORS

3.1 BACKGROUND

The “target” properties detailed in Chapter 2 constitute the dependent variables for establishing or “training” the technique for data-driven prediction of how CWAs permeate in protective barrier materials which is described in Chapter 4. Training also requires independent variables, namely molecular properties (or “descriptors”) of the solvents which are the “data chemicals”. Descriptors also need to be calculated for the “query chemicals”, for which the target properties are going to be predicted using the trained method. This chapter describes how to develop this database of descriptors for data and query chemicals, starting from molecular formulas, using commercial computational chemistry software (and Matlab® codes developed in-house to customize the descriptor database). Descriptors can also be calculated using other software or imported from websites.

3.2 METHODS DEVELOPMENT

3.2.1 Molecular Descriptors

It is understood that no molecule will be amenable to a single all-encompassing absolute definition, but any molecule can be characterized by a number of descriptors for each phenomenon of interest. When all these finite sets for the enormous variety of phenomena are put together, even after allowing for overlaps, the compilation of molecular descriptors can begin to approach an infinite set. Nevertheless, molecular descriptors lend themselves to compact classifications [16]. Sorted by the level of abstraction, descriptors fall in three classes:

- Macroscopic properties such as molecular weight or the octanol/water partition coefficient, refractive index, molar refractivity, parachor, density, solubility, partition coefficient, dipole moment, chemical shift, chromatographic retention time, spectroscopic signal (or even complete spectra), rate constant, equilibrium constant, vapor pressure, boiling/freezing point, and acid dissociation constant.
- Derived properties such as the surface distribution of electrostatic potential, the empirical absorbability index (a group-contribution index of carbon adsorption from aqueous solutions), or charge descriptors (calculated using quantum chemistry).
- More abstract measures such as BCUT, topological indices, sub-structural fingerprints, and feature counts [17-23].

Another pertinent classification is based on molecular dimensionality:

- One-dimensional (1D) descriptors depend only on the formula (e.g., molecular weight).
- 2D descriptors depend on topology—the connectivity of bonds between the atoms (e.g., the Balaban Connectivity Index).
- 3D descriptors depend on stereochemistry and geometry (e.g., dipole moment).
- 4D descriptors take into account the conformational variability (e.g., the global flexibility index).

The cited literature on molecular descriptors is highly evolved and amply details descriptor types, invariance, and degeneracy. Hence, these details need not be repeated here. It is instructive, however, to list a few considerations that pertain to data-driven predictions:

- Descriptors need not be complex to be useful; e.g., molecular weights or volumes can be quite predictive of transport properties such as diffusion coefficients.

- Descriptors are "good" if small tweaks of the descriptors cause small changes in the targeted behavior, and "poor" if the elicited responses are large or abrupt [24]. Prediction methods such as ANN are not incapable of handling such highly nonlinear dependences, but the necessary training will require large numbers of neurons and, concomitantly, large databases of target properties and descriptors.
- The ideal set of descriptors would be minimalist (offering sufficient representation with the fewest descriptors), would be fundamental instead of derivative, and would have little correlation among descriptors but a high correlation with the target property.
- The descriptor database should be entirely theoretical or computational in origin, in order to include virtual or uncharacterized molecules. That is, computed descriptors—i.e., purely theoretical constructs such as topographic indices or physical properties calculated using theoretical or empirical equations—are preferable to measured properties, since measured descriptors may not be available for all the molecules in the database.

While a comprehensive database may be set up and continually updated with theoretical indices as well as measured properties, the prediction method need not always use the entire database, but instead can use smaller, field-specific subsets of the database [18], with each search being restricted to a certain cluster of chemicals or class of descriptors culled via different chemical selection criteria.

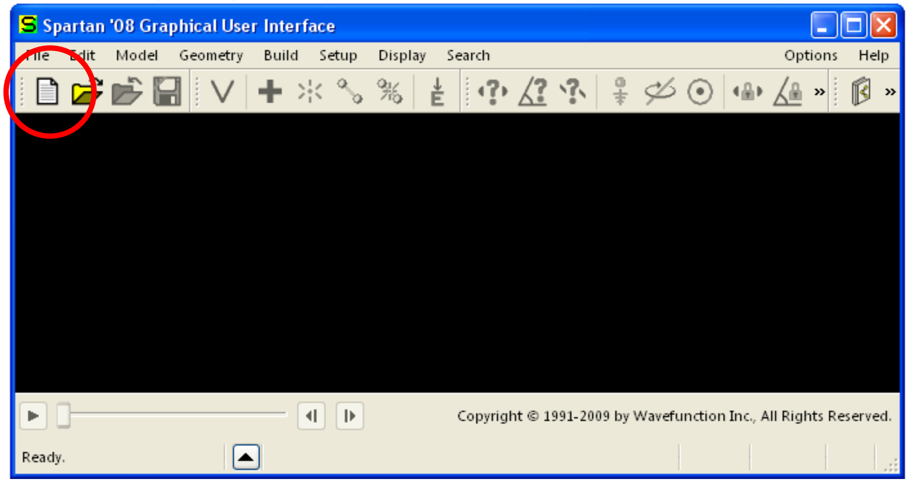
3.2.2 Machine-Readable Molecular Structures

For calculating molecular descriptors, it is not enough to know just the molecular formulas. The structures have to be rendered in a machine-readable format such as:

- Simplified Molecular Input Line Entry Specification (SMILES)
- SYBYL© Line Notation (SLN)
- Structure-Data File (SDF)

For most known chemicals, molecular structures can be found in public [25] or commercial [26] databases; format conversion (e.g., from SLN to SMILES) is also straightforward [27]. For envisioned, proprietary, or classified chemicals, however, such structures may be unavailable or inaccessible. This is the case with some of the CWAs, simulants, and IPFS chemicals. When machine-readable structures are inaccessible, the structures have to be *built*—using computational chemistry software such as Spartan® [28].

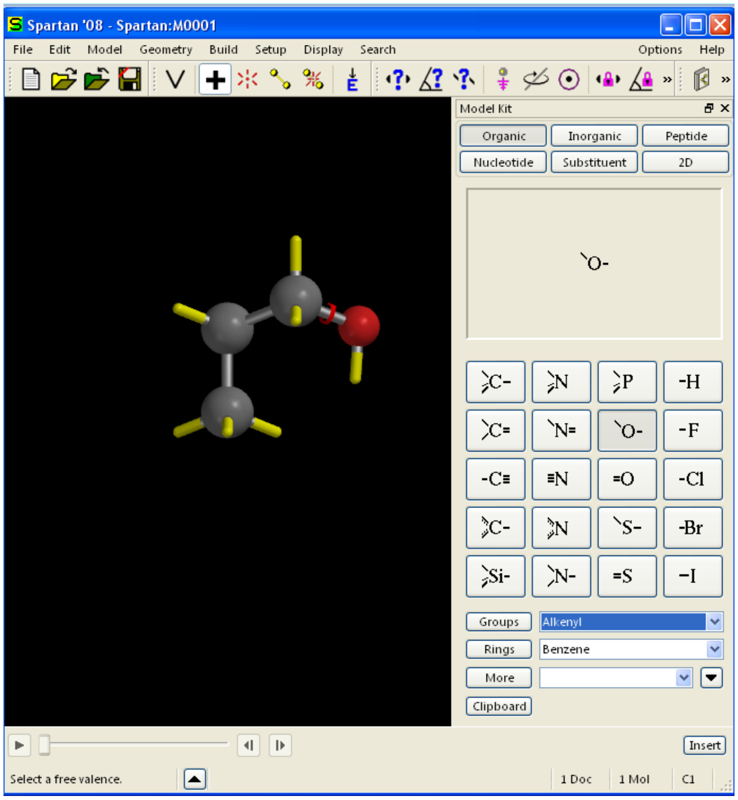
Spartan® is a versatile computational chemistry software tool that has many sophisticated capabilities. Here, however, only the simplest of these capabilities is invoked, namely Spartan®'s 3D molecular modeling tools to construct the molecules. It is easiest to build each molecule individually and then export the structures of a batch of molecules to the descriptor calculation software. The first step is to build each molecule. To do this, Spartan® is opened from the desktop (Figure 13). A new molecule can be constructed by clicking on the "New" button, circled in red. This will bring up the Model Kit sidebar (Figure 14), which allows the construction of the molecule atom by atom. In the case of more complex molecules, highly interconnected aromatics, for example, it may be easier to build the molecule using the "Groups" or "Rings" buttons, which allow the rapid incorporation of functional groups into the molecule.



Copyright © 1991-2009 by Wavefunction Inc., All Rights Reserved.

Ready.

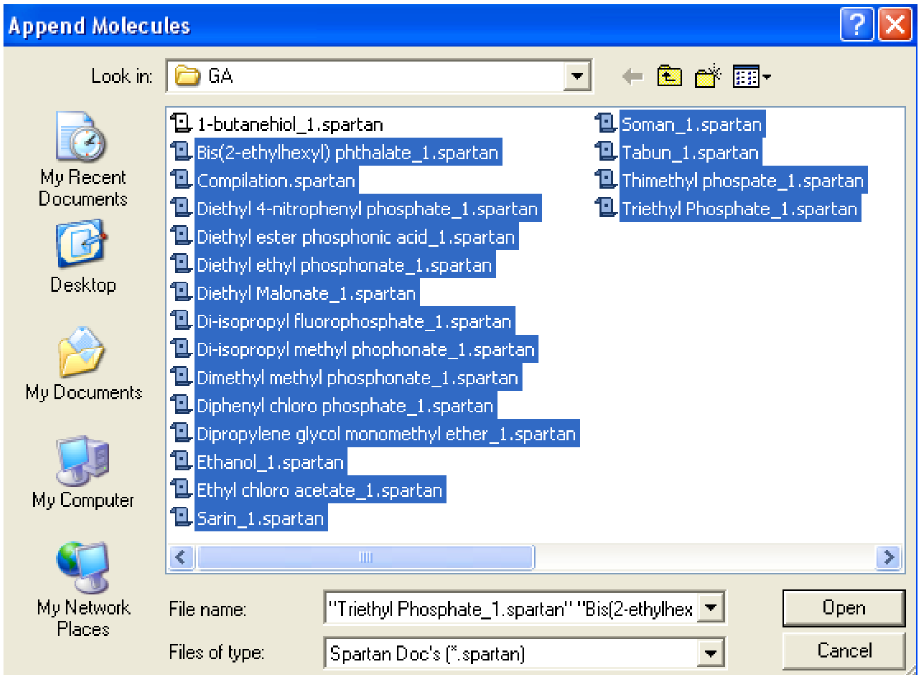
Figure 13 The Spartan GUI



Select a free valence.

Figure 14 Spartan Model Kit

Once the molecule has been completed, the structure is saved, typically with the name of the molecule, in SDF. This process is repeated for as many molecules as is desired, and all are incorporated into one compilation file which can be easily exported to the descriptor-calculation software, Sarchitect® [29]. To do this, the first molecule in the group is opened; returning to the toolbar at the top of the screen and clicking “File > Append Molecule(s)…” permits more molecules to be added. In the example in Figure 15, 1-Butanethiol is the molecule which is already open. The other molecules are being added to the file.



Append Molecules

Look in: GA

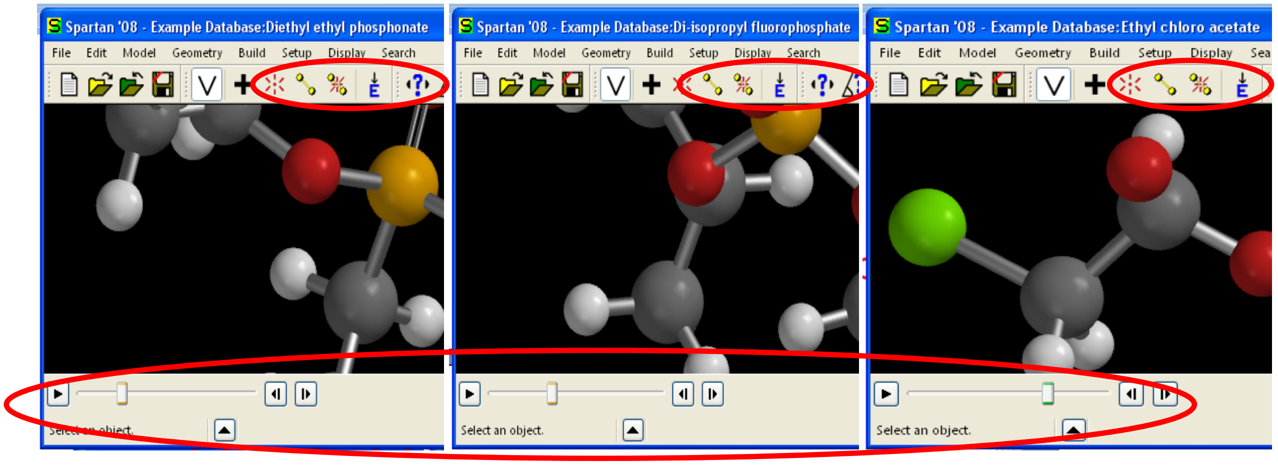
1-butanehiol_1.spartan
Bis(2-ethylhexyl) phthalate_1.spartan
Compilation.spartan
Diethyl 4-nitrophenyl phosphate_1.spartan
Diethyl ester phosphonic acid_1.spartan
Diethyl ethyl phosphonate_1.spartan
Diethyl Malonate_1.spartan
Di-isopropyl fluorophosphate_1.spartan
Di-isopropyl methyl phophonate_1.spartan
Dimethyl methyl phosphonate_1.spartan
Diphenyl chloro phosphate_1.spartan
Dipropylene glycol monomethyl ether_1.spartan
Ethanol_1.spartan
Ethyl chloro acetate_1.spartan
Sarin_1.spartan
Soman_1.spartan
Tabun_1.spartan
Thimethyl phospate_1.spartan
Triethyl Phosphate_1.spartan

File name: "Triethyl Phosphate_1.spartan" "Bis(2-ethylhex

Files of type: Spartan Doc's (*.spartan)

Figure 15 Spartan File Processing

Once this is complete, "Save As" yields a new (SDF) file, which is representative of the group ("GA Simulant Database", for example). The different molecules in the set can then be viewed by moving the slider or clicking the arrows at the bottom of the screen (Figure 16). The name at the top of the screen should change to match the molecule showing in the viewer.



Spartan '08 - Example Database:Diethyl ethyl phosphonate

Spartan '08 - Example Database:Di-isopropyl fluorophosphate

Spartan '08 - Example Database:Ethyl chloro acetate

Select an object.

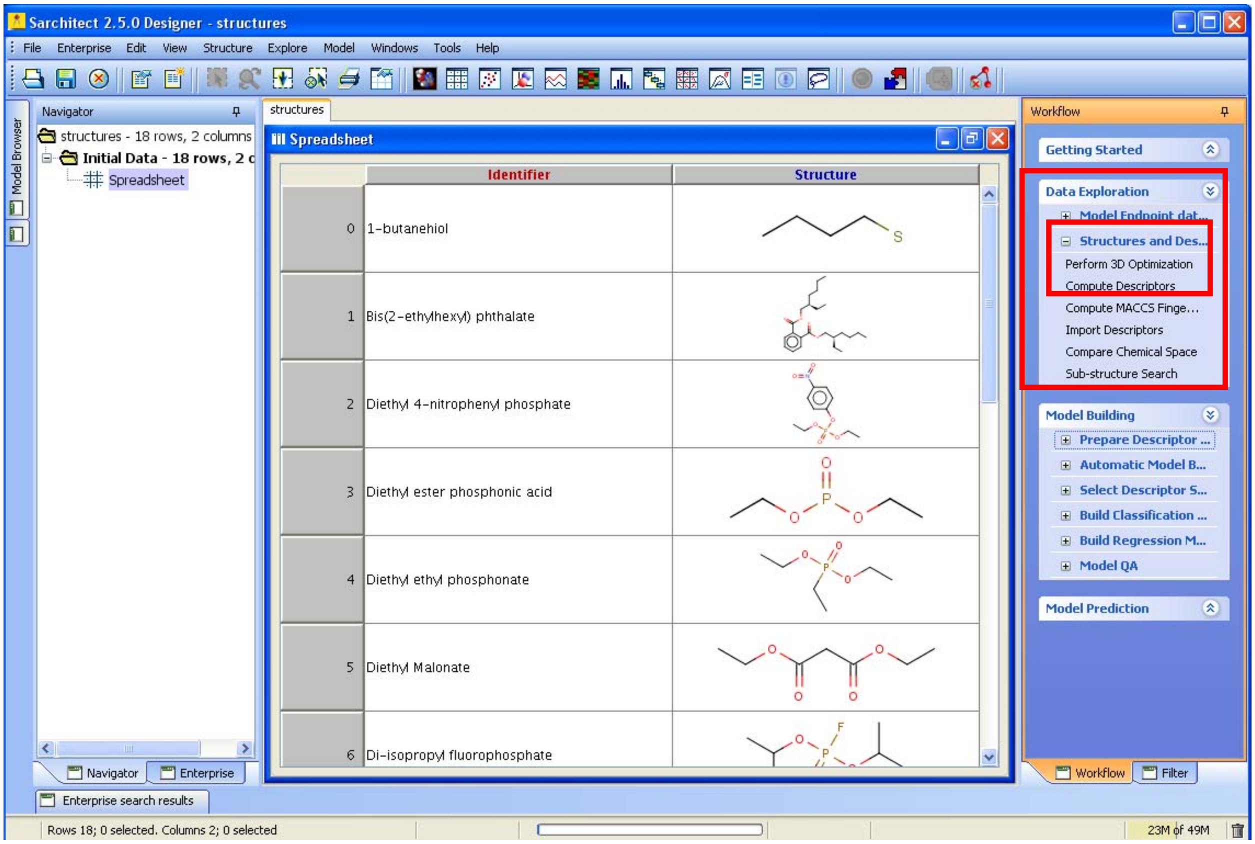
Figure 16 Spartan Examples

The database of molecules can then be imported into Sarchitect®.

3.2.3 Molecular Descriptors Calculation

The commercial Sarchitect® software was used in this work, although, as mentioned previously, descriptors can be calculated—with varying degrees of rigor and flexibility—using other computational chemistry software, commercial or free. Sarchitect Designer® is a commercial software platform for mining, modeling, and predicting drug-relevant properties of molecules.

In Sarchitect®, the user opens the SDF (or SMI) file (Figure 17) that was made in Spartan®, optimizes the structures, and calculates molecular descriptors.



