J Math Chem (2012) 50:1765–1790
DOI 10.1007/s10910-012-0005-y

ORIGINAL PAPER

A scalable algorithm for molecular property estimation in high dimensional scaffold-based libraries

Sofia Izmailov · XiaoJiang Feng · Genyuan Li · Herschel Rabitz

Received: 22 February 2012 / Accepted: 29 March 2012 / Published online: 20 April 2012

Abstract An algorithm is presented for the estimation of molecular properties over a library built around a scaffold, which has N sites for functionalization with M_i moieties at the ith scaffold site, corresponding to a library of $\prod_{i=1}^{N} M_i$ molecules. The algorithm relies on a series of operations involving (i) synthesis and property measurement of a minimal number of T randomly sampled members of the library, (ii) expression of the observed property in terms of a high-dimensional model representation (HDMR) of the moiety → property map, (iii) optimization of the ordered sequence of moieties on each site to regularize the HDMR map and (iv) interpolation using the map to estimate the properties of as yet unsynthesized compounds. The set of operations is performed iteratively aiming to reach convergence of the predictive HDMR map with as few synthesized samples as possible. Through simulation, the number T of required random molecular samples is shown to scale very favorably with $T << \prod_{i=1}^{N} M_i$ for cases up to $N = 20$ and $M_i = 20$. For example, high estimation quality was attained for simulated libraries with $T \sim 5{,}000$ sampled compounds for a library of 20^{12} members and $T \sim 12{,}500$ sampled compounds for a library of 20^{20} members. The algorithm is based on the assumption that a systematic pattern exists in the moiety → property map provided that the moieties are optimally ordered on the scaffold sites within the context of HDMR. The overall procedure is referred to as the substituent reordering HDMR algorithm (SR-HDMR). The technique was also successfully tested with laboratory data for estimating C^{13}-NMR shifts in a tri-substituted benzene library and for lac operon repression binding.

Keywords QSAR · HDMR · Substituent reordering · Property prediction

S. Izmailov · X. Feng · G. Li · H. Rabitz (✉)
Department of Chemistry, Princeton University, Princeton, NJ 08544, USA
e-mail: hrabitz@princeton.edu

1 Introduction

Accurate estimation of the chemical and physical properties of compounds has applications in a broad range of fields [3,5]. However, synthesis and testing of very large libraries of compounds to seek members with desired properties is often expensive and time-consuming. Computational techniques such as quantitative structure-activity relationships (QSARs) and neural networks [7,13] have been utilized to assist in property estimation over compound libraries. QSAR is a widely used technique for estimating molecular properties based on a set of descriptors serving as independent variables. A variety of descriptors can be employed including octanol-water partition coefficients, molecular surface area, electrostatic properties, molecular structural features, etc. [7,13]. Given a defined set of molecular descriptors, QSAR aims to quantify the relationships between these descriptors and the target property by means of a multi-variable input-output model function. The estimation quality of QSAR-based models rests on the proper choice of descriptors and the type of input-output function. These choices in turn generally depend on the library type and the target property. Since the inherent structure-property relationship is unknown in most applications, the predictive quality of QSAR often requires case-specific selections of descriptors and the input-output functions. Neural networks and other learning-based methods also rely on an input of molecular descriptors for property estimation. With neural networks, a set of "neurons" process the inputs in the form of molecular descriptors to produce an output. The weights given to the different inputs are adjusted in order to ensure that the final output (i.e., the estimated property of interest) is close to the property value for each member of a training set of molecules. Again, a key step is choosing the correct set of molecular descriptors for each particular application. Even a large set of descriptors may not produce good estimated properties if essential descriptors are inadvertently excluded, and overfitting may accurattely capture the descriptor-property relationship for the training set, yet yield poor predictive quality of the test set.