	Identifier	Structure
0	1-butanehiol	
1	Bis(2-ethylhexyl) phthalate	
2	Diethyl 4-nitrophenyl phosphate	
3	Diethyl ester phosphonic acid	
4	Diethyl ethyl phosphonate	
5	Diethyl Malonate	
6	Di-isopropyl fluorophosphate	

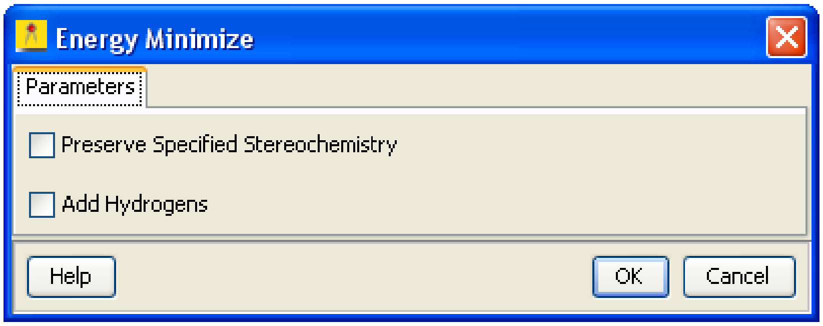
Rows 18; 0 selected. Columns 2; 0 selected

Figure 17 Sarchitect Initial Screen

Sarchitect® is opened from the desktop by the user. A dialog box opens, asking if the user would like to open an existing project, import structures from files, import a file, or browse existing models. The user selects "Import Structure(s) from Files" and chooses the SDF file made previously using Spartan®. (All options in the dialog box can be left at their default settings.) Alternatively, the user may choose an SMI file of structures in the SMILES format. Then, the user would see a spreadsheet similar to the one in Figure 17, in the main working area. The various side panels can be minimized using the little "pin" button, leaving only the "Navigator" and "Workflow" panels visible, as these are used most frequently. Every action which is performed using the Workflow panel is tracked in the Navigator panel such that at any time the user can look back at the database as it was in the previous step. All of the operations available in the Workflow panel on the right-hand side are also available from the top toolbar, but they are organized in a much more user-friendly manner. All the operations performed here are located in the Structures and Descriptors group under the Data Exploration tab.

With the molecules loaded, the user can optimize the structures so as to reach the proper bond lengths, bond angles, and stereo-chemical orientations and the lowest energy configuration. Sarchitect® calculates descriptors only on optimized structures. So the user has to click the "Perform 3D Optimization" button. (Structure optimization could also be done in Spartan®, in which case the optimization in Sarchitect® would simply converge more quickly.)

The dialog box shown in Figure 18 opens, and, barring any specific stereochemistry which needs to be maintained, the user should be able to click OK without changing any of the default settings. As can be seen in Figure 19, another column is added, containing 3D structures akin to the ones made in Spartan®. These structures, however, are optimal. Descriptors can then be calculated.



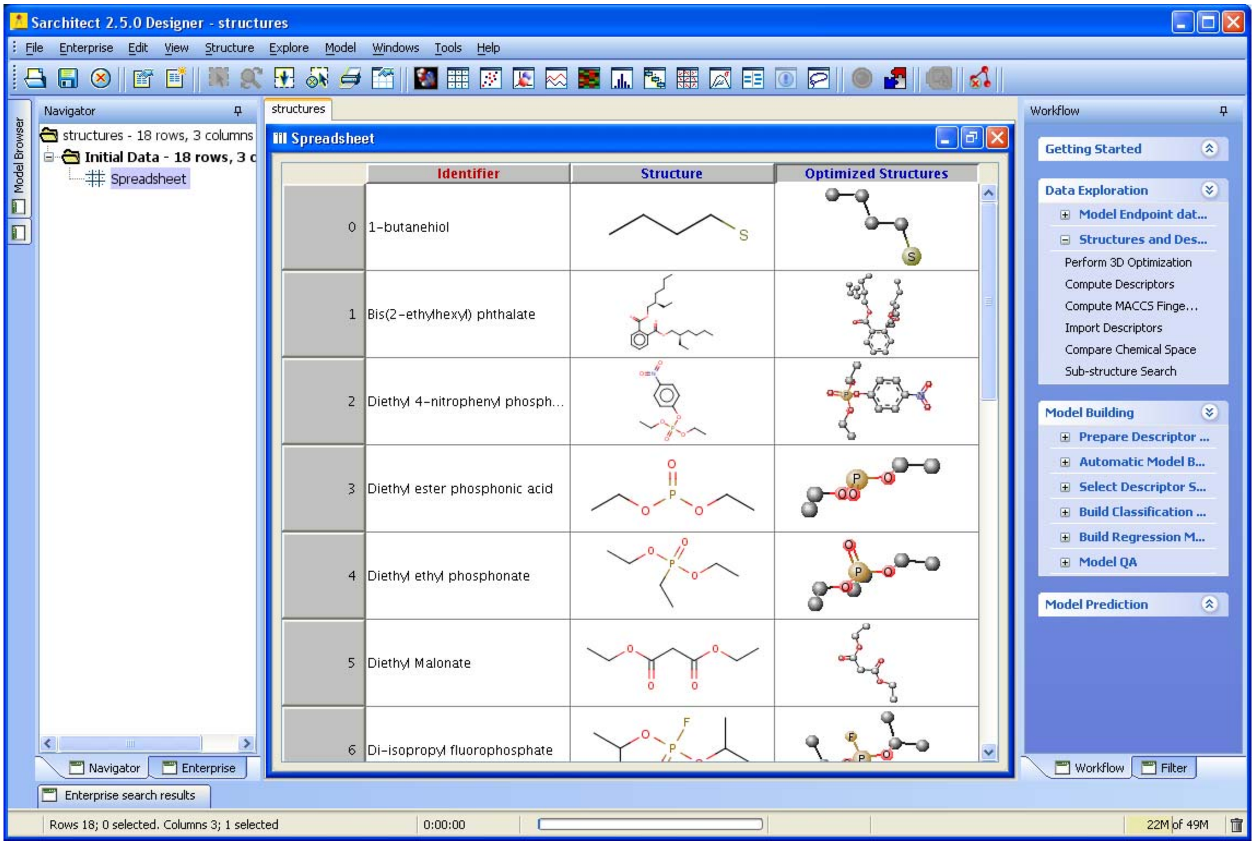
Energy Minimize

Parameters

☐ Preserve Specified Stereochemistry

☐ Add Hydrogens

Figure 18 Energy Optimization



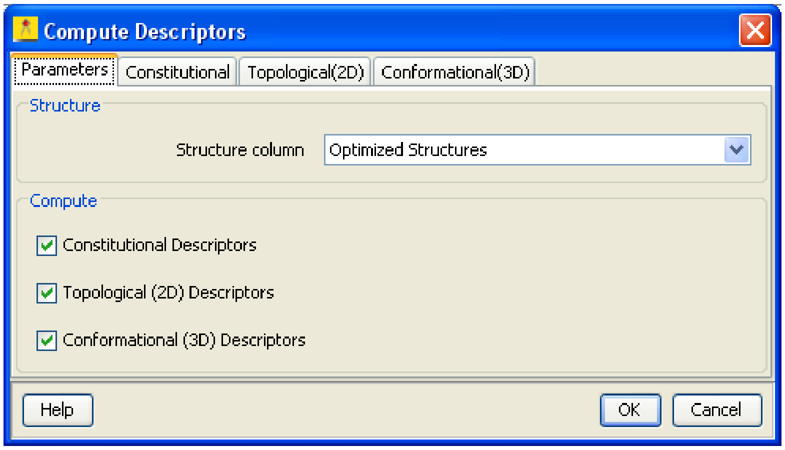
Sarchitect 2.5.0 Designer - structures

	Identifier	Structure	Optimized Structures
0	1-butanehiol		
1	Bis(2-ethylhexyl) phthalate		
2	Diethyl 4-nitrophenyl phosph...		
3	Diethyl ester phosphonic acid		
4	Diethyl ethyl phosphonate		
5	Diethyl Malonate		
6	Di-isopropyl fluorophosphate		

Rows 18; 0 selected. Columns 3; 1 selected   0:00:00   22M of 49M

Figure 19 After Energy Optimization

There are three different categories of molecular descriptors calculated by Sarchitect®: constitutional, topological, and conformational. These are useful for characterizing molecules in different ways. In most cases, it is appropriate to calculate all of the available descriptors and allow the prediction methods to be used in Matlab® to select and combine those descriptors as appropriate. To proceed, the user returns to the Workflow panel on the right-hand side and selects “Compute Descriptors”, directly underneath the “Perform 3D Optimization” button. Unless a subset of descriptors is being omitted, all groups should be checked (Figure 20); hitting the OK button results in the calculation of 1084 descriptors (Figure 21). A similar procedure can be used for Molecular Access System (MACCS) descriptors. In total, Sarchitect® yields 1250 descriptors for each molecule.



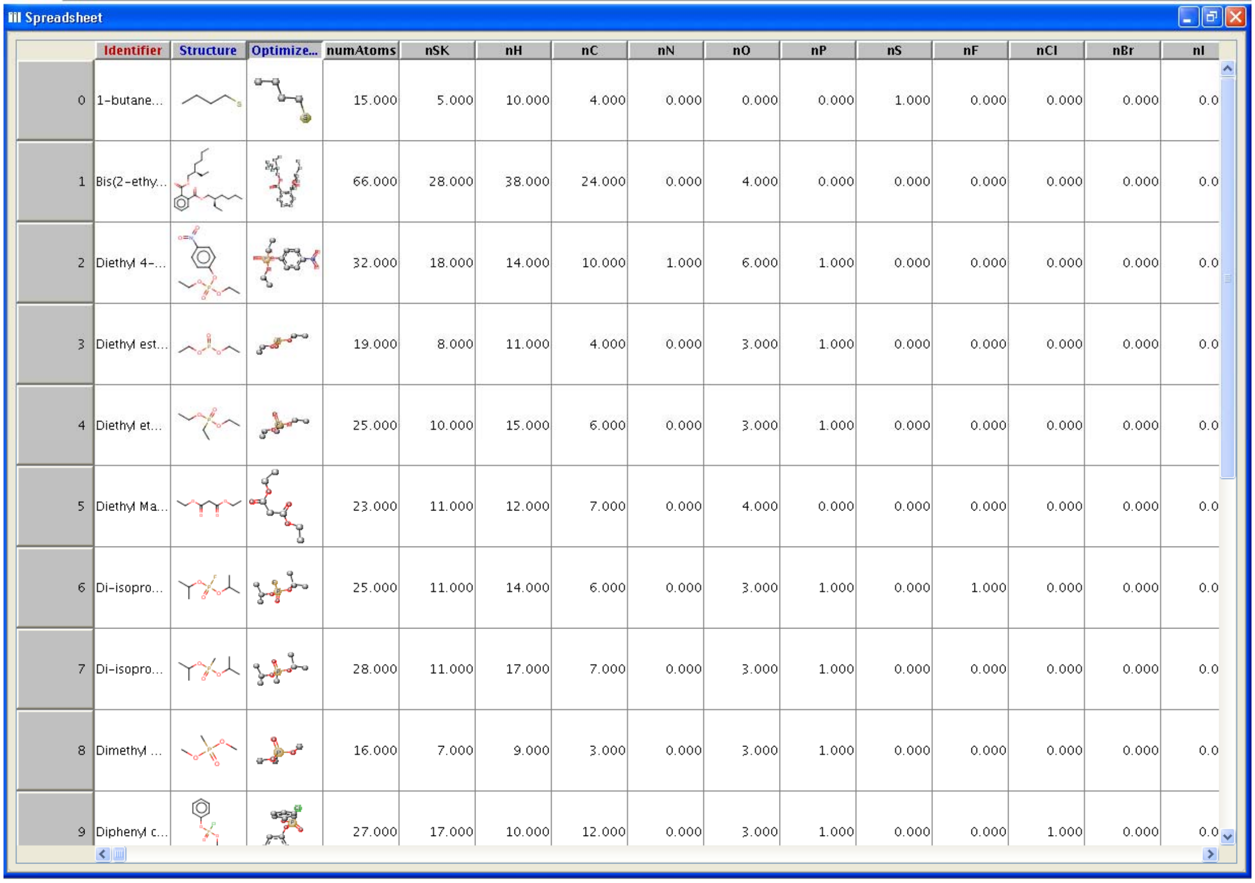
Structure column: Optimized Structures

☑ Constitutional Descriptors

☑ Topological (2D) Descriptors

☑ Conformational (3D) Descriptors

Figure 20 Descriptor Calculation



	Identifier	Structure	Optimize...	numAtoms	nSK	nH	nC	nN	nO	nP	nS	nF	nCl	nBr	nI
0	1-butane...			15.000	5.000	10.000	4.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.0
1	Bis(2-ethy...			66.000	28.000	38.000	24.000	0.000	4.000	0.000	0.000	0.000	0.000	0.000	0.0
2	Diethyl 4-...			32.000	18.000	14.000	10.000	1.000	6.000	1.000	0.000	0.000	0.000	0.000	0.0
3	Diethyl est...			19.000	8.000	11.000	4.000	0.000	3.000	1.000	0.000	0.000	0.000	0.000	0.0
4	Diethyl et...			25.000	10.000	15.000	6.000	0.000	3.000	1.000	0.000	0.000	0.000	0.000	0.0
5	Diethyl Ma...			23.000	11.000	12.000	7.000	0.000	4.000	0.000	0.000	0.000	0.000	0.000	0.0
6	Di-isopro...			25.000	11.000	14.000	6.000	0.000	3.000	1.000	0.000	1.000	0.000	0.000	0.0
7	Di-isopro...			28.000	11.000	17.000	7.000	0.000	3.000	1.000	0.000	0.000	0.000	0.000	0.0
8	Dimethyl ...			16.000	7.000	9.000	3.000	0.000	3.000	1.000	0.000	0.000	0.000	0.000	0.0
9	Diphenyl c...			27.000	17.000	10.000	12.000	0.000	3.000	1.000	0.000	0.000	1.000	0.000	0.0

Figure 21 Page of Descriptors

The descriptors range in complexity from the very simple (number of atoms in the molecule) to the complex and obscure (e.g., GETAWAY "Geometry, Topology, and Atom-Weights Assembly" descriptors which are based upon the Molecular Influence Matrix).

Even though Sarchitect® can perform many types of statistical prediction (not just calculation of descriptors), the algorithms are set and not easily customized. Accordingly, it is best to hand off the descriptor database at this point to Matlab®, in which the user can develop custom codes to implement the latest developments from the literature.

To save the data for export, the user returns to the top toolbar and selects "File > Export to File > Data". This brings up a "Save As" dialog box, permitting the results to be saved as a comma separated file (*.csv), which can then be read by a Matlab® code to sort the descriptors and save the different types in different sheets of an Excel workbook. In turn, the Excel sheets are read by another Matlab® code for making predictions.

3.2.4 Formatting the Descriptor Database

After transferring the *.csv file from the previous step to the directory in which the Matlab® code read_desc resides, it is straightforward to run the code and generate the *.xlsx file which, in turn, is read by the prediction codes described in Chapter 4. It is important to note that the *.xlsx workbook thus created will have only the descriptors. It is necessary for the user to copy and paste the column(s) of target properties (from the output Excel files described in Chapter 2) in the sheet named "Targets" in this work book. The prediction code seeks only one target property, specifically the one in Column 2 (and ignores the properties in the subsequent columns). Also, optionally, the user can copy and paste the structures in SMI format from the *.csv file into the sheet named "Structures" in the *.xlsx file.

3.3 RESULTS AND DISCUSSION

In summary, using commercial computational chemistry software (Spartan®), machine-readable molecular structures in SDF format were built for the 23 organic solvents ("data chemicals"), as well as about 60 threat agents and simulants ("query chemicals"). Using another commercial computational chemistry software (Sarchitect®), these molecular structures were read in SDF format (and saved in SMILES format); about 1,000 constitutional, topological, and conformational properties ("molecular descriptors") were calculated for all the data and query chemicals. Finally, the resulting descriptor database was formatted further (to be the input for the analysis described in Chapter 4) by a Matlab® code developed by NSRDEC.

CHAPTER 4 PREDICTIONS

4.1 BACKGROUND

This chapter details the process for developing the methodology for training the data-driven prediction method—Artificial Neural Networks (ANN)—using the target properties (diffusion coefficients) detailed in Chapter 2 as dependent variables, the descriptors of the data chemicals detailed in Chapter 3 as independent variables, and the molecular descriptors of the query chemicals (also detailed in Chapter 3) as inputs. It then presents and discusses the results of using the trained ANN to predict the target properties of the query chemicals.

At the beginning of the project, other data-driven methods (e.g., Fuzzy Logic and Generalized Linear Regression) were also investigated, but eventually it was decided to focus exclusively on ANN, the most powerful of the lot. For example, a neural network reduces to linear regression when the number of "hidden neurons" (or nonlinear terms) is set to zero.

4.2 METHODS DEVELOPMENT

4.2.1 Neural Networks

Neural networks are highly developed in computer science and statistics, with applications in image processing, data mining, quantitative finance, algorithmic stock trading [30-35], and QSAR/QSPR [36-40].

ANN is essentially regression, but since the nonlinear chain of fitting equations stays hidden, it is more "black box" than regression with an explicit linear or nonlinear model. The terms of ANN hence can be clarified by comparison with the familiar terms of standard regression (Table 8).

Table 8 Comparison of Regression and Neural Networks

LINEAR/NONLINEAR REGRESSION	NEURAL NETWORK
Dependent variable	Target property
Independent variables	Descriptors
Fitting, to evaluate the regression coefficients	Training
Applying the regression model	Predicting
Coefficients	Weights
Regularization (to avoid overfitting)	Bayesian network
Intercept	Bias

In the literature, this method is interchangeably denoted as neural networks, ANN, or simply, the network. Instructional schematics are presented in Appendix B which touch on key neural network concepts (i.e., hidden layers, individual neurons, layers of neurons, and linear, log-sig, and tan-sig transfer functions) and the need for multiple network initializations, etc. Expanding on such topics is beyond the scope of this report; more information about neural networks can be found in Refs. 30-40.

4.2.2 Descriptor Sorting

Using all the descriptors *en bloc* in the calculations is tantamount to losing valuable information, namely the qualitative differences between different types of descriptors. Accordingly, as noted in Chapter 3, the descriptors are sorted into constitutional, topological, conformational, and MACCS types and are stored in different sheets of an Excel workbook that form the input to the prediction code. In turn, the Excel sheets sort further within each type: binary (0 or 1) or analog (not confined to 0 or 1). For predicting permeation, the analog constitutional descriptors were used. Other targets may need other descriptor types; the codes provide various options.

4.2.3 Descriptor Pruning

While it is convenient that computational chemistry readily yields hundreds of descriptors, ironically the prediction algorithm cannot and need not use this plethora in full. The folly of attempting to do so can be understood by recalling that the descriptors serve as independent variables, with each one associated with at least one parameter (coefficient or weight) in the fitting equation or network. For all these parameters to be meaningful, it is desirable that the number of rows of target data points equals or exceeds the number of columns of descriptors. Not meeting this requirement can be a problem—namely, an under-determined system with too few degrees of freedom when the target database has only a limited number of rows (e.g., 20, as in the present case). Also, a large matrix of descriptors will be computationally intensive to work with.

The abundance of descriptors is thus a problem, a well-known one at that, called “the curse of dimensionality” [41]. The remedy of “dimensionality reduction” is likewise well-known and continues to be an evolving research area with no single panacea, but many useful approaches. The premise of the remedies is that not all the numerous descriptors are independent of (or more precisely, orthogonal to) one another, vary significantly across the rows of chemicals, nor are relevant to every target property. The prediction code developed in this project incorporates several formal and informal dimensionality-reduction techniques, using built-in codes in the Matlab® statistical toolbox™ or Matlab® freeware suites:

- Principal components [42, 43]: These are linear combinations of the descriptors arrived at after some statistical matrix operations on the original descriptors. There are as many columns of principal components as there are columns of descriptors, but only the first few columns (ranked in decreasing order of variance across rows) are important. That is, along these few principal-component directions in the hyperspace of descriptors, the instances (in the present case, the chemicals) show the most variation from one another. These few principal components are retained and can be used in place of the original descriptors in training the network. As might have been noticed, principal components have a curious flaw in that they are arrived at based only on the descriptor matrix, without any regard to the target data. Thus, it is possible to have principal components that display a great variance across the rows of chemicals, but have little correlation with the target!
- Probabilistic principal components [44]: These are the same as the principal components above, but are calculated in an indirect way that avoids the inversion of large matrices. This will be useful when dealing with large or incomplete databases.
- Nonlinear principal components [45, 46]: Since principal components are linear combinations of the raw descriptors, their use—while mitigating the problem of high dimensionality—can introduce another problem; namely, confounding of the target-to-descriptor relations, when the relationships are nonlinear. Using nonlinear principal components instead is better in principle. In practice, however, their use in prediction is computationally intensive: first the hundreds of raw descriptors are condensed using an

auto-associative network [47] into a few nonlinear principal components that are then used as the dependent variables in another network to make the actual predictions.

- Internal dimensionality reduction: As a remedy to the problem of computational intensity associated with nonlinear principal components, an alternative approach was developed: It involves two hidden layers instead of the usual single layer: the first layer does the dimensionality reduction, while the second layer does the prediction. This novel option is fully functional, and its utility is being explored.
- A "filter" method to keep only the descriptors that are strongly correlated with the target, but are only weakly correlated with one another: A filter is in effect a one-off up-front pruning. There are also "wrapper" techniques that make predictions using descriptors filtered with one set of criteria and renew the predictions after altering the filter criteria in an iterative fashion, until the best possible predictions are found. In the parlance of computer science, "wrappers utilize the learning machine of interest as a black box to score subsets of variable according to their predictive power; filters select subsets of variables as a pre-processing step, independently of the chosen predictor" [48]. The present code does not include wrappers (which were explored initially), but has a filter.
- Expert choice: This involves choosing a few descriptors on a physicochemical basis as the ones most likely to determine the target property. The advantage of this method, besides the savings in computer time, is that it does not confound any nonlinear relationships that may exist between the target property and individual descriptors. The disadvantage is the need for experts to choose the descriptors! A wrapper method can be developed that circumvents the need for such expertise—by arriving at the best set of descriptors automatically, by iteration, e.g., using a genetic algorithm [49].

4.2.4 Avoiding Overfitting

A peril of all regression—least squares as well as neural networks— is "overfitting". A method may make excellent predictions for the chemicals on which it was trained, but poor ones for others. Overfitting is especially acute when the number of instances (data chemicals) is small. The traditional remedy is to divide the data chemicals into training and validation sets and begin training the network using only the training set while monitoring the prediction-versus-data error for both training and validation sets. The error for the training set will keep on dropping; the error for the validation set will also drop initially, but will begin to rise eventually. The training stops at this point. This "early stopping" is fine for large datasets, but not for datasets that are too small to begin with and hence not amenable to further division.

Here, a better alternative was chosen: Bayesian regularization ANN (BRANN) [50-52] that uses the full training dataset but avoids overfitting by the use of a penalty function, just as "ridge regression" does for standard least-squares. This alternative works well for datasets both small and large. The advantages of BRANNs as elaborated in Ref. 50, verbatim, are:

- They obviate the validation procedure of normal regression methods and automatically address many needs of QSAR: choice of model (i.e., network layout), robustness, validation, and optimization of the layout.
- They are difficult to over-train by virtue of a built-in criterion for stopping the training.
- They are difficult to overfit because they converge to an effective number of parameters, even if the user specifies a starting layout that has an excessive number of hidden neurons. (In the converged optimal network, multiple columns of descriptors end up sharing the same parameters/weights, thus making do with fewer effective parameters.)

- They are inherently insensitive to how elaborate the network layout is, as long as a minimal architecture has been provided.
- Unlike ordinary neural networks, they need neither a validation set nor a test set, since they produce the best possible model most consistent with the data. As a result, all the available data can be dedicated to model building (as opposed to setting aside a substantial number of rows for validation and testing as with the usual networks)—which is clearly a boon where data are scarce and expensive to acquire.

In total, it is doubly fortunate that a switch to BRANN was made and that the Matlab® Neural Networks Toolbox™ has built-in routines that implement BRANN.

4.2.5 Tailored Training

Instead of using all the available data chemicals to train the neural network, the code has robust options for a "tailored training" approach, using only a subset of data chemicals that is most similar to the query chemical, and, for a controlled comparison, using another subset of the same size but with randomly chosen chemicals. The similarity ranking can be done using a variety of distance measures (default: Euclidean distance). This powerful option—while quite involved in terms of coding strategy—has not been of much use with the present system (since the target database is small to begin with), but may be useful when larger databases become available.

4.2.6 Cross-Validation

From the discussion so far, it should be clear that, using a network that is trained using the target data for the data chemicals (i.e., the 23 solvents for which log D data are available), predictions will be made for the query chemicals (i.e., the 53 CWA and IPFS chemicals for which there are no data and predictions are sought). The ultimate validation of the predictions awaits the availability of experimental data for the query chemicals, but, in the meantime, a metric is needed to judge the networks. Actually, such a metric is automatically generated in the process of making predictions for the *data* chemicals. That is, the data chemicals are also first treated as query chemicals: the information for one data/query chemical is excluded, and the network is trained with the remaining 22 data chemicals. The metric consists in how well the prediction compares with the data for the set-aside data chemical. The process is repeated to cover all the data chemicals, one at a time.

This is known as leave-one-out "n-fold cross-validation" [53], with "n" being the total number of data chemicals: here, 23. That is, set aside one of the data chemicals, train the network using the remaining 22 data chemicals, use the trained network to make a prediction for that one chemical which was left out, and repeat this 23 times, leaving one chemical out at a time. For each of the 23 training sets, in turn, the network is trained repeatedly (e.g., multiple initializations and multiple variations for the number of neurons in the hidden layer). Such repetition is necessary to avoid the acceptance of a sub-optimal network. The network resulting from each repetition is used to make predictions for the set-aside chemical. The best network is one that maximizes a merit measure, e.g., the ratio of the square of the training correlation coefficient (between data and predictions for the 22 chemicals included in the training) and the residual (i.e., difference between data and prediction using the trained network) for the set-aside data chemical.

Cross-validation is an established statistical technique, and it offers a rigorous and conservative internal test of the model's predictive capability. It is worth repeating that cross-validation is a much more severe test of a model than the all too common practice of fitting all the training data and omitting any validation, internal or external. So, if the cross-validation predictions for the data chemicals are reasonable, then the predictions (now using all the training data) for the query chemicals would probably be acceptable.

The code also permits a visual judgment of the prediction method by a straightforward comparison with the target data; the predictions that fall within certain bounds (e.g., target data ± 5%) can be accepted as correct. That may seem like a high tolerance (± 5% in log D translates to an order-of-magnitude variation in D), but it is comparable with the judgment criterion of the solubility challenge [54, 55]: ± 0.5 log units.

4.2.7 Code Probing

As the prediction codes evolved into a rather complex and convoluted software package, it became of interest to pressure-test the coding and simultaneously improve the predictions.
The prediction codes were first probed with simulated test "data," with *a priori* knowledge of the exact dependence of the target property on the molecular descriptors. Specifically, probes were conducted first with artificial, meaningless data (both noise free and with noise) and then with artificial, meaningful data that were noise free. This probing yielded several confirmations and insights.

The first step in the probe was to calculate the "data" using an equation of no physical significance:

$$D = \sum_{i=1}^{3} v_i^2 \, . \qquad (75)$$

$v_1 =$ Number of rotatable bonds,

$v_2 =$ Molecular weight, and

$v_3 =$ Log (octanol/water partition coefficient).

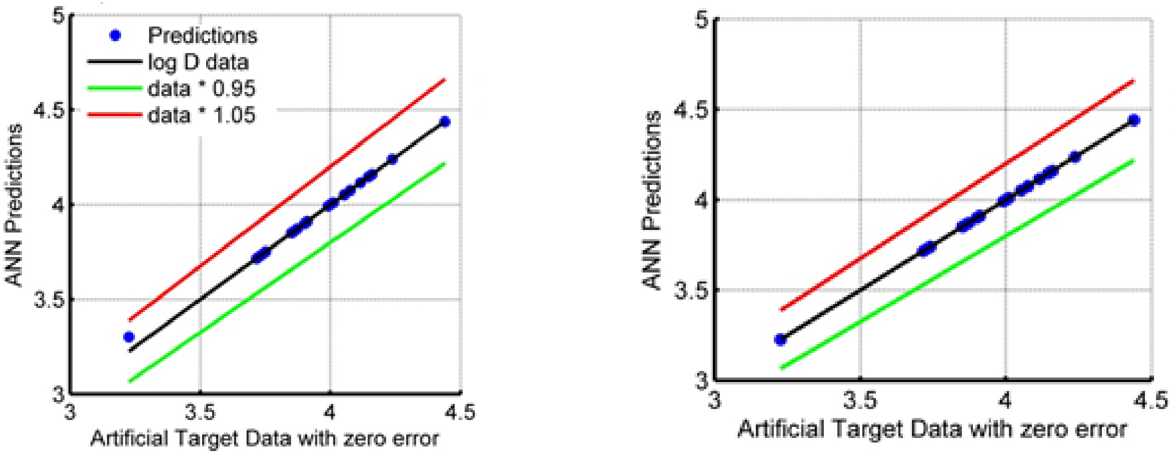
The calculated target data were then fed to a neural network—with and without added random noise—and trained with selected descriptors, specifically the same three descriptors that were used in Eq. 75 to calculate the targets. If the method is logical and the coding is correct, the cross-validation predictions from the network should be error free.

The results are presented in Figures 22 to 26. Each of those figures has two parts: a summary plot and a leave-one-out example:

- Each point in the summary plot is the result of setting aside the datum for that chemical, training the neural network with the rest of the data chemicals (usually a total of 22, i.e., all the available data except the datum for the set-aside chemical), and using the trained network to make a prediction for the set-aside chemical. The plot condenses the results from 23 separate neural networks (each of which, in turn, is the best among several initializations and variations of the numbers of neurons in the hidden layer.)
- The leave-one-out plot is an example (usually for chemical #4) of the predictions in the summary plot. That is, *each* prediction point in the summary plot is the result of a leave-one-out prediction like the one displayed in the second plot in the pair.

Figure 22 shows the results from the probe using artificial, meaningless, noise-free data, and Figure 23 shows the results from the probe using artificial, meaningless data with noise. The other details used in each of those probes were:

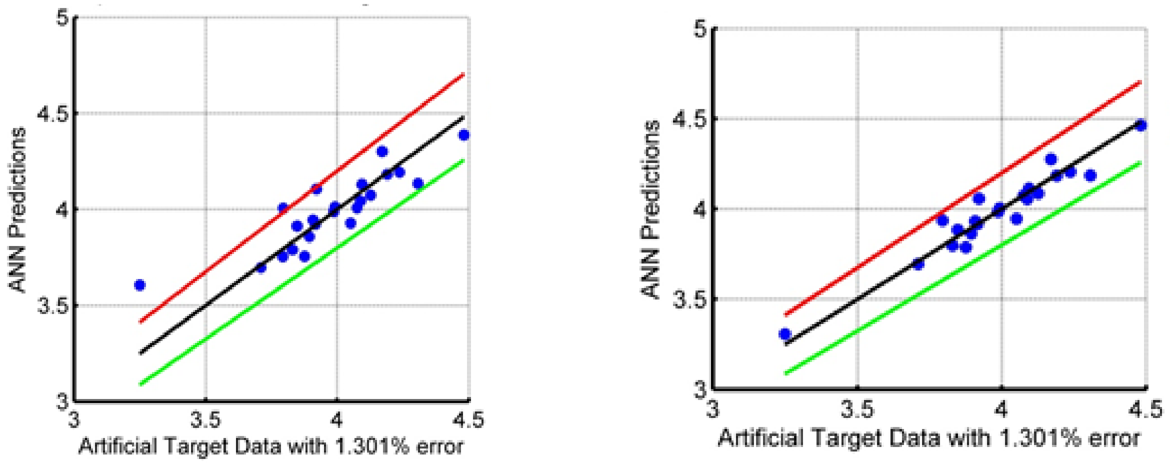
- "Data" calculated using an ad hoc equation
- Predictions based on three expert-chosen raw descriptors (analog constitutional)
- Training with all 22 non-query data chemicals; prediction for each set-aside chemical



(a) Summary of cross-validation predictions
100% of predictions are ±5% of targets; R^2 = 0.997

(b) Basis of a single point in the summary
Chemical # 4; Prediction error = 0%
100% of predictions are ±5% of targets; R^2 = 1.0
Target Variance 0.025; Skew -0.964; Kurtosis 5.43
Best case: 10 hidden neurons; Initialization # 3

Figure 22 Probe Using Artificial, Meaningless, Noise-Free Data–
Fit Based on Three Expert-Chosen Raw Descriptors: (a) Summary, (b) Basis



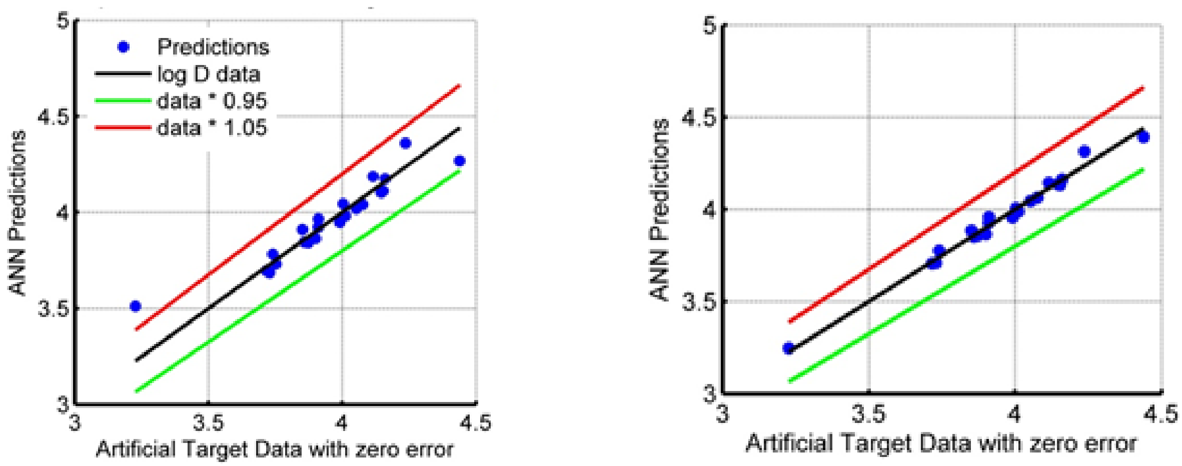
(a) Summary of cross-validation predictions
91% of predictions are ±5% of targets; R^2 = 0.769

(b) Basis of a single point in the summary
Chemical # 4; Prediction error = -1%
100% of predictions are ±5% of targets; R^2 = 0.926
Target Variance 0.026; Skew -0.835; Kurtosis 5.19
Best case: 10 hidden neurons; Initialization # 3

Figure 23 Probe Using Artificial, Meaningless, Noisy Data–
Fit Based on Three Expert-Chosen Raw Descriptors: (a) Summary, (b) Basis

Indeed, the prediction was perfect when the data were noise-free (Figure 22), but got worse as the data got noisier (Figure 23). From these two cases and others (not shown) with higher noise levels, a good correlation was observed between training/prediction error and data noise; the flip-side of this point is that, when the data were noisy, the neural network was unable to filter out the noise and capture the true data. If the network were able to do this, the underlying noise-free data would have been recovered even when the network was presented with noisy data. So, the network, somewhat like a spline fit, seemed to be tracking the errors as part of the data pattern. This is a troubling indicator of the dreaded "overfitting" which the BRANN is claimed to be immune to. The noise levels in these tests (20% to 50% in D), however, were much higher than what would be encountered in practice.

Then the BRANN was trained using principal components (which are linear combinations of many more descriptors than the few that were used in calculating the data), and a second probe using the same artificial, meaningless data was conducted. The fitting in this probe was based on five linear principal components and began with noise-free data, which were then made progressively noisier; all the other details were the same as those used in the first probe (Figures 22 and 23). The predictions were quite good (Figure 24), but not perfect even when the data were noise-free. The difficulty was compounded as the data got noisier. Also, principal components are *linear* combinations of descriptors and hence probably have trouble capturing these "data", which are based on the squares of the descriptors. (This point was probed by squaring all the descriptors before calculating the principal components. Squaring the descriptors did not help even though the target "data" were based on the squares of certain descriptors, since the principal components end up mixing in the squares of many other descriptors that were not connected with the targets.) In sum, when the target-descriptors relation is nonlinear, raw descriptors are preferable to linear principal components.



(a) Summary of cross-validation predictions
96% of predictions are ±5% of targets; $R^2 = 0.886$

(b) Basis of a single point in the summary
Chemical # 4; Prediction error = 0%
100% of predictions are ±5% of targets; $R^2 = 0.984$
Target Variance 0.025; Skew -0.964; Kurtosis 5.43
Best case: 10 hidden neurons; Initialization # 2

Figure 24 Second Probe with Artificial, Meaningless, Noise-Free Data– Fit Based on Five Principal Components: (a) Summary, (b) Basis

Next a second test using artificial data was conducted, but the "data" were calculated using a physically more meaningful equation−the logarithm of the octanol/water partition coefficient:

$$\begin{aligned}\mathrm{MlogP} = &-1.041 + 1.244(\mathrm{CX})^{0.6} - 1.017(\mathrm{NO})^{0.9} + 0.406(\mathrm{PRX}) - 0.145(\mathrm{UB})^{0.8} + \\ &0.511(\mathrm{HB}) + 0.268(\mathrm{POL}) - 2.215(\mathrm{AMP}) + 0.912(\mathrm{ALK}) -0.392(\mathrm{RNG}) -3.684(\mathrm{QN}) + \\ &0.474(\mathrm{NO2}) + 1.582(\mathrm{NCS}) + 0.773(\mathrm{BLM})\end{aligned} \tag{76}$$

It was calculated in Sarchitect® (the computational chemistry software used in Chapter 3 to calculate molecular descriptors) using the Moriguchi equation [Sarchitect documentation and 56]. The model variables (which are also descriptors calculated in Sarchitect®) are number counts or frequencies (denoted by N) or presence/absence (denoted by B for binary: 1 or 0) of some molecular features. Their descriptions are listed here for completeness (even though they are just parameters in the present exercise):

CX N Summation of weighted numbers of carbon and halogen atoms; the weights are: 0.5 for F, 1.0 for C and Cl, 1.5 for Br, and 2.0 for I.

NO N Total number of Ns and Os.

PRX N Proximity effect of N/O: 2 for X-Y and 1 for X-A-Y (X, Y: N and/or O; A: C, S, or P; -: saturated or unsaturated bond) with a correction (-1) for -CON< and -SO2N<

UB N Number of unsaturated bonds including semi-polar bonds such as *N*-oxides and sulfoxides, except those in NO2.

HB B Binary variable for the presence of intramolecular hydrogen bond as *ortho*-OH and -CO-R, -OH and -NH2, -NH2 and -COOH, or 8-OH/NH2 in quinolines, 5 or 8-OH/NH2 in quinoxalines, *etc*.