For a library of molecules built around a common scaffold with N functionalized sites, the moieties found at each site form an *inherent* set of molecular descriptors [19,20,24]. Beyond a simple labeling of the moieties at the sites, no extra measurements are needed to determine these descriptors, and they specify each molecule in the library uniquely and completely. In this formulation, the ith site is the independent variable x_i and its "values" are the substituent moieties in that site. In order to be useful for property estimation, the variable values, initially the moiety names, must be converted to a numerical label which can be used to estimate the overall property value. There are many possible choices for labeling the moieties, and one method is to assign the substituents at site i one of M_i discrete, equally-spaced values over some range. A properly assigned ordering of these values for the variables at all of the sites is assumed to produce a smooth property landscape $F(\boldsymbol{x}) = F(x_1, x_2, \ldots, x_N)$, which can be used to estimate the property value of a molecule which has not been tested.

The substituent reordering algorithm [19,20,24] outlined above for property estimation entails (a) randomly sampling a minimal subset of T molecules out of the total number of $\prod_{i=1}^{N} M_i$ library members, (b) measuring the property value of the

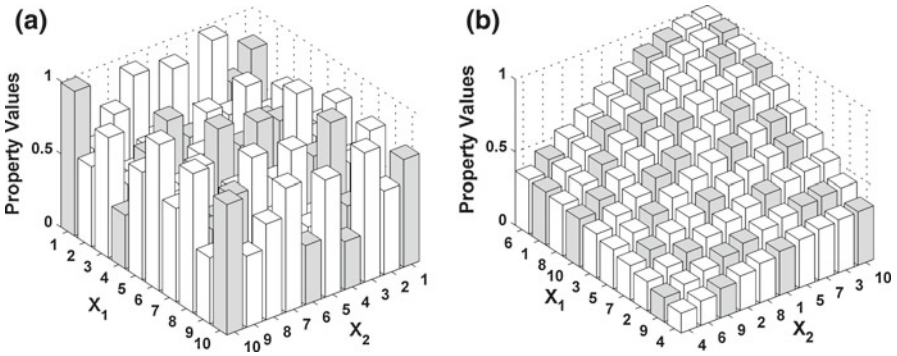

Fig. 1 Comparison between a random and a reordered hypothetical property landscape described by $F(x_1, x_2) = \exp[(-(x_1 - 1)^2 - (x_2 - 1)^2)/81]$. The positive quadrant $x_1, x_2 > 0$ is utilized where each variable is uniformly sampled over [1, 10]. Plot (**a**) shows an arbitrary ordering of the discrete substituents using integer labels for x_1 and x_2. Plot (**b**) shows the result of reordering (note the relabeling along the x_1 and x_2 axes). A coarse sampling of the landscape $F(x_1, x_2)$ is represented by *shaded bars*. The *white bars* are the true, but unmeasured property values of other samples. Under SR-HDMR, the reordering is done just with the data represented by the *shaded bars* [24]. The ordering in (**b**) permits a far better interpolation quality for the unmeasured property values than with the same sampling in (**a**) having a random order

T sampled compounds and (c) using this training set of data to estimate the property of unsampled library members. This procedure corresponds to interpolation over the sparsely sampled library with $T << \prod_{i=1}^{N} M_i$. A key issue in step (c) is the proper ordering of the x_i variable values for all of the M_i substituents at the ith substitution site (e.g., in the case of an amino acid site on a protein scaffold, each of the residues would be assigned an unique integer value from 1 to 20 based on the residue's contribution to the protein property in coordination with residues at other sites) so that the resultant property landscape can provide the best estimation capability for the unsampled compounds. The basic foundations of the molecular ordering algorithm have been applied to scaffolds [19,20,24] with $N = 2$ sites, and this paper will introduce special techniques permitting the efficient treatment of cases for large values of N.

Figure 1 illustrates the reordering algorithm for a hypothetical property landscape $F(x_1, x_2) = \exp[-[(x_1 - 1)^2 + (x_2 - 1)^2]/81]$ with $N = 2$ sites and associated variables x_1 and x_2. A molecular property landscape is inherently discrete, which is represented here by x_1 and x_2 uniformly taking integer values over the range [1, 10]. The order of the integer labels is randomly permuted in Fig. 1a. The consideration of random labeling reflects the fact that initially the moiety → property relationship will not be known and will generally produce an apparently rough property landscape.