POL N Number of aromatic polar substituents (aromatic substituents excluding Ar-C(X)(Y)- and Ar-C(X)=C; X, Y: C and/or H). Upper limit = 4.

AMP N Amphoteric property; a-aminoacid = 1, aminobenzoic acid = 0.5, pyridinecarboxylic acid = 0.5.

ALK B Binary variable for alkane, alkene, cycloalkane, cycloalkene (hydrocarbons with 0 or 1 double bond) or hydrocarbon chain with at least 7 carbon atoms.

RNG B Binary variable for the presence of ring structures except benzene and its condensed rings (aromatic, heteroaromatic, and hydrocarbon rings).

QN N Quaternary nitrogen >N+<: 1; N-oxide: 0.5.

NO2 N Number of nitro groups.

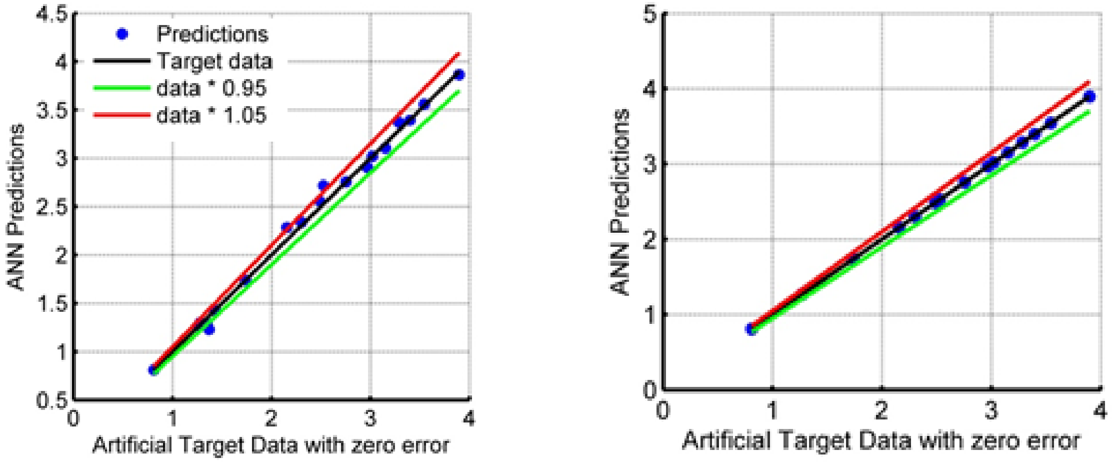
NCS N Isothiocyanate (-N=C=S): 1.0; thiocyanate (-S-C#N): 0.5.

BLM B Binary variable for the presence of ß-lactam.

The network was trained with the same 13 descriptors that were used in Eq. 76 to calculate the target's octanol/water partition coefficients. The conditions used in the probe were:

- "Data" calculated using an ad hoc equation using 13 descriptors
- Predictions based on the same 13 raw descriptors
- Training with all 21 non-query data chemicals; prediction for each set-aside chemical

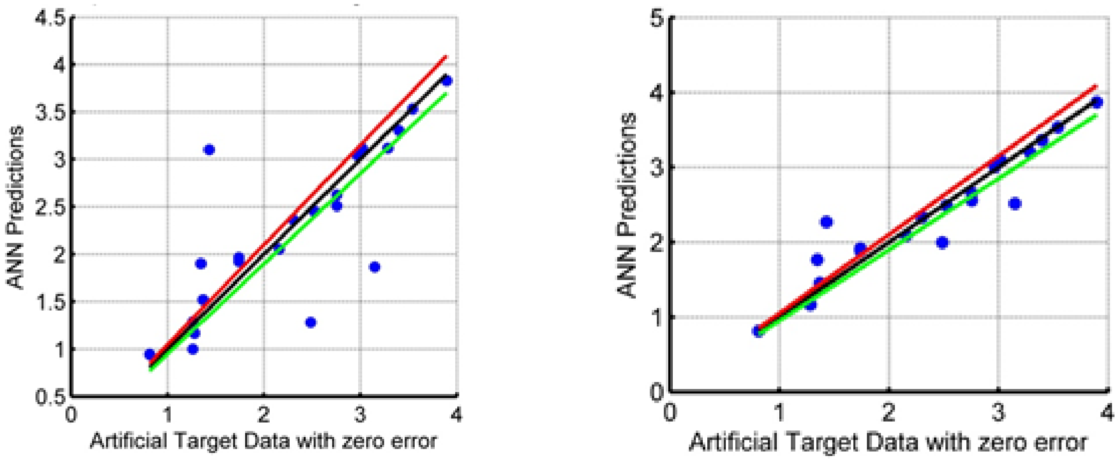
The predictions were excellent (Figure 25). Then a second probe using the same artificial, meaningful, noise-free, data was conducted with five linear principal components instead of the raw descriptors, and the predictions were poor (Figure 26). (Chemical #1, acetylene, was too much of an outlier to be included in this test.) Again, it appears that when the target depends on the descriptors nonlinearly using raw descriptors is preferable to using linear principal components.



(a) Cross-validation summary
86% predictions are ±5% of targets; R^2 = 0.995

(b) Basis of a single point in the summary
Chemical # 4; Prediction error = 0%
100% of predictions are ±5% of targets; R^2 = 1.0
Target Variance 0.054; Skew -0.036; Kurtosis 1.8
Best case: 10 hidden neurons; Initialization # 3

Figure 25 Probe Using Artificial, Meaningful, Noise-Free Data–Fit Based on 13 Raw Descriptors: (a) Summary, (b) Basis



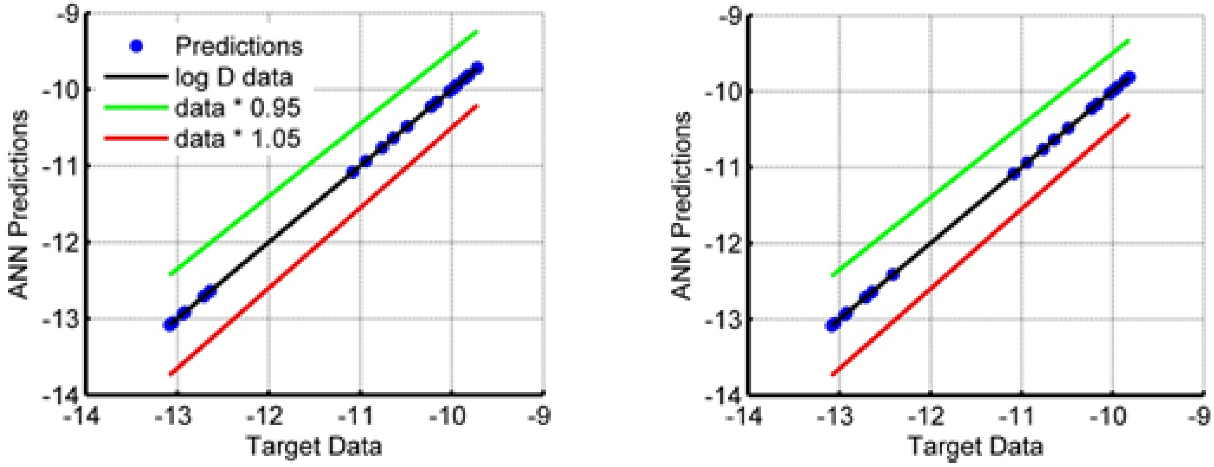
(a) Cross-validation summary
46% predictions are ±5% of targets; R^2 = 0.641

(b) Basis of a single point in the summary
Chemical # 4; Prediction error = -21%
57% of predictions are ±5% of targets; R^2 = 0.896
Target Variance 0.054; Skew -0.036; Kurtosis 1.8
Best case: 10 hidden neurons; Initialization # 1

Figure 26 Second Probe Using Artificial, Meaningful, Noise-Free Data–Fit Based on Five Principal Components: (a) Summary, (b) Basis

4.2.8 Fine-Tuning Best Predictions

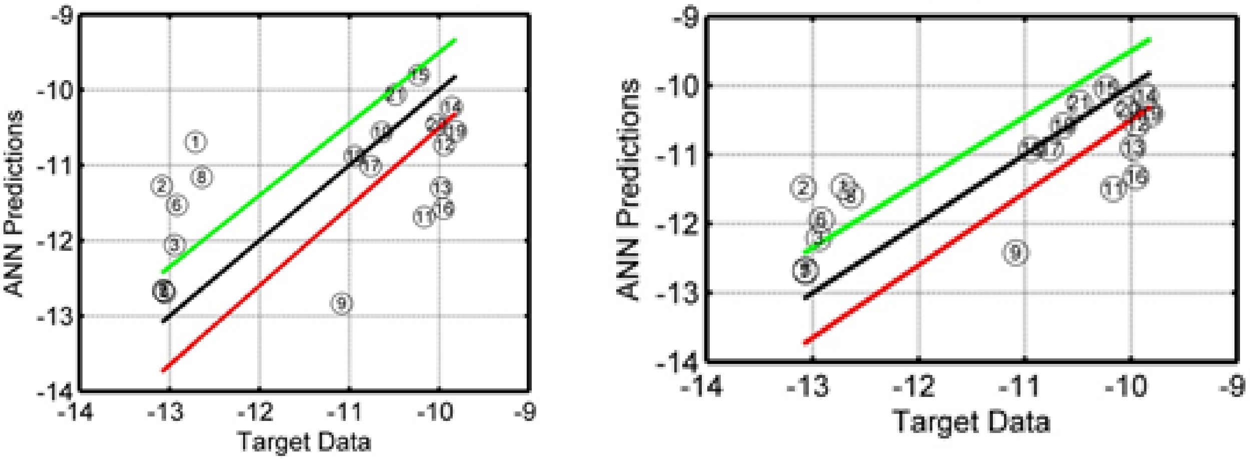
Guided by the results for these test cases, the focus was shifted to the actual log D data. The results of the probes using actual data are presented in Figures 27 to 29. Each of those figures has two parts as in Figures 22-26: a summary plot and a leave-one-out example. Initially, all 23 solvents were included as data chemicals (based on five expert-chosen raw descriptors), but when all the leave-one-out predictions were examined, two solvents (#9: benzaldehyde and #17 tetrahydrofuran) stood out (Figure 27). Upon reflection, it was recognized that when either chemical was left out, the network trained rather well. This suggests that both are outliers that detract from the training (although it is unclear why the training fit is fine when one or the other of these two outliers is still included in the fits shown in Figure 27). Accordingly, without further ado, the two chemicals were dropped from the set of data chemicals, and the cross-validation fitting was continued using five linear principal components with training of all 20 non-query data chemicals and prediction for the each set-aside chemical (Figure 28). Next, five raw descriptors (number of oxygen atoms, number of rotatable bonds, van der Waals volumes, molecular weight, and octanol/water partition coefficient) were used instead of the principal components; they were chosen by expert intuition regarding the phenomenon in question, namely diffusion of guest molecules through a rubbery polymer. Even within this small set of descriptors, there may be some redundancy (e.g., a correlation between molar volumes and molecular weights). Also, the inclusion of the partition coefficient (which is not a transport property, but instead an equilibrium property) may be questionable, but the awareness (that the diffusion coefficients here are lumped parameters that may be concentration dependent) prompted their inclusion.



(a) Chemical # 9 set aside
Chemical # 9; Prediction error = -5%
100% of predictions are ±5% of targets; $R^2 = 1$
Target Variance 0.106; Skew -0.397; Kurtosis 1.37
Best case: 10 hidden neurons; Initialization # 1

(b) Chemical # 17 set aside
Chemical # 17; Prediction error = 25%
100% of predictions are ±5% of targets; $R^2 = 1$
Target Variance 0.109; Skew -0.238; Kurtosis 1.27
Best case: 10 hidden neurons; Initialization # 2

Figure 27 Probe Using Actual Data—Fit Based on Five Expert-Chosen Raw Descriptors: (a) Summary, (b) Basis



(a) Cross-validation summary
48% of predictions are ±5% of targets; $R^2 = 0.347$

(b) Basis of a single point in the summary
Chemical # 4; Prediction error = -3%
45% of predictions are ±5% of targets; $R^2 = 0.594$
Target Variance 0.106; Skew -0.454; Kurtosis 1.44
Best case: 3 hidden neurons; Initialization # 5

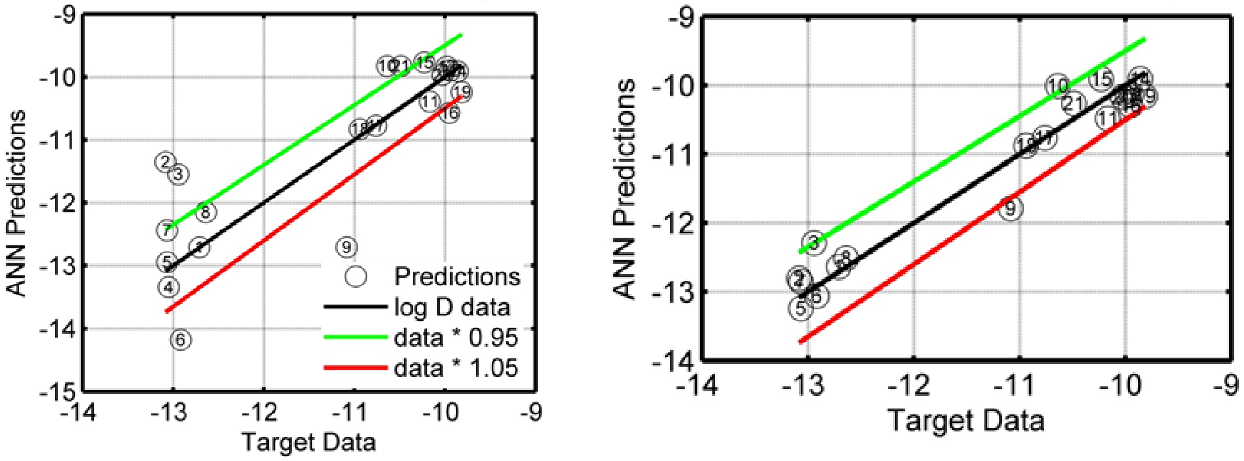
Figure 28 Probe with Actual Data—Fit Based on Five Principal Components with Two Chemicals Excluded: (a) Summary, (b) Basis

A key expectation from this study is that in using neural networks for QSAR, as long as the relationship between the target data and the molecular descriptors is not confounded (by data noise or the use of principal components instead of the raw descriptors for the training of the network), the training database need not be excessively large (unlike in the case of neural networks applied to pattern recognition—face, hand-writing, stock prices, etc.). Further, similarity between data chemicals and query chemicals is less of a requirement than merely that the range of descriptors in the training set be broad enough for the network to “learn” the dependence of the target property on the descriptors. The minimum number of data chemicals hence should just equal (or exceed slightly) the number of essential parameters (in the target-descriptor relationship). This “effective number” is usually unknown but cannot be more than a handful in macroscopic phenomena like permeation. Of course, more data chemicals would be needed as the data become noisier (or display less variance in a key descriptor); assuming that the noise is Gaussian, the phenomenon of “regression to the mean” should result in successful training and prediction with increasingly large databases.

These expectations were probed here, by re-visiting the test (where the “data” were calculated using an equation of no physical significance) and systematically reducing the number of data chemicals in the training set (randomly chosen from the full set of 23). The predictions stayed excellent even when the number of data chemicals was as low as 7, but that number could not be set to the lower limit of 3, presumably because the chemicals are not distinct enough to provide adequate training with the minimal set. It would be of interest to extend this exercise—of using fewer and fewer data chemicals—to predictions based on actual experimental data as well.

4.3 RESULTS

The best set of predictions, for the data available at the time of testing, were obtained in the cross-validation using the five raw descriptors that were chosen based on physicochemical intuition. These predictions are plotted in Figure 29. The results plotted in Figure 29a are also presented in Tables 9, 10, and 11 for data, CWA and simulant, and IPFS chemicals, respectively. Predictions may be improved with more uniform data, as suggested by the two distinct clusters that can be seen in the plot with a pronounced intervening gap (and as foreseen in the context of Figure 12 in Chapter 2).



(a) Cross-validation summary
67% of predictions are ±5% of targets; R^2 = 0.715

(b) Basis of a single point in the summary
Chemical # 4; Prediction error = -2%
85% of predictions are ±5% of targets; R^2 = 0.938
Target Variance 0.106; Skew -0.454; Kurtosis 1.44
Best case: 3 hidden neurons; Initialization # 5

Figure 29 Best Predictions with Actual Data—Fit Based on Five Expert-Chosen Raw Descriptors: (a) Summary, (b) Basis

Table 9 Best Predictions for Data Chemicals

$\log_{10}$ (D (m^2/s)) in butyl rubber

CHEMICAL	DATA	PREDICTION
acetonitrile	-12.71	-12.71
1-Butanol	-13.08	-11.35
2-ethoxyethanol	-12.94	-11.55
N-N-dimethylformamide	-13.05	-13.34
N-N-dimethylacetamide	-13.07	-12.95
ethylene glycol butyl ether	-12.92	-14.18
1-methyl-2-pyrolidinone	-13.07	-12.44
benzonitrile	-12.64	-12.15
ethyl acetate	-11.08	-12.71
1-2-Dichloroethane	-10.64	-9.83
butylamine	-10.17	-10.39
dichloromethane	-9.94	-9.90
benzene	-9.98	-9.83
hexane	-9.86	-9.91
heptane	-10.23	-9.77
triethylamine	-9.95	-10.57
Para Xylene	-10.76	-10.78
Mesitylene	-10.94	-10.83
Chloroform	-9.81	-10.24
trichloroethylene	-10.03	-9.97
tetrachloroethylene	-10.49	-9.83

Notes: The data are from column 6 of Table 7, after division by 10^{12} and taking the common logarithm. The predictions are “hands off” cross-validation predictions; i.e., the prediction for each chemical was obtained by excluding that chemical from the training set.

Table 10 Best Predictions for Query Chemicals: CWAs and Simulants

$\log_{10}$ (D (m^2/s)) in butyl rubber

CHEMICAL	PREDICTION
Triethyl Phosphate	-12.75
Bis(2-ethylhexyl) phthalate	-13.27
Diethyl 4-nitrophenyl phosphate	-12.92
Diethyl ester phosphonic acid	-12.99
Diethyl ethyl phosphonate	-12.73
Diethyl Malonate	-12.90
Di-isopropyl fluorophosphate	-12.70
Di-isopropyl methyl phophonate	-11.72
Dimethyl methyl phosphonate	-11.60
Diphenyl chloro phosphate	-11.38
Dipropylene glycol monomethyl ether	-13.13
Ethanol	-13.39
Ethyl chloro acetate	-12.84
Sarin	-10.76
Soman	-10.00
Tabun	-10.65
Trimethyl phospate	-12.79
HD	-12.80
CEES-HM	-11.39
CEMS	-10.64
CEPS	-9.97
Diethyl Adipate	-13.47
DMA	-12.90
MS[	-9.84
Diethyl Pimelate	-12.96
Lewsite	-12.62
Phenylarsine Oxide	-12.97
Lewsite Oxide	-14.84
VX	-12.52
Amiton	-13.23
BIS	-12.57
Bis(2-ethyl 1-hexyl) 2-ethyl 1-hexyl phosphonate	-12.15
DEP	-12.70
DEPPT	-11.00
DES	-13.27
Malathion	-13.28
Parathion	-12.96

Table 11 Best Predictions for Query Chemicals: IPFS

$\log_{10}$ (D (m^2/s)) in butyl rubber

CHEMICAL	PREDICTION
Molecule 2	-8.88
Molecule 3	-8.92
Molecule 4	-8.70
Molecule 5	-9.34
Molecule 6	-9.69
Molecule 7	-8.97
Molecule 8	-12.87
Molecule 9	-13.08
Molecule 10	-12.90
Molecule 11	-12.85
Molecule 12	-13.22
Molecule 13	-12.94
Molecule 14	-12.19
Molecule 15	-12.33
Molecule 16	-12.80
Molecule 17	-12.85

4.4 DISCUSSION

With reference to the predictions and data in Table 9 for the solvents that are the data chemicals, the prediction accuracy seems up to par with the entries in the "solubility challenge" [54, 55], summarized in Table 12. Any perceived inadequacy has to be viewed in perspective with the negligible prediction errors in the test cases which demonstrated the capability of the neural networks technique and the correctness of its implementation. Also, the "log D data" here are not raw data (unlike aqueous solubility), but a parameter extracted from desorption measurements under complications such as swelling and concentration dependence. The intense effort expended on diffusivity estimation (detailed in Chapter 2) is thus vindicated.

Not surprisingly, data-driven techniques require a focus on the "data". That is, for QSPR, prediction accuracy depends crucially on data clarity and coverage. What the network is being fed as target data must genuinely represent the phenomenon being investigated and not be too obscured by spurious effects, imprecise measurements, or fitting-model inaccuracy (if the "data" are actually estimates of some parameter, e.g., diffusion coefficient, as in the present case). Also, the target data points should cover a broad range of the descriptors that are significant to the underlying phenomenon. For example, if molecular weight is a key descriptor, the weights of the molecules chosen must include at least one high value and one comparatively low value (as in a two-level factorial design of experiments). When these requirements are met and due care is invested into assembling the "data" in data-driven prediction (i.e., when data of minimal error are obtained for chemicals chosen using a "uniform coverage" design [57]), the ANN can "learn" well with no guidance from theory and make fairly accurate predictions. With perfect data, the number of data chemicals need not be much larger than the effective number of descriptors which pertain to the phenomenon being modeled. Which descriptors are pertinent is usually unknown, but their number cannot be more than a handful in macroscopic phenomena like permeation. In practice, however, it is better to have as large a database as is feasible, since data errors cannot be totally eliminated and a database of only a few chemicals may not span a significant range of the relevant descriptors. The need for data quality is particularly important when the database is small. While a large database size does not guarantee prediction accuracy, it is better to train with a large database than a small one when the data are fuzzy; with a large database, the network may be able to avoid overfitting to the noise and detect the true patterns, i.e., capture the underlying generalities in the data without memorizing data idiosyncrasies. However, as noted in Chapter 2, cost considerations restrict the database size to a small number of chemicals; going forward, databases will be restricted to even fewer than the 23 solvents in the present study. That puts a premium on data quality.

Table 12 A Sampling of Results from the Solubility Challenge

FULL 32	28 MEASURED	28	24 4-OUTLIERS	24
±0.5logS	±0.5logS	R^2	±0.5logS	R^2
% CORRECT	% CORRECT		% CORRECT	
46.9	39.3	0.313	45.8	0.581
46.9	42.9	0.583	50.0	0.671
34.4	32.1	0.174	37.5	0.470
46.9	50.0	0.562	58.3	0.797
46.9	42.9	0.620	50.0	0.793
28.1	32.1	0.298	37.5	0.630
40.6	42.9	0.357	50.0	0.659
53.1	50.0	0.361	58.3	0.669
50.0	53.6	0.366	62.5	0.663
56.3	57.1	0.291	66.7	0.565
53.1	50.0	0.290	58.3	0.499
53.1	50.0	0.548	58.3	0.605
40.6	39.3	0.221	45.8	0.592
43.8	46.4	0.305	54.2	0.648
21.9	21.4	0.144	25.0	0.459
50.0	50.0	0.234	58.3	0.611
34.4	32.1	0.509	37.5	0.777
31.3	28.6	0.465	33.3	0.622
34.4	32.1	0.444	37.5	0.799

Source: Ref. 55, Table 2

This discussion is tacitly about the prediction quality for the 20 or so solvents, since only for these "data chemicals" both experimentally determined diffusivities and data-driven cross-validation predictions are available for a specific butyl rubber sample. The ultimate validation of the prediction method awaits the availability of experimentally determined diffusivities (preferably on the same membrane material) for the CWAs and the IPFS chemicals. In this context, it is appropriate to close with a promising comparison: For Di-isopropyl methyl phophonate (DIMP), the present estimate of 1.9 x 10^{-8} cm^2/s (from the entry in Table 10: -11.72 for the common logarithm of D in m^2/s) is straddled by the experimentally determined values of 7 x 10^{-8} cm^2/s (from immersion data) and 9 x 10^{-9} cm^2/s (from breakthrough data) previously reported [58] for Neoprene disks (50% rubber, 30% carbon black, and 9% plasticizer).

CHAPTER 5 CONCLUSIONS

The goals of the task were to develop:

- Experimental methods to measure chemical permeation through barrier materials.
- Computational-chemistry capabilities to calculate a variety of properties or "molecular descriptors" given only the formula of a chemical.
- Data-driven algorithms to predict the permeation of chemicals through barrier materials.

As detailed in the preceding chapters on "data," "descriptors," and "predictions," these goals were met, with varying degrees of success. In outline:

- Desorption transients of 23 organic solvents from a commercial butyl rubber sheet were measured using the immersion method. They were then comprehensively analyzed to obtain integral diffusion coefficients based on the linear constant-thickness Crank's Solution, as well as on a novel nonlinear model that accounts for swelling, and to obtain the parameters for solubilities using Flory-Huggins, Hildebrand, and Hansen theories. (Other measurement methods such as the "drop volume technique" can also be used to generate the diffusivity data.)
- Using commercial computational chemistry software (Spartan®) machine-readable molecular structures in SDF format were built for these organic solvents ("data chemicals") as well as about 60 threat agents and simulants ("query chemicals").
- Using commercial computational chemistry software (Sarchitect®), and based on the molecular structures, about 1,000 constitutional, topological, and conformational properties ("molecular descriptors") were calculated for all the data and query chemicals.
- A machine learning technique—ANN, for regression—was implemented by means of NSRDEC-developed Matlab® software and detailed probing studies, to provide data-driven prediction of target properties that characterize how chemical threat agents would permeate protective barrier materials. The network is trained using data chemicals for which descriptors, as well as target properties, are known and then used to make predictions for query chemicals for which only the descriptors are known. The predictions, while not perfect, are good enough to be on par with literature precedents for predicting aqueous solubility of pharmaceutical chemicals.
- Besides numerical data, descriptors, and predictions, this work has generated:
    - Protocols (for using commercial computational-chemistry software to render the structures and calculate the descriptors)
    - Tutorials (on ANN applied to property estimation)
    - Matlab® codes (one set of codes for extracting diffusivity estimates from desorption transients and another that, after some modifications, can be used for predicting a host of other properties of interest besides diffusion coefficients). The codes have many options and features, a coding-style aimed at error avoidance, extensive commentary and displays, and provisions for comprehensive record-keeping.

This work has brought out the power and promise of machine learning for property estimation, and delineated the scope for deploying "big data" techniques on small databases:

- BRANN is ideal for small datasets since it avoids the splitting of training data into training and validation sets, thus reducing the data demand.

- QSAR or QSPR, prediction accuracy depends crucially on data clarity and coverage. That is, what the network is being fed as training target data must genuinely represent the phenomenon being investigated and not be obscured by spurious effects, imprecise measurements, or fitting-model inaccuracy, especially if the "data" are not directly measured quantities such as aqueous solubility, but are actually theory-based estimates of some parameter, e.g., diffusion coefficient, as in the present case.
- Because structure-activity relations may involve nonlinearities that cannot always be linearized by logarithms, it is better to use the raw descriptors directly as independent variables, instead of linear combinations of the descriptors such as principal components.
- The data chemicals should cover a broad range of the descriptors that are significant to the underlying phenomenon; for example, if molecular weight is a key descriptor, the weights of the molecules chosen must include at least one high value and one comparatively low value as in a two-level factorial design of experiments.
- Not all the molecular descriptors are used in the calculations, whether in training the network or in using the trained network to make predictions. Only a select few key descriptors are used, in order to avoid "the curse of dimensionality". The culling of key descriptors was left to expert opinion in this work, but with some additional coding—using genetic algorithms for instance—descriptor selection can be automated as well.
- If the range of key descriptors in the training set is broad enough for the network to "learn" the dependence of the target property on the descriptors, similarity between data and query chemicals is less of a requirement.
- The minimum number of data chemicals should equal or exceed the number of effective parameters (in the target-descriptor relationship), which may not be more than a few in macroscopic phenomena like permeation. In fact, *a posteriori* estimates (that are a feature of the Bayesian regularization method used for training the network) put the number of effective parameters around five or six in the present cases.

The last point brings out an important difference between standard data mining and QSAR. Standard "big data" involves pattern recognition by sifting through very large but subjective databases—faces, hand-writing, key strokes, stock prices, voting, and such. Cheminformatics aimed at pharmaceutical drug discovery involves needle-in-the-haystack pattern recognition amidst elusive and complex biological phenomena. In contrast, data-driven QSAR involves a clinical analysis of carefully collected data on simple physicochemical systems.

In sum, the training database for data-driven QSAR need not be big, in principle. In practice, however, more data chemicals would be needed if the data are noisy or display less variance in key descriptors. While a large database size does not guarantee prediction accuracy, with a large database the network may be able to avoid overfitting to the noise and detect the true patterns. Also, the phenomenon of "regression to the mean" should result in successful training and prediction with increasingly large databases. However, since cost considerations usually restrict the database size to a small number of data chemicals, data quality remains paramount.

Data-driven methods can be used to predict not just permeation, but also many other physicochemical properties: solubilities, vapor pressures, partition coefficients, chemical degradation products, and, with additional effort, toxicity metrics. That is, the prediction codes developed in this work are not restricted to the diffusivity database, but can be used to make predictions of other physicochemical or toxicity properties, with some modifications, given the appropriate databases. Subject-matter expertise would be helpful in descriptor selection, however, i.e., in deciding which molecular features are important for the target property to be predicted. With some additional coding descriptor selection can be automated as well.

Despite the challenges of the original remit of the task, the effort has demonstrated and strengthened NSRDEC expertise in cheminformatics, a discipline that has much potential for addressing important DTRA objectives such as computational toxicology of NTAs.

This document reports research undertaken at the
U.S. Army Natick Soldier Research, Development and
Engineering Center, Natick, MA, and has been
assigned No. NATICK/TR- 14/021 in a
series of reports approved for publication.

CHAPTER 6   REFERENCES

1. J. Crank and G.S. Park, "Diffusion in Polymers" Academic Press, New York, 1968.
2. P.M. Bungay, H.K. Lonsdale, and M.N. de Pinho (editors), "Synthetic Membranes: Science, Engineering, and Applications," NATO ASI series, D. Reidel, Boston, 1983.
3. P. Neogi (editor), "Diffusion in Polymers," Marcel-Dekker, New York, 1996.
4. D. Rivin, R.S. Lindsay, W.J. Shuely, and A. Rodriguez, "Liquid Permeation Through Nonporous Barrier Materials," *Journal of Membrane Science*, 246 (2005) 39-47.
5. C. Nohilé, P.I. Dolez, and T.Vu-Khanh, "Parameters Controlling the Swelling of Butyl Rubber by Solvents," *Journal of Applied Polymer Science,* 110 (2008) 3926-3933.
6. E.T. Zellers, "Three-Dimensional Solubility Parameters and Chemical Protective Clothing Permeation. I. Modeling the Solubility of Organic Solvents in Viton Gloves," *Journal of Applied Polymer Science*, 50 (1993) 513-530.
7. E.T. Zellers, D.H. Anna, R. Sulewski, and X. Wei, "Critical Analysis of the Graphical Determination of Hansen's Solubility Parameters for Lightly Crosslinked Polymers," *Journal of Applied Polymer Science*, 62 (1996) 2069-2080.
8. K.M. Evans, W. Guo, and J. Hardy, "Modeling Solubility Parameters and Permeation Data of Organic Solvents Versus Butyl Gloves from Four Manufacturers," *Journal of Applied Polymer Science,* 109 (2008) 3867-3877.
9. C.M. Hansen, "Hansen Solubility Parameters: A Users' Handbook," 2nd Edition, Appendix A, Table A.1, CRC Press, Boca Raton, Florida, 2007.
10. J. Crank, "The Mathematics of Diffusion," 2nd edition, Clarendon Press, Oxford, 1979.
11. C.M. Balik, "On the Extraction of Diffusion Coefficients from Gravimetric Data for Sorption of Small Molecules by Polymer Thin Films," *Macromolecules,* 29 (1996) 3025-3029.
12. S.A. Altinkaya, N. Ramesh, and J.L. Duda, "Solvent Sorption in a Polymer-Solvent System—Importance of Swelling and Heat Effects," *Polymer*, 47 (2006) 8228-8235.
13. "Front-Fixing Methods," Chapter 5 in J. Crank, "Free and Moving Boundary Problems," Oxford University Press, 187-216, 1987.
14. R.D. Skeel and M. Berzins, "A Method for the Spatial Discretization of Parabolic Equations in One Space Variable," *SIAM J. Sci. Stat. Comput*., 11 (1990) 1-32.
15. L.F. Shampine, I. Gladwell, and S. Thompson, "Solving ODEs with Matlab®," Cambridge University Press, 2003.
16. R. Todeschini, V. Consonni, "Molecular Descriptors for Chemoinformatics," 2nd edition, Wiley-VCH, Weinheim, Federal Republic of Germany, 2009.
17. P. Willett, J.M. Barnard, G.M. Downs, "Chemical Similarity Searching," *J. Chem. Inf. Comput. Sci*., 38 (1998) 983-996.
18. S.C. Basak, B. Gute, D. Mills, "Similarity Methods in Analog Selection, Property Estimation and Clustering of Diverse Chemicals," *ARKIVOC*, 9 (2006), 157-210.
19. S.C. Basak, "Role of Mathematical Chemodescriptors and Proteomics-Based Biodescriptors in Drug Discovery," *Drug Dev. Res*., 72 (2010) 1-9.

20. L. Franke, O. Schwarz, L. Müller-Kuhrt, C. Hoernig, L. Fischer, S. George, Y. Tanrikulu, P. Schneider, O. Werz, D. Steinhilber, G. Schneider, "Identification of Natural-Product-Derived Inhibitors of 5-lipoxygenase Activity by Ligand-Based Virtual Screening," *J. Med. Chem*., 50 (2007) 2640-2646.
21. J. Xu, A. Hagler, "Chemoinformatics and Drug Discovery," *Molecules*, 7(2002) 566-600.
22. Q. Zhu, M.S. Lajiness, Y. Ding, and D.J. Wild, "WENDI: A Tool for Finding Non-obvious Relationships Between Compounds and Biological Properties, Genes, Diseases, and Scholarly Publications," *J. Cheminf*., 2:6 (2010) 1-9.
23. B.D. Gute, S.C. Basak, D. Mills, and D.M. Hawkins, "Tailored Similarity Spaces for the Prediction of Physicochemical Properties," *IEJMD*, 1 (2002) 374-387.
24. R.C. Glen, and S.E. Adams, "Similarity Metrics and Descriptor Spaces—Which Combinations to Choose?" *QSAR Comb. Sci*., 25 (2006) 1133-1142.
25. Source for molecular structure search using Pubchem, retrieved from http://pubchem.ncbi.nlm.nih.gov/search/search.cgi, 11/29/2013.
26. Source for molecular structure search using Chemical Abstracts Scifinder, retrieved from http://www.cas.org/products/scifinder, 11/29/2013.
27. Source for molecular structure format inter-conversion using Xemistry Cactvs toolkit, retrieved from http://www.xemistry.com/, 11/29/2013.
28. Source of Spartan computational chemistry software, retrieved from http://www.wavefun.com/products/spartan.html, 11/29/2013.
29. Source of Sarchitect computational chemistry software, retrieved from http://www.strandls.com/sarchitect/index.html, 11/29/2013.
30. M.R. Berthold, C. Borgelt, F. Höppner, and F. Klawonn, "Guide to Intelligent Data Analysis—How to Intelligently Make Sense of Real Data," Springer_Verlag, London, 2010.
31. T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning—Data Mining, Inference, and Prediction," 2nd edition, Springer, 2009.
32. D. Hand, H. Mannila, and P. Smyth, "Principles of Data Mining," MIT Press, 2001.
33. K.L. Priddy, and P.E. Keller, "Artificial Neural Networks—an Introduction," SPIE Press, 2010.
34. G. Montavon, G. Orr, and K-R. Müller, "Neural Networks: Tricks of the Trade," Lecture Notes in Computer Science / Theoretical Computer Science and General Issues (Book 7700), 2nd edition, Springer, 2012.
35. M.H. Beale, M.T. Hagan, and H.P. Demuth, "Neural Network Toolbox™—User's Guide," Mathworks (2013), retrieved from http://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf , 11/29/2013.
36. D.J. Livingstone (Editor), "Artificial Neural Networks—Methods and Applications," Methods in Molecular Biology, J.M. Walker (Series editor), Vol. 458, Humana Press, 2008.
37. M.T.D. Cronin, and D.J. Livingstone (Editors), "Predicting Chemical Toxicity and Fate," Methods in Molecular Biology, CRC Press, 2004.
38. J.A. Burns, and G.M. Whitesides, "Feed-Forward Neural Networks in Chemistry: Mathematical Systems for Classification and Pattern Recognition," *Chem. Rev*. 93 (1993) 2583-2600.

39. J.R. Votano, M. Parham, L.H. Hall, L.B. Kier, and L.M. Hall, "Prediction of Aqueous Solubility Based on Large Datasets Using Several QSPR Models Using Topological Structure Representation," *Chemistry & Biodiversity*, 1 (2004) 1829-1841.

40. J. Huuskonen, "Estimation of Aqueous Solubility for a Diverse Set of Organic Compounds Based on Molecular Topology," *J. Chem. Inf. Comput. Sci*. 40 (2000) 773-777.

41. Source of background information on the topic "Curse of dimensionality," retrieved from http://en.wikipedia.org/wiki/Curse_of_dimensionality, 11/29/2013.

42. L.I. Smith, "A Tutorial on Principal Components Analysis," retrieved from http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf, 11/29/2013.

43. J. Shlens, "A Tutorial on Principal Components Analysis," retrieved from http://www.snl.salk.edu/~shlens/pca.pdf, 11/29/2013.

44. S. Roweis, "EM Algorithms for PCA and SPCA," *Neural Information Processing Systems*, 10 (1997) 626-632.

45. M. Scholz, "Nonlinear PCA Toolbox for Matlab®," retrieved from http://www.nlpca.org , 11/29/2013.

46. M. Scholz, "Validation of Nonlinear PCA," *Neural Processing Letters*, 36 (2012) 21-30.

47. M.A. Kramer, "Nonlinear Principal Component Analysis Using Auto-Associative Neural Networks," *AIChE J*., 37 (1991) 233-243.