In Fig. 1, the shaded bars represent the known $T = 33$ property values for a set of measured samples and the white bars represent the true property values of samples which were not tested (i.e., these are the property values to be estimated). In the landscape of Fig. 1a with random ordering, there are no visible trends, making it impossible to accurately estimate the true property values of the untested samples even from "neighbors" that are available. In the reordered smooth landscape of Fig. 1b, based on just using the T samples, it is possible to estimate the property value of an

untested compound utilizing the tested compounds near it. Translation of this procedure into a molecular discovery algorithm [19,20,24] rests on the assumption that a well-defined library of compounds will have a smooth property function $F(\boldsymbol{x})$ upon proper identification of the site index ordering. This assumption has been verified and the reordering strategy has been successfully implemented in diverse applications, including estimation of the glass transition temperatures [24] in copolymers, photoluminescent quantum yields and emission energies of transition metal complexes [19], and inhibitor efficacies of a pharmaceutical library [20]. Importantly, the algorithm uses local interpolation, thus a specific global form for $F(\boldsymbol{x})$ does not need to be specified.

While these previous applications of the substituent reordering procedure were very successful in achieving reliable property estimation, all the compound libraries had $N = 2$ substitution sites. When the dimensionality N of the library increases, the reordering and property estimation algorithms used for these low-dimensional libraries do not scale well in terms of the sampling effort for reliable operation. The original work [24] speculated that property estimation could be performed efficiently for high dimensional libraries, and this paper presents a practical procedure to accomplish this task. We will integrate the high dimensional model representation (HDMR) technique [17] with the substituent reordering method to address the need for efficient molecular discovery in high dimensional scaffold-based libraries (e.g., for proteins). HDMR is a general procedure for nonlinear high-dimensional data analysis, data-driven modeling, and property interpolation. It operates by decomposing a high-dimensional input $\rightarrow$ output function into a hierarchy of lower-dimensional, generally nonlinear HDMR component functions. The HDMR decomposition is the central factor enabling favorable sampling scalability and high efficiency for property estimation with increasing N. The HDMR decomposition has been successfully used to model systems with large numbers of input variables [25,26]. However, these applications did not require reordering of the input variables. This paper exploits the hierarchical component function breakdown in HDMR to manage the nominally difficult reordering problem in high dimensions.

The relationship between the property $F(\boldsymbol{x})$ and the substituents $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ for the training set of T compounds is decomposed by HDMR into a family of lower-dimensional component functions of one variable, two variables, etc. The substituent reordering technique is then applied to each of the HDMR component functions to *separately* obtain their respective optimal substituent ordering. The sum of these optimally ordered component functions constitutes the map of the contributions of the substituents at the N sites to the property, and this map specifies the N-dimensional landscape used for subsequent property interpolation (estimation). In this fashion HDMR permits the identification of a family of optimal site moiety orderings with each tuned to the various site-site interaction contributions to the property. As a result, the integration of HDMR with the substituent reordering procedure produces a scalable technique for molecular property estimation and optimization in high-dimensional libraries. The scalable character of the algorithm arises because each of the T randomly sampled training set members corresponds to a point in the N dimensional moiety space which projects into *each* of the low dimensional HDMR component functions. In this fashion good moiety coverage is provided, despite T being much