48. I. Guyon, and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, 3 (2003) 1157-1182.

49. S. Perez, "Apply Genetic Algorithm to a Learning Phase of a Neural Network," Unpublished source of background information on the topic "Genetic Algorithm," retrieved from http://www.ics.uci.edu/~dramanan/teaching/ics273a_winter08/projects/sperez1_GANN.pdf, 11/29/2013.

50. F. Burden, and D. Winkler, "Bayesian Regularization of Neural Networks," Chapter 3, pages 25-44 in Ref. 31 (above cited).

51. D. Gianola, H. Okut, K.A. Weigel, and G.J.M. Rosa, "Predicting Complex Quantitative Traits with Bayesian Neural Networks; a Case Study in Jersey Cows and Wheat," *BMC Genetics*, 12 (2012) 12:87, retrieved from http://www.biomedcentral.com/1471-2156/12/87/, 11/29/2013.

52. D.K. Williams Jr., A.L. Kovach, D.C. Muddiman, and K.W. Hanck, "Utilizing Artificial Neural Networks in Matlab® to Achieve Parts-Per-Billion Mass Measurement Accuracy with a Fourier Transform Ion Cyclotron Resonance Mass Spectrometer," *J. Am. Soc. Mass Spectrom*. 20 (2010) 1303-1310.

53. Source of background information on the topic "Cross-validation," retrieved from http://en.wikipedia.org/wiki/Cross-validation_(statistics), 11/29/2013.

54. A. Llinas, R.C. Glen, and J.M. Goodman, "Solubility Challenge: Can You Predict the Solubilities of 32 Molecules Using a Database of 100 Measurements," *J. Chem. Info. Model*. 48 (2008) 1289-1303.

55. A.J. Hopfinger, E.X. Esposito, A. Llinas, R.C. Glen, and J.M. Goodman, "Findings of the Challenge to Predict Aqueous Solubility," *J. Chem. Info. Model*. 49 (2009) 1-5.

56. Source of background information on the topic “Moriguchi equation,” retrieved from http://www.talete.mi.it/help/dproperties_help/index.html?molecular_properties.htm, 11/29/2013.

57. V. Vijay, R.E. Baynes, S.S. Young, and J.E. Riviere, “Selection of Appropriate Training Set of Chemicals for Modeling Dermal Permeability Using Uniform Coverage Design,” *QSAR Comb. Sci*., 28 (2009) 1478 – 1486.

58. A.P. Angelopoulos, N.S. Schneider, and J.H. Meldon, “Numerical Simulation for the Permeation of Barrier Materials by Neat Liquid Droplets,” U.S. Army Technical Report MTL TR 88-35, 77-78, 1988.

This page intentionally left blank

APPENDIX A
UNITS OF PERMEABILITY

The conventions are different for the permeation of gases and liquids. Also, the units are not SI.

GASES The Barrer (in honor of the late Professor R.M. Barrer, a pioneer in membrane science) is commonly used for gas permeability:

P units: 1 Barrer = 10^{-10} cm^3@STP / (cm . s. cmHg) = 10^{-10} cm^3@STP . cm / (s . cm^2 . cm-Hg)

Experimentally P is determined by multiplying the observed permeation flux (which is permeation rate divided by membrane area) by the membrane thickness and dividing by the trans-membrane partial pressure difference of the permeant:

- The term cm^3 @STP / s refers to the molar trans-membrane permeation rate of the diffusing species converted from the permeation temperature and atmospheric pressure to the standard conditions of 0°C and 1 atm (divide by 22400 to get gram moles / s).
- The cm refers to the membrane thickness.
- The cm^2 refers to membrane area.
- The cm-Hg refers to the trans-membrane partial pressure as measured by a mercury barometer.
- The 10^{-10} is just a dimensionless constant, necessary to handle the typical range of (low) gas permeabilities in polymers.

Soubility, defined as the partition coefficient or Henry's law slope (i.e. the ratio of solute concentration in the membrane and the solute partial pressure at equilibrium in the surrounding gas phase) yields the amount of solute (cm^3@STP) per unit volume of membrane (cm^3) per unit partial pressure (cmHg)

S units = cm^3@STP / (cm^3. cm-Hg)

Diffusivity has the familiar units: D units = cm^2 / s

These units, as can be easily verified, obey the equation P = S $*$ D.

Note: Permeance is *not* the same as permeability, but is the ratio of permeability and thickness.

$$\text{Permeance} \equiv \overline{\mathrm{P}} = \frac{\text{Permeabilty}}{\text{Thickness}} \qquad \Rightarrow \overline{P} = Permeance = \frac{P}{L} \neq \text{Permeability}$$

Multiplying permeability by membrane area and partial pressure difference and dividing by membrane thickness should yield the permeation rate.

LIQUIDS The difference from gases has to do with how solubility of single component liquid permeants is reported: instead of the partition coefficient (a relative measure that compares the solubility in the membrane with solubility or activity in the adjacent fluid), the absolute solubility in the membrane is used (since the fluid phase activity is 1).

S units = Molar: mole / (cm^3 of membrane) or Mass: g / (cm^3 of membrane)

D units = cm^2 / s   P = S $*$ D = mole / (cm $*$ s) or g / (cm $*$ s)

For these experiments $w_{SP} = \frac{W_s}{W_P} = \frac{W_0 - W_\infty}{W_\infty} \quad \frac{\text{g solvent}}{\text{g polymer}}$

where W_∞ is the weight at the end of the final immersion step and W_0 is the weight at the end of the desorption or "drying" step. Multiplying this "weight fraction" or, more precisely, solvent-to-polymer-weight ratio by the polymer density,

$$S_{mass} = w_{SP} * \rho_P \quad \frac{\text{g solvent}}{\text{cm}^3 \text{ polymer}}$$

Dividing by the solvent molecular weight, $S_{mole} = \frac{w_{SP} * \rho_P}{M_W} \quad \frac{\text{mole solvent}}{\text{cm}^3 \text{ polymer}}$

Putting it all together,

$$P_{mass} = D * w_{SP} * \rho_P \quad \frac{\text{g solvent}}{\text{cm} \quad \text{s}} \qquad P_{mole} = \frac{D * w_{SP} * \rho_P}{M_W} \quad \frac{\text{mole solvent}}{\text{cm} \quad \text{s}}$$

Dividing permeability by membrane thickness should give the mass or molar flux:

g solvent / cm^2 s or g solvent / cm^2 s, respectively. Multiplying the flux by the membrane area should give the permeation rate: g solvent / s or g solvent / s.

ASTM results for gloves are reported as "rates," which are actually mass fluxes in μg/(cm^2 min).

$$\text{Flux}_{mass} = \frac{D * w_{SP} * \rho_P}{L_{cm}} \quad \frac{\text{g}}{\text{cm}^2 \quad \text{s}}$$
$$= 6.0 \times 10^8 \frac{D * w_{SP} * \rho_P}{L_{cm}} \quad \frac{\text{mg}}{\text{m}^2 \quad \text{min}}$$
$$= 6.0 \times 10^7 \frac{D * w_{SP} * \rho_P}{L_{cm}} \quad \frac{\mu\text{g}}{\text{cm}^2 \quad \text{min}}$$

Note: If the membrane thickness is reported in meters, the ASTM rate will be

$$6.0\times10^{5}\frac{D*w_{SP}*\rho_{P}}{L_{m}} \quad \frac{\mu g}{cm^{2}\quad min}$$

D = Diffusion coefficient (cm^2/s)

L_{cm} = Membrane thickness (cm)

L_m = Membrane thickness (m)

$$w_{SP} = \text{solubility} \quad = \frac{W_0 - W_\infty}{W_\infty} \quad \frac{\text{g solvent}}{\text{g polymer}}$$

$$\rho_P = \text{polymer density} \quad \frac{g}{cm^3}$$

Evans et al.* plotted the product of membrane thickness and permeation fluxes (with the product in the units of μg / m min). For comparison with their Figure 2, the diffusivity and solubility results were combined as follows:

$$SSPR = Flux_{mass} = \frac{D*w_{SP}*\rho_{P}}{L} \quad \frac{g}{cm^{2}\quad s}$$

$$L*SSPR = D*w_{SP}*\rho_{P} \quad \frac{g}{cm\quad s} = 6.0\times10^{9}*D*w_{SP}*\rho_{P} \quad \frac{\mu g}{m\quad min}$$

* K.M. Evans, W. Guo, and J. Hardy, "Modeling Solubility Parameters and Permeation Data of Organic Solvents Versus Butyl Gloves from Four Manufacturers," *Journal of Applied Polymer Science,* 109 (2008) 3867-3877.

This page intentionally left blank

APPENDIX B
A TUTORIAL ON NEURAL NETWORKS

Figure B-1 shows an in-silico prediction model. Figure B-2 is a schematic of the input and desired output for a neural network. Figures B-3 and B-4 are schematics of neural networks with one and two hidden layers, respectively.

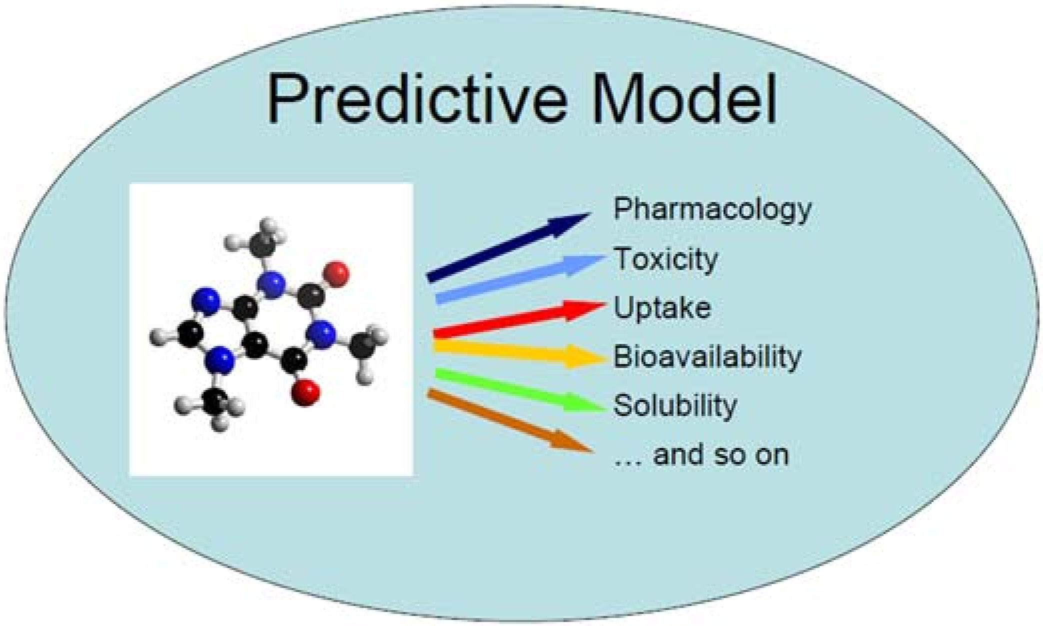


Figure B-1 In-Silico Predictions in Biology and Chemistry[1]

Note: Except for Figures B-9, B-12, B-13, B-14, and B-15, the rest of the schematics in this appendix are from Matlab® documentation[2] and session screen-shots.

[1] M. Cronin, "Quantitative Structure-Permeability Relationships—Useful or Useless?" School of Pharmacy and Chemistry, Liverpool John Moores University, PDF retrieved from an internet search: http://www.google.com/webhp?nord=1#nord=1&q=cronin+quantitative+structure+permeability+relationships, 11/29/2013.

[2] M.H. Beale, M.T. Hagan, and H.P. Demuth, "Neural Network Toolbox™—User's Guide," Mathworks (2013), retrieved from http://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf , 11/29/2013.

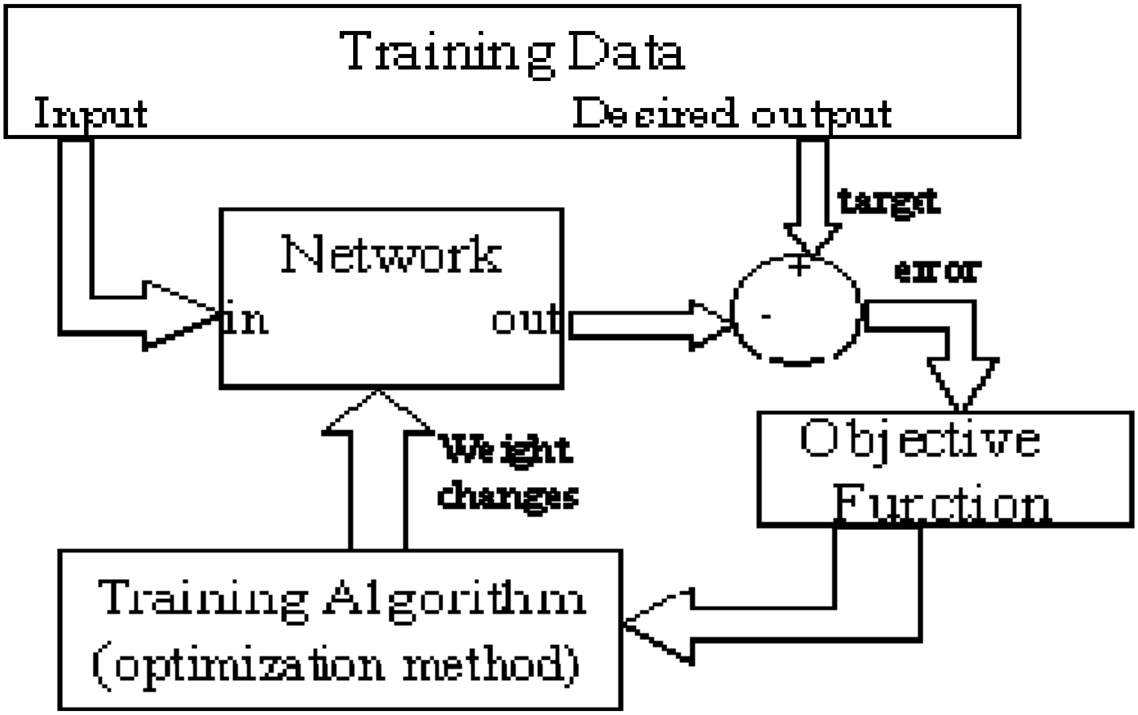


Figure B-2 Building a Neural Network

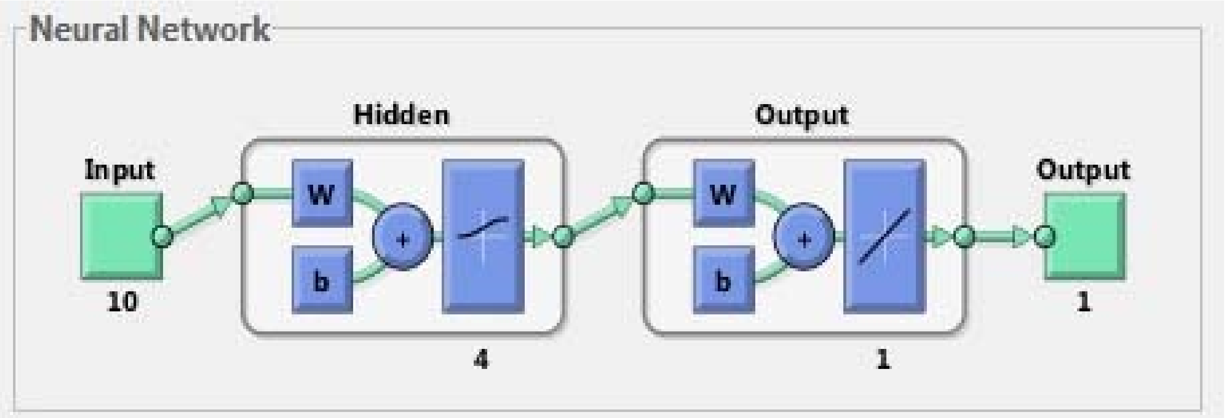


Figure B-3 A Neural Network with One Hidden Layer

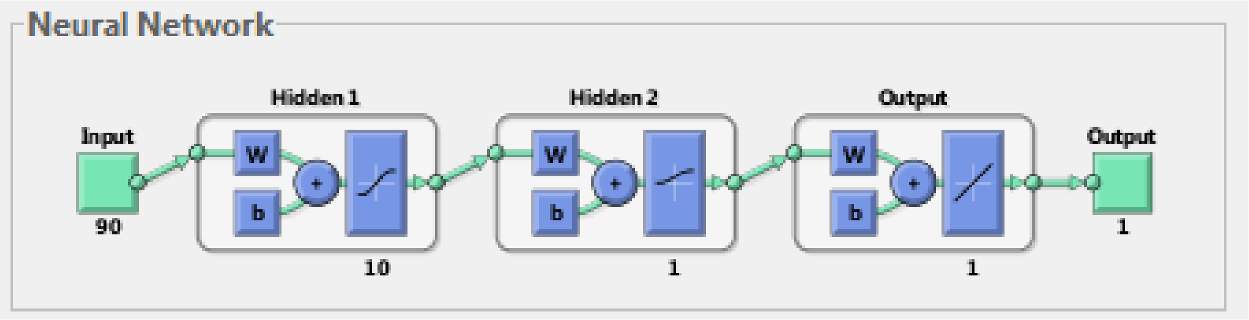


Figure B-4 A Neural Network with Two Hidden Layers

Unlike a standard linear or nonlinear least squares regression (in which the fitting equation is explicit), neural networks use a hidden, highly nonlinear, fitting function that is made up of nonlinear elements (or neurons illustrated in Figures B-5 and B-6). Notice that the neuron shown in Figure B-6 acts on every descriptor. The descriptors are multiplied by appropriate weights, the results are summed (along with a bias or intercept), and, finally, the sum is put through a transfer function to produce the output of this neuron.