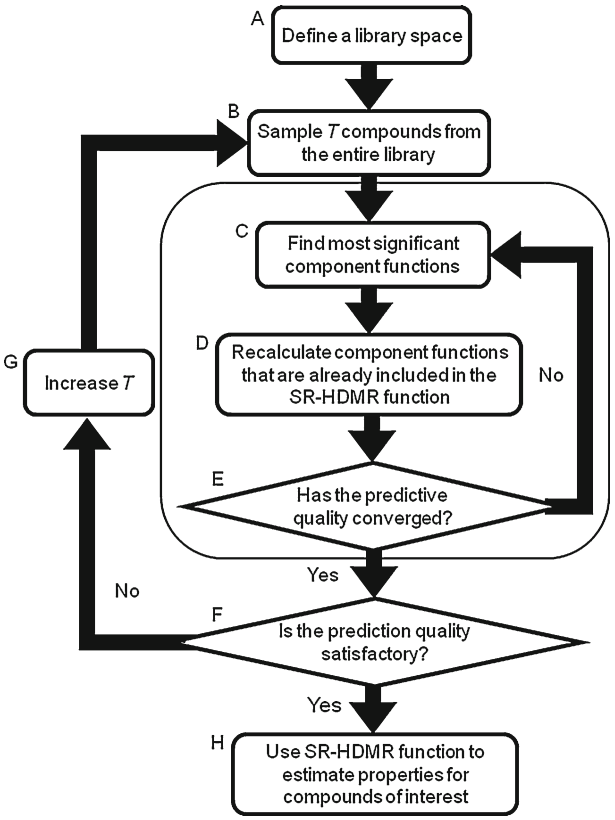

Fig. 2 Summary of the SR-HDMR procedure. In step *A*, the library is defined by choosing the scaffold, sites, and substituents. In step *B*, a set of T compounds from the library are synthesized in the laboratory and their property values are measured forming a training set to determine the HDMR expansion. The operations *C*, *D*, and *E* enclosed in the square generate the SR-HDMR for the property being estimated. In step *C*, the component functions which are not yet included in the HDMR approximation are searched over to determine the most significant possible new members. Each candidate component function has its optimal ordering determined and then expressed in terms of basis functions. The process starts with first order component functions, and when no further first order functions are significant then the second order component functions are considered, etc. Subsequently, backfitting of the component functions occurs in step *D*. In this step every component function in the HDMR approximation is removed one at a time, recalculated (with optimal reordering and refitting), and included in the HDMR expansion if it is still significant. In step *E*, the procedure repeats if the estimation quality has not converged. If the estimation quality for the training set, as measured by r^2 or some other metric, is not satisfactory in step *F* once it has converged, the training set is increased in step *G*. The process is then restarted with a larger portion of the library. If the predictive quality for the training set is satisfactory, then step *H* uses the SR-HDMR as a predictive map for estimation of the property value of other as yet unsynthesized library members

smaller than the overall library size. Figure 2 shows the general operational steps of the integrated technique, which we refer to as the Substituent Reordering HDMR (SR-HDMR) algorithm.

To illustrate the capabilities of SR-HDMR, the procedure will be applied to estimate C^{13} NMR shifts for a trisubstituted benzene library (i.e., $N = 3$ variables and

library size 10^3) and for lac operon repression with $N = 4$ and a library of size 6,400. In order to fully evaluate the capabilities of the SR-HDMR technique for handling high-dimensional libraries, we also performed tests using several types of simulated property landscapes for up to $N = 20$ substitution sites and $M_i = 20$ substituents on each site i. The results show that sampling ~15,000 compounds (out of 20^{20} possible library members) is sufficient to achieve excellent estimation quality in a broad variety of tests. These collective illustrations demonstrate that integration of HDMR and substituent reordering provides a scalable and generally applicable SR-HDMR strategy for molecular discovery in high-dimensional scaffold-based libraries.

Section 2 explains the SR-HDMR procedure. Section 3 presents illustrative applications of SR-HDMR to estimate C^{13} NMR shifts, lac operon repression, and the properties of large simulated libraries. Section 4 provides concluding remarks. The Appendices A, B, and C present additional details on the procedure described in Sect. 2.

2 The SR-HDMR algorithm

The SR-HDMR strategy melds together substituent reordering and HDMR. A synopsis of both techniques is presented here with additional details given in the Appendices.