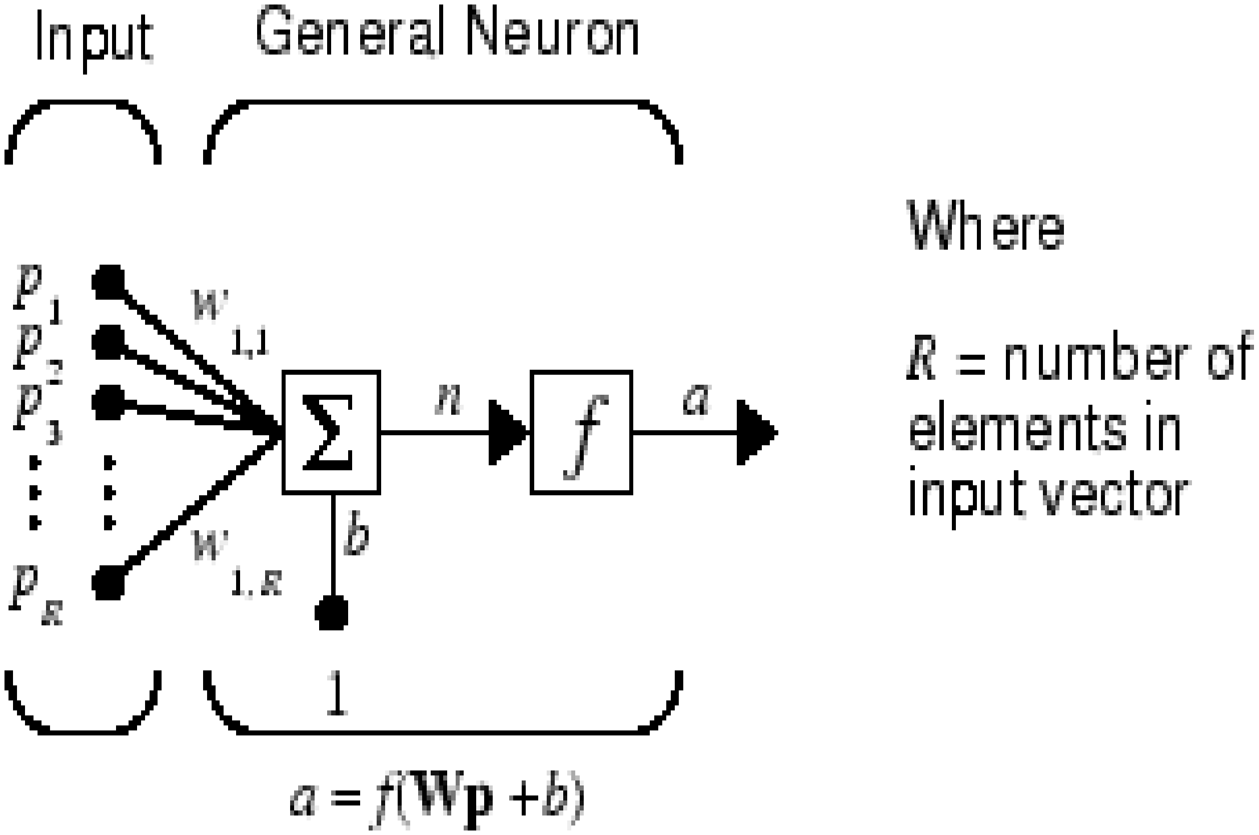
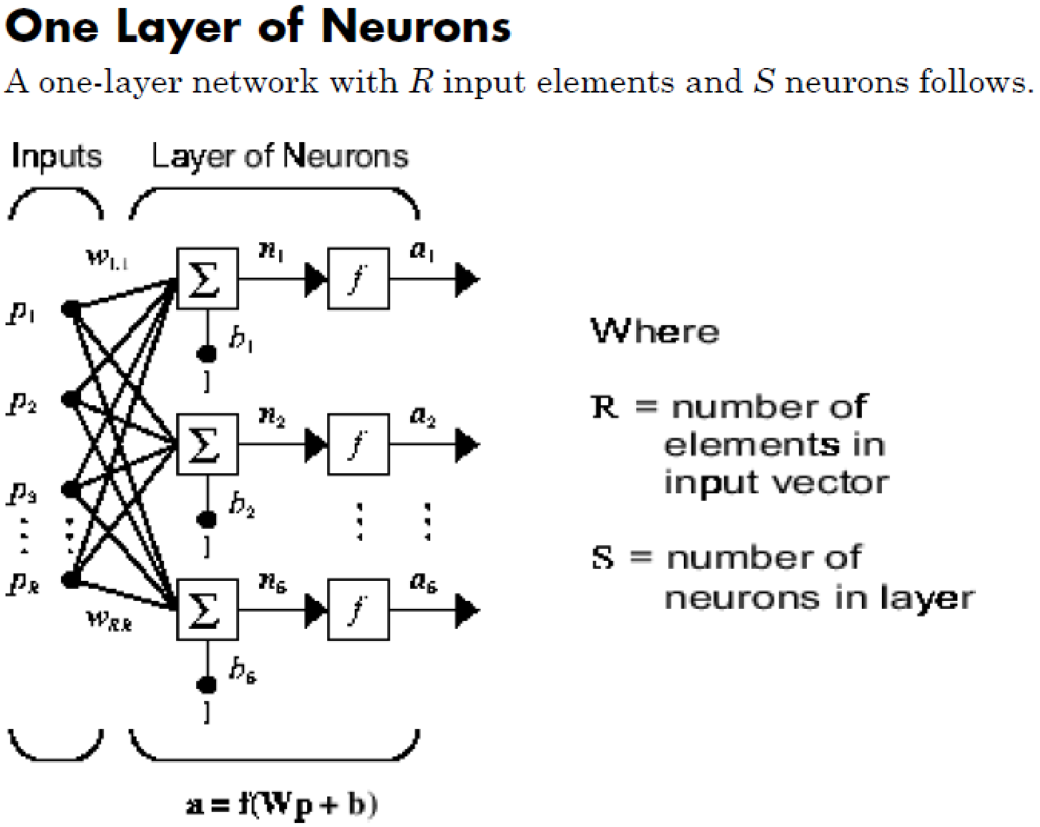


Figure B-5 A Single Hidden Neuron

The operations of a single neuron illustrated in Figure B-5 are shown in Figure B-6 as repeated for each neuron in a layer of neurons. For the first layer of neurons, the inputs are the descriptors; for subsequent layers of neurons, the inputs are the outputs of the preceding layer. The output from the very last layer constitutes the network's prediction. It may become apparent by now how the neural network is really fitting the data to a complicated quilt of equations.



One Layer of Neurons

A one-layer network with R input elements and S neurons follows.

Where

R = number of elements in input vector

S = number of neurons in layer

$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p} + \mathbf{b})$$

Figure B-6 A Layer of Hidden Neurons

Notice that the output for the log-sigmoid transfer function shown in Figure B-7 can only be positive. Hence, that transfer function is good for target properties that are always positive. Also, Figures B-7 and B-8 bring out another pertinent aspect of neural networks; namely, the network only deals with predictions that are bounded by 0 to 1 or -1 to +1. Actual target data hence have to be suitably normalized using a min-max algorithm; a similar center-and-scale normalization is done for the descriptors. After the fitting is done, the normalizations are of course reversed, rendering the predictions in the original scale of the data.

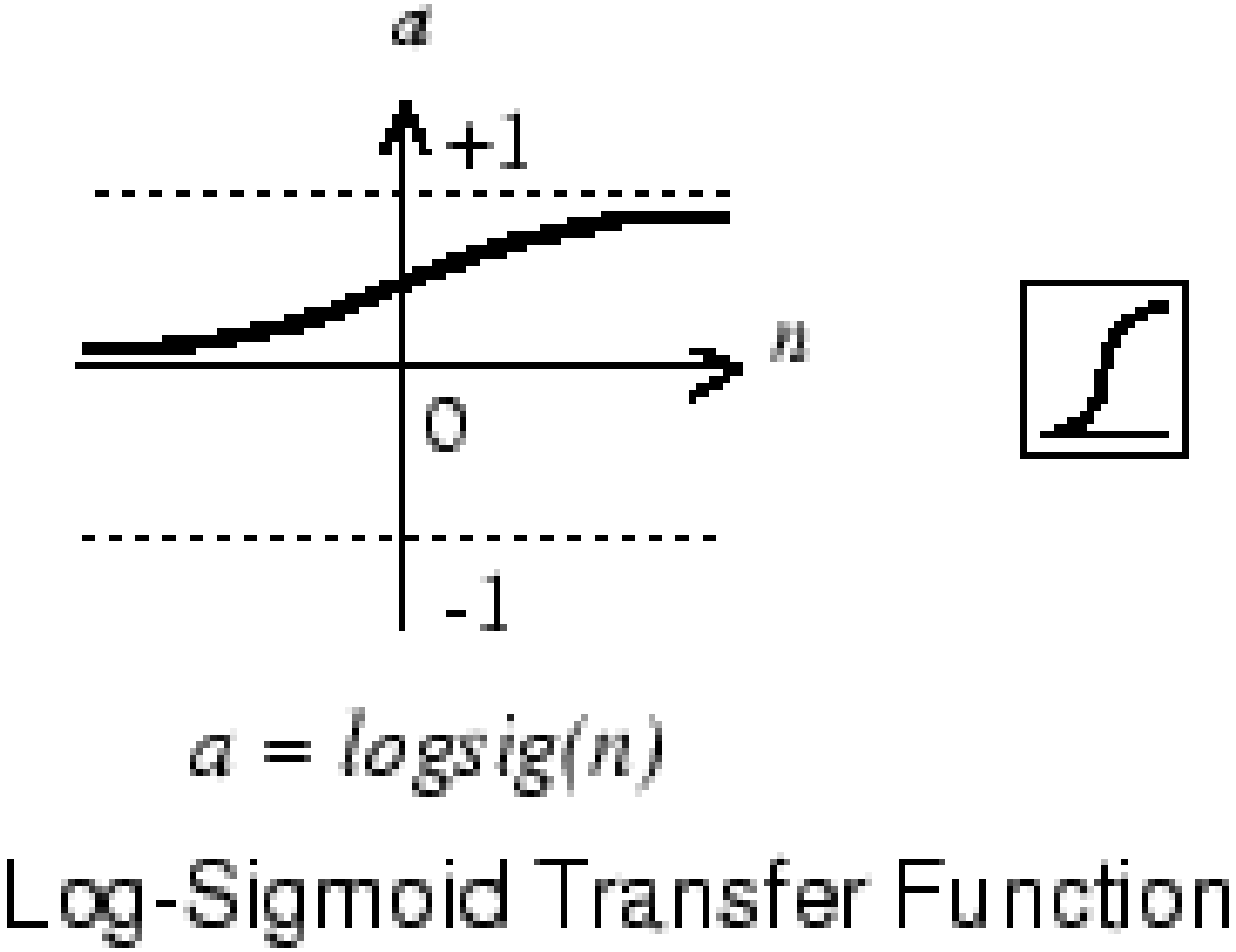


Figure B-7 A Nonlinear Transfer Function with Only Positive Output Potential

Notice that the output for the tan-sigmoid transfer function shown in Figure B-8 can be both negative and positive. Hence that transfer function is good for target properties that are not always positive.

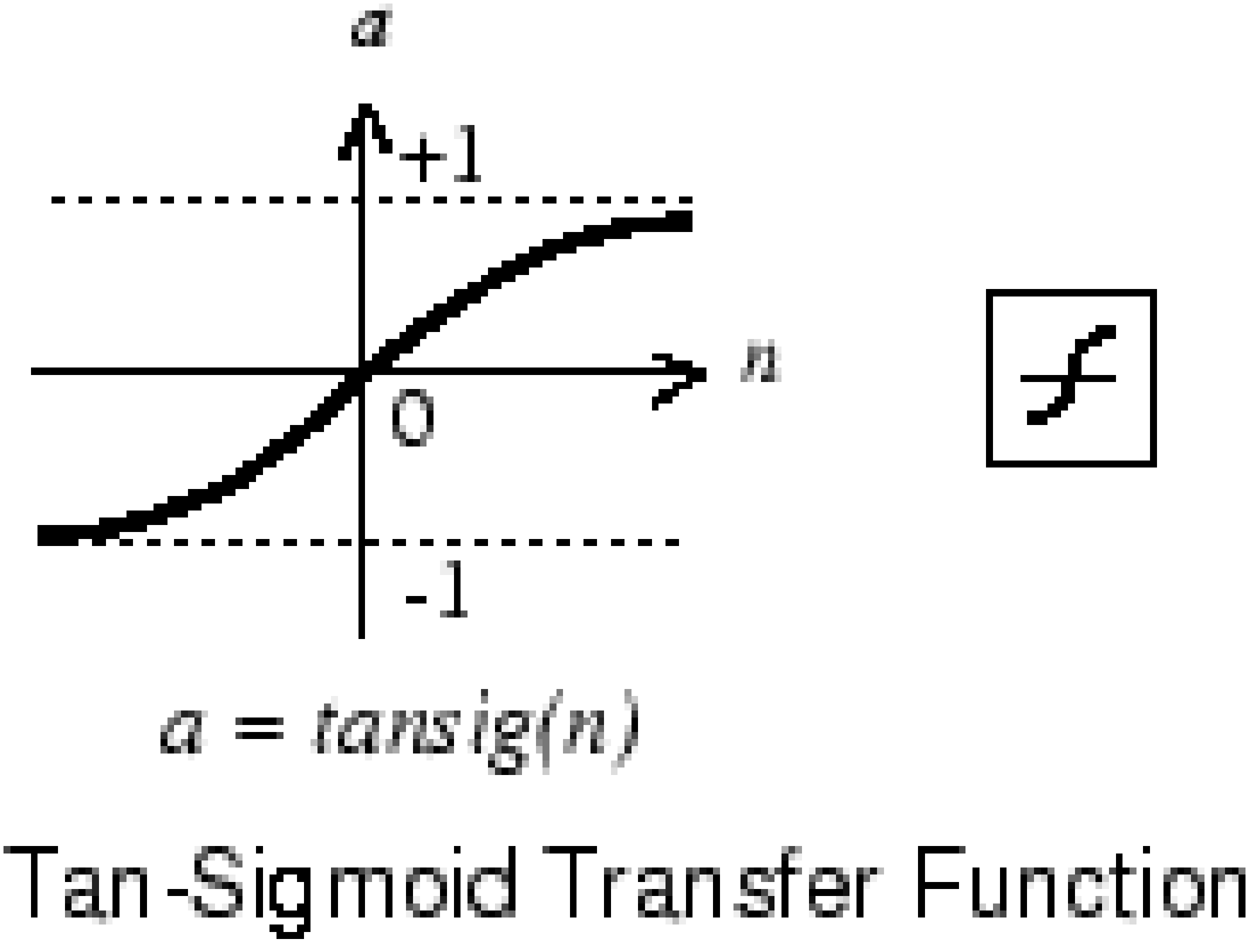


Figure B-8 A Nonlinear Transfer Function with Both Positive and Negative Output Potential

Fitting with a neural network essentially consists of indirectly adjusting such scale parameter that, in turn, alters a single nonlinear term, namely a hidden neuron. (The argument of the sigmoidal transfer function is a weighted sum of the inputs to this neuron, typically the descriptors.) By combining several neurons, a neural network can in principle fit any nonlinear data. The trick is not going overboard using too many neurons to get a perfect fit of the training set (in which case the network ends up simply memorizing every idiosyncrasy of the training data and not learning the general trends) and short-changing the generalization ability of the network, leading to poor fits for test sets.

Figure B-9 is a plot of the sigmoid function $\sigma(v) = 1/(1+\exp(-v))$ (red curve), commonly used in the hidden layer of a neural network. Included are $\sigma(sv)$ for $s = 1/2$ (blue curve) and $s = 10$ (purple curve). The scale parameter s controls the activation rate, and it can be seen that large s amounts to a hard activation at $v = 0$. Note that $\sigma(s(v - v_0))$ shifts the activation threshold from 0 to v_0.

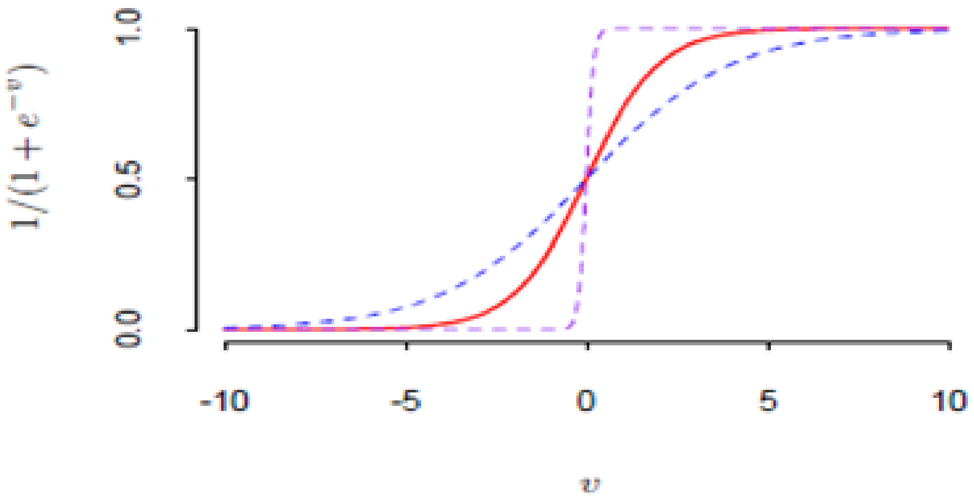


Figure B-9 Predictive Power of the Sigmoidal Transfer Function[3]

A linear transfer function (Figure B-10) is typically used in the final, output layer of a neural network. It can be discerned that if the network has only the output layer (and no hidden layers, i.e., no nonlinear elements) the resulting fit reduces to linear regression.

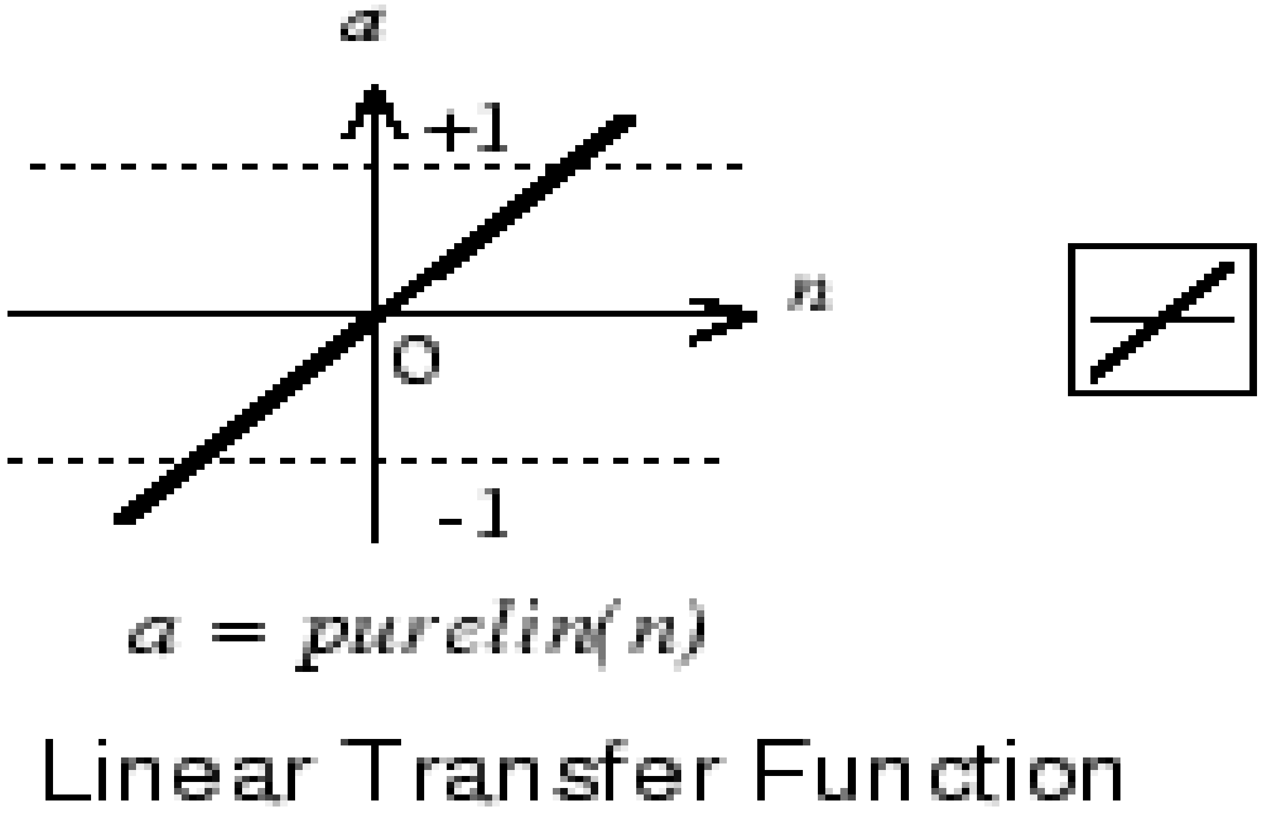


Figure B-10 Linear Transfer Function

[3] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning—Data mining, Inference, and Prediction," 2nd edition, Springer, 2009, Figure 11.3.

As shown in Figure B-11, converging to a local optimum (and hence failing to reach the global optimum) is a problem for nonlinear curve-fitting or optimization in general and for feed-forward neural networks in particular. Hence, it is important to re-start the convergence from new starting points through repeated initializations and accept the best of the fits.

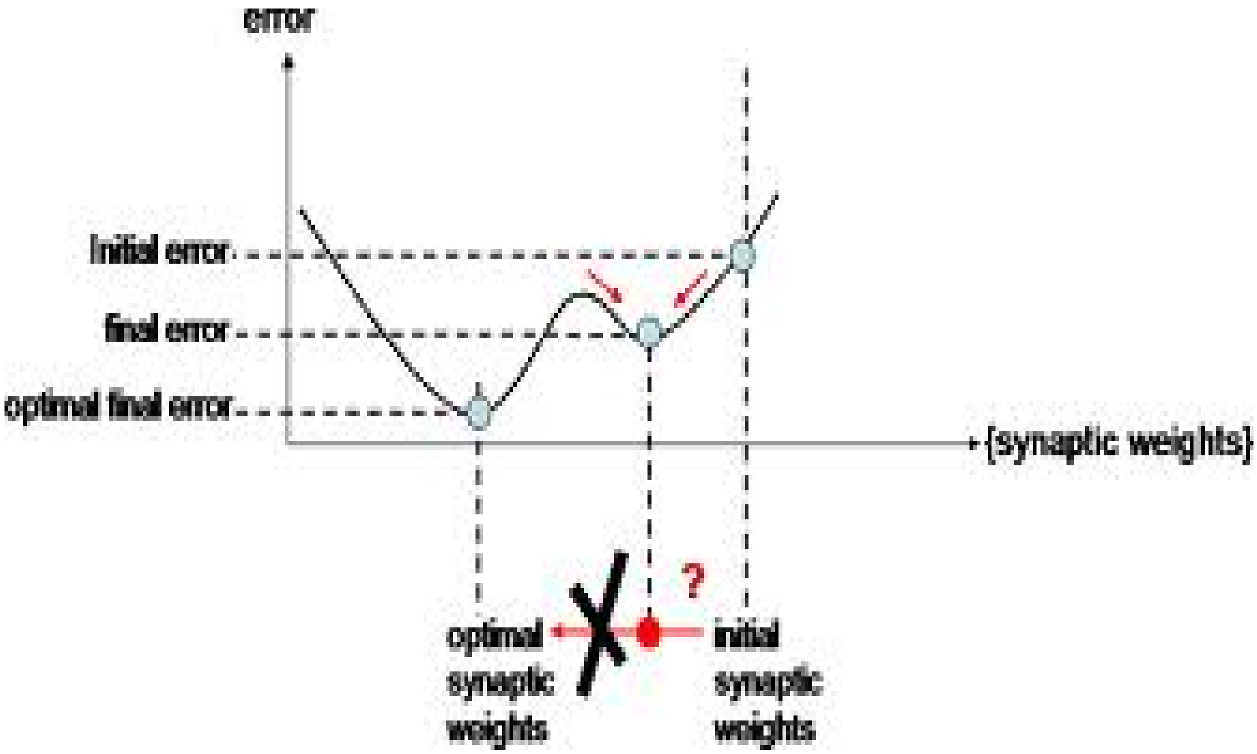


Figure B-11 Convergence Traps due to Local Minima

Overfitting is a known problem in linear least squares fitting, illustrated in Figure B-12 with a polynomial fit.   M is the degree of the polynomial. Low M entails "bias"; too high an M entails "variance", i.e., overfitting.

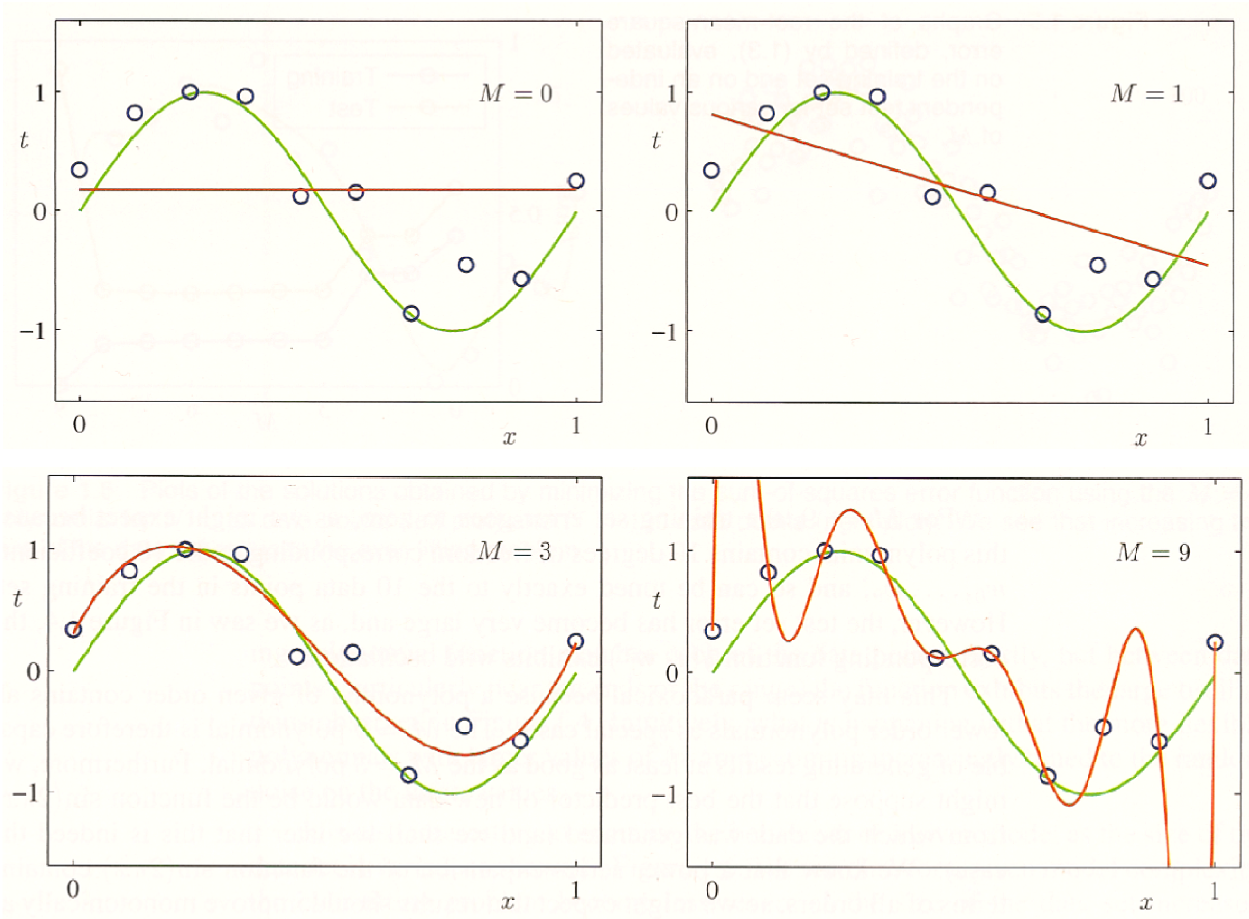


Figure B-12 Overfitting in Linear Regression[4]

Overfitting is known in nonlinear neural networks also. In the example shown in Figure B-13, M is the number of hidden neurons. M = 1 entails "bias" (i.e., an inadequate fit); M = 3 offers the optimal fit that captures the general trend without being side-tracked by the idiosyncrasies; M = 10 is unacceptable because of "variance", i.e., overfitting, with the fit faithfully going through every error-laden datum.

[4] Source of background information on the topic "Overfitting in Linear Regression," retrieved from http://www.google.com/webhp?nord=1#nord=1&q=lecture+3+linear+regression+machine+learning+cuny, 11/29/2013.

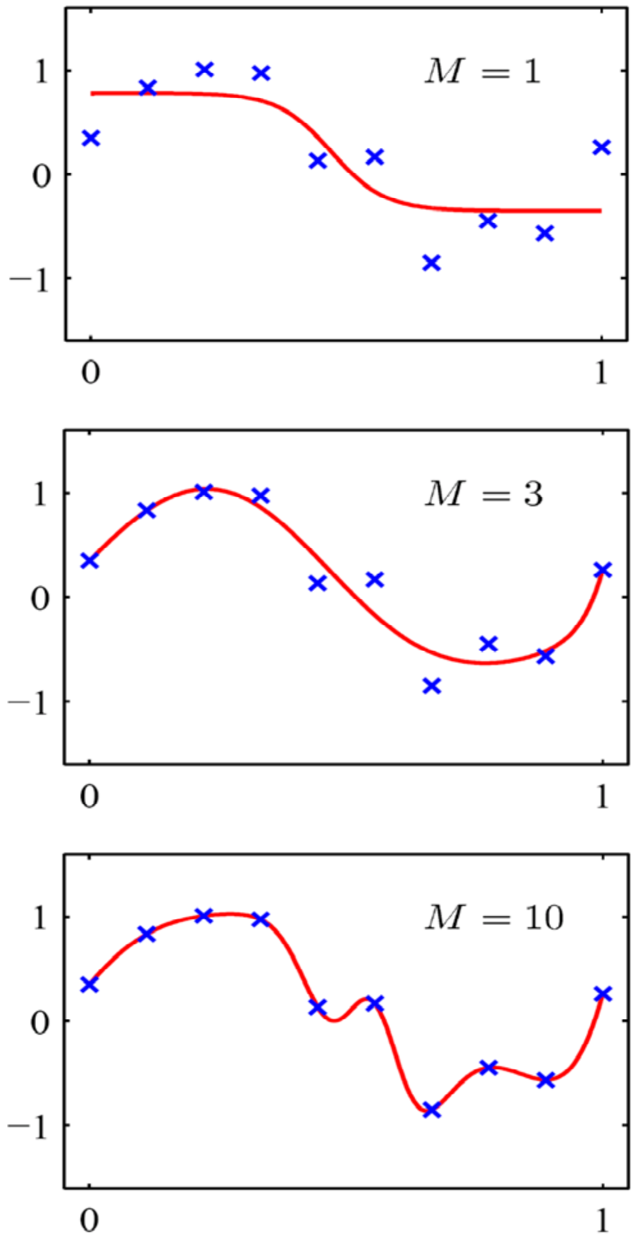


Figure B-13 Overfitting in Neural Networks[5]

The recipe for the "early stopping" approach to avoid overfitting (Figure B-14) is to set aside a portion of the target database for validation and the rest of the database for training and begin training the network using only the training set, while monitoring the prediction-versus-data error for both training and validation sets. The error for the training set will keep on dropping; the error for the validation set will also drop initially but will begin to rise eventually. Stop the training at this point.

[5] Source of background information on the topic "Overfitting in Neural Networks," retrieved from http://www.cedar.buffalo.edu/~srihari/CSE574/Chap5/Chap5.5-Regularization.pdf, slide 5, 11/29/2013.

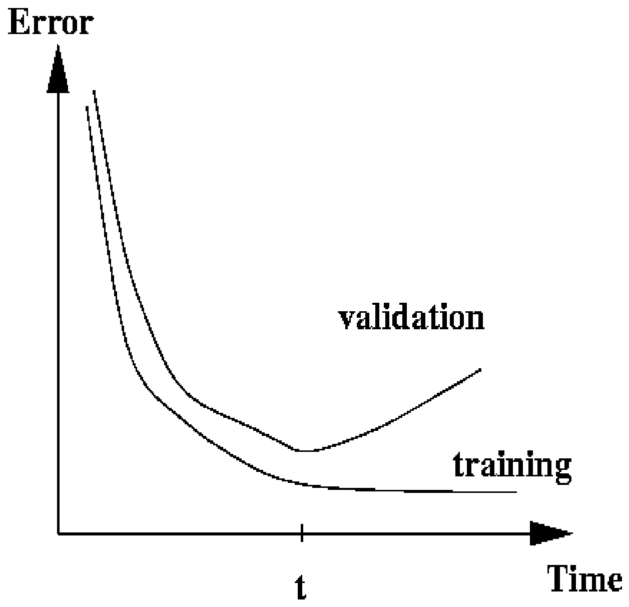


Figure B-14 Early Stopping to Avoid Overfitting[6]

By making the output the same as the input with an intermediate bottle-neck layer, dimensionality can be reduced while preserving nonlinearity. Nonlinear principal components are extracted using such networks (Figure B-15).[7] With the original set of descriptors as the input as well as the output of the network, the number of hidden neurons in and the outputs of the bottle-neck layer are the number and values of the nonlinear principal components, respectively.

[6] Source of background information on the topic "Early Stopping to Avoid Overfitting in Neural Networks," retrieved from http://www.willamette.edu/~gorr/classes/cs449/overfitting.html, 11/29/2013.

[7] "Curse of Dimensionality", retrieved from http://en.wikipedia.org/wiki/Curse_of_Dimensionality, 11/29/2013.

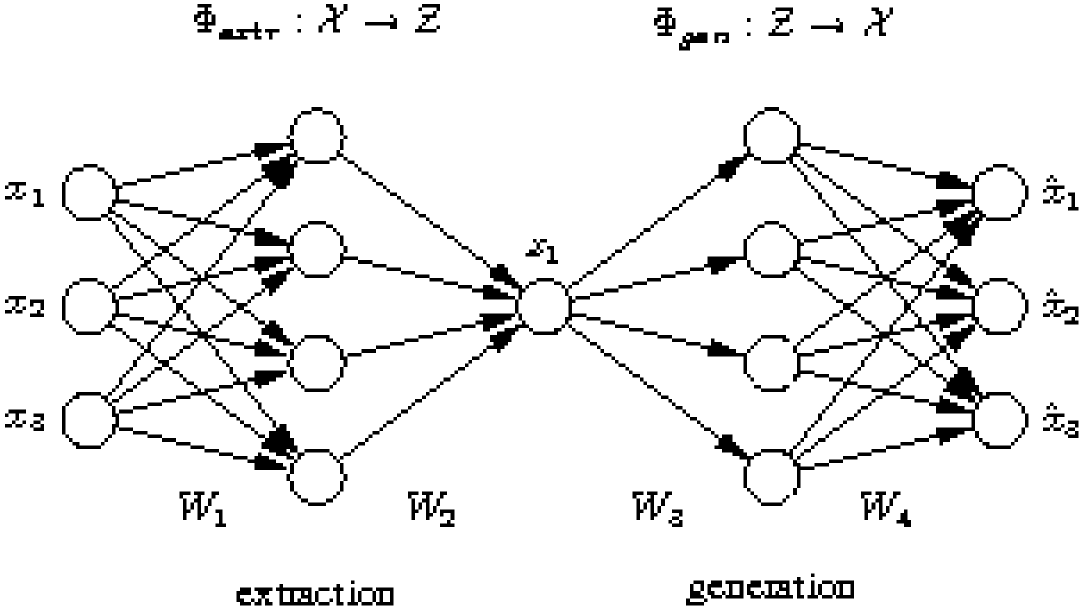

$\Phi_{extr} : \mathcal{X} \rightarrow \mathcal{Z}$
$\Phi_{gen} : \mathcal{Z} \rightarrow \mathcal{X}$
x_1
x_2
x_3
z_1
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
W_1
W_2
W_3
W_4
extraction
generation

Figure B-15 An Auto-Associative Neural Network

NOTATION

a_s	Solvent activity at the external film boundaries	Unit-less
C_0	Initial concentration of solvent in the film	Kg Mol / m^3(film+solvent)
C_s	Concentration of solvent in the film	Kg Mol / m^3(film+solvent)
C_{se}	Concentration of solvent at the film's external boundaries	Kg Mol / m^3(film+solvent)
d	Film thickness	m
D	Solvent diffusion coefficient in film, normalized by D_{ref}	Unit-less
D_s	Solvent diffusion coefficient in the film (solvent-polymer binary diffusion coefficient)	m^2/s
h	Film half-thickness	m
h_0	Initial film half-thickness	m
$\frac{M_t}{M_0}$	Solvent loading in the film at time t, relative to the initial loading	Unit-less
t	time	s
V_s	Molar volume of solvent	m^3/ Kg Mol
W_A	Weight of additives	Kg
W_F	Weight of the pristine disk, before any sorption or desorption	Kg
W_P	Weight of polymer	Kg
W_S	Weight of solvent	Kg
W_0	Weight at the beginning of desorption, after the soaking step	Kg
W_∞	Weight at the end of desorption	Kg
w_{SP}	Solvent to polymer weight ratio	Unit-less
w_{SA}	Solvent to additives weight ratio	Unit-less

x	Distance perpendicular to the film surface (measured from film center)	m
y	Distance, normalized by the instantaneous film half-thickness	Unit-less
δ_{dS}	Solvent solubility parameter, dispersion	$MPa^{0.5}$
δ_{hS}	Solvent solubility parameter, hydrogen bonding	$MPa^{0.5}$
δ_{pS}	Solvent solubility parameter, polar	$MPa^{0.5}$
δ_{dP}	Polymer solubility parameter, dispersion	Unit-less
δ_{hP}	Polymer solubility parameter, hydrogen bonding	Unit-less
δ_{pP}	Polymer solubility parameter, polar	Kg / m^3
ϕ_P	Polymer volume fraction	Unit-less
ϕ_s	Solvent volume fraction	Unit-less
ϕ_{se}	Solvent volume fraction in the film external boundaries	Unit-less
ϕ_{s0}	Solvent volume fraction at the beginning of desorption.	Unit-less
ρ_P	Polymer density	Kg / m^3
ρ_S	Solvent density	Kg / m^3
χ	Flory-Huggins solvent-polymer interaction parameter	Unit-less
χ_H	Energy part of the Flory-Huggins interaction parameter	Unit-less
χ_S	Entropy part of the Flory-Huggins interaction parameter	Unit-less
θ	Solvent volume fraction, normalized	Unit-less
ξ	Film thickness, normalized by its initial value	Unit-less
τ	Time, normalized by a characteristic diffusion time	Unit-less

ACRONYMS

ANN	Artificial Neural Network
BRANN	Bayesian Regularization Neural Network
BCUT	Burden, Chemical Abstracts Service, and the University of Texas
CART	Classification and Regression Trees
CAS	Chemical Abstracts Service
CWAs	Chemical Warfare Agents
ED	Euclidean Distance
GA	Genetic Algorithm
GETAWAY	Geometry Topology and Atom-Weights Assembly
IPFS	Integrated Protective Fabric System
MACCS	Molecular Access System
MSE	Mean Square Error
NSRDEC	Natick Soldier Research Development and Engineering Center
PCA	Principal Components Analysis
QSAR	Quantitative Structure Activity Relations
QSPR	Quantitative Structure Property Relations
RMSE	Root Mean Square Error
SMILES	Simplified Molecular Input Line Entry Specification
SLN	SYBYL Line Notation
SDF	Structural Data File
TC	Tanimoto Coefficient
WENDI	Web Engine for Non-obvious Drug Information