The perspective adopted with SR-HDMR is to (i) minimally sample T members of a library and then (ii) estimate the properties of the *entire* remainder of the yet unsynthized memebers. The overall gain can be dramatic with $T << \prod_i^N M_i$. However, an even further reduction in sampling would likely be afforded by directly appling suitable optimization algorithms seeking to find *one* library memeber with favorable properties. With SR-HDMR aiming to capture the entire property landscape, all good library members may be identified.

2.1 High dimensional model representation

HDMR provides a mapping $\boldsymbol{x} \rightarrow F$ for functions $F(\boldsymbol{x}) = F(x_1, x_2, \ldots, x_i, \ldots, x_N)$ with the specific goal of interpolating over $F(\boldsymbol{x})$ from a known coarsely sampled set of input values $\boldsymbol{x}^r$, $r = 1, 2, \ldots, T$ and the associated observed outputs $F(\boldsymbol{x}^r)$. Each of the independent variables $x_1, x_2, \ldots, x_i, \ldots, x_N$ of the function is assumed to take on a finite range of values that can then be scaled to the [0, 1] domain. The function $F(\boldsymbol{x})$ is decomposed by HDMR into a sum of lower dimensional components [17]:

$$F(\boldsymbol{x}) = f_0 + \sum_{i=1}^{N} f_i(x_i) + \sum_{1 \le i < j \le N} f_{ij}(x_i, x_j) + \cdots + f_{12\ldots N}(x_1, x_2, \ldots, x_N) \quad (1)$$

where each of the component functions represents the unique contribution of its variables to the value of the property $F(\boldsymbol{x})$: f_0 is the base contribution which is independent of the values of the $\boldsymbol{x}$ variables, $f_i(x_i)$ is the contribution of substituents at site i on the scaffold acting alone, $f_{ij}(x_i, x_j)$ is the cooperative contribution of substituents at sites i and j, etc.

The component functions of the HDMR decomposition can be expressed as:

$$f_0 = \frac{1}{\prod_{k=1}^{N} M_k} \sum_{x_1=1}^{M_1} \cdots \sum_{x_k=1}^{M_k} \cdots \sum_{x_N=1}^{M_N} F(\boldsymbol{x}) \tag{2a}$$

$$f_i(x_i) = \frac{1}{\prod_{k=1,k\neq i}^{N} M_k} \left(\sum_{x_1=1}^{M_1} \cdots \sum_{x_{i-1}=1}^{M_{i-1}} \sum_{x_{i+1}=1}^{M_{i+1}} \cdots \sum_{x_N=1}^{M_N} F(\boldsymbol{x}|x_i) \right) - f_0 \tag{2b}$$

$$f_{ij}(x_i, x_j) = \frac{1}{\prod_{k=1,k\neq i,j}^{N} M_k} \left(\sum_{x_1=1}^{M_1} \cdots \sum_{x_{i-1}=1}^{M_{i-1}} \sum_{x_{i+1}=1}^{M_{i+1}} \cdots \sum_{x_{j-1}=1}^{M_{j-1}} \sum_{x_{j+1}=1}^{M_{j+1}} \cdots \right. \tag{2c}$$
$$\left. \cdots \sum_{x_N=1}^{M_N} F(\boldsymbol{x}|x_i, x_j) \right) - f_i(x_i) - f_j(x_j) - f_0$$

Here we have used the notation $F(\boldsymbol{x}|x_i)$, $F(\boldsymbol{x}|x_i, x_j)$, etc to clarify the operations involved. The expression $F(\boldsymbol{x}|x_i)$ is the property value of a sample which has substituent x_i fixed at site i. The remaining sites $k \neq i$ variables can take on all of their M_k values. Similarly, $F(\boldsymbol{x}|x_i, x_j)$ is the property value when the substituents x_i, x_j are fixed (at the respective ith and jth sites) and the remaining sites with $k \neq i, j$ can again take on any of their M_k values. Similar notation would apply to higher order terms as well. The component functions also satisfy:

$$\sum_{l_i=1}^{M_i} f_i(x_i^{l_i}) = 0 \tag{3a}$$

$$\sum_{l_i=1}^{M_i} f_{ij}(x_i^{l_i}, x_j) = 0 \tag{3b}$$

$$\sum_{l_j=1}^{M_j} f_{ij}(x_i, x_j^{l_j}) = 0 \tag{3c}$$

$$\cdots$$

where $x_i^{l_i}$ is the l_ith substituent in the ith site and the sums are over all such substituents. The HDMR formulation in Eqs. (3)a–c assures that the component functions f_i, f_{ij}, ... are orthogonal to each other [18]. In practice, the entire set of library members will be unavailable in accord with the overall goal of estimating the properties of missing members. Under these conditions, the component functions in the HDMR expansion are determined using a random sample of T compounds from the entire library. The details of how this sample is used to determine the component functions are discussed in Appendix A.

The HDMR expansion of $F(\boldsymbol{x})$ taken to the Nth order in Eq. (1) is always an exact [17] representation of $F(\boldsymbol{x})$. In most realistic applications, the HDMR component functions up to the second or third order are typically sufficient to quantitatively

describe the input → output relationship [18] $\boldsymbol{x} \rightarrow F$. In particular, it is expected that $F(\boldsymbol{x}) \approx f_0 + \sum_{i=1}^{N} f_i(x_i) + \sum_{1 \le i < j \le N} f_{ij}(x_i, x_j)$ often should be adequate for the representation of molecular properties, as this coincides with the statement that up to pairwise cooperative interaction between substituents at different sites should be sufficient. In this fashion, HDMR reduces the initial function of N variables to a set of component functions of less than three variables. Importantly, each of these component functions now resembles a property landscape analogous to the low dimensional scaffold-based libraries treated previously without HDMR [20,24].

The HDMR expansion provides the basis for an effective interpolator over the full library of compounds, provided that (i) an efficient means can be established to determine the low order component functions from a minimal sampling of T library members and (ii) the optimal reordering of the variable values $x_i^{l_i}$, $i = 1, 2, \ldots, N$ with $l_i \in [1, M_i]$ can be performed effectively for each component function. These steps are described below.

2.2 Optimal reordering of variables and the determination of individual component functions

This paper considers the SR-HDMR strategy at the level of first and second order, involving the component functions $f_i(x_i)$ and $f_{ij}(x_i, x_j)$, respectively, although the logic involved can be taken to any order. While a compound in the library is specified by a point $\boldsymbol{x}$ in the full N-dimensional moiety space, at the component function level of $f_i(x_i)$ or $f_{ij}(x_i, x_j)$ the compound is reflected in the value of x_i and the pair x_i, x_j, respectively, in the reduced dimensional space. The goal is to reliably determine the set of all relevant HDMR component functions and render them as smooth as possible by variable reordering for finally interpolating over the library to estimate the property value of as yet unsynthesized compounds.

The process of determining a component function involves (a) establishing an initial approximation for the component function, (b) reordering the substituents based on the initial approximation, and (c) expressing the reordered initial approximation in terms of an expansion in suitable basis functions to provide a final approximation which can be used in the SR-HDMR expansion. This process can be iterated as needed. A summary of these operations is given here (See Appendix A for details). The initial approximation for a group of substituents, such as (x_i, x_j) for $f_{ij}(x_i, x_j)$, is determined by calculating the unordered property contribution of samples from the training set of size T with substituents x_i and x_j using Eqs. (2)a–c. In doing so the factors $\prod_{k=1}^{N} M_k$, $\prod_{k=1,k\neq i}^{N} M_k$, etc. are replaced by the number of samples of type $(\boldsymbol{x}|x_i)$, $(\boldsymbol{x}|x_i, x_j)$, etc. The substituent reordering is then performed with the goal of rearranging the substituent labels so that the resulting component function $f_{ij}(x_i, x_j)$ is as smoothly varying as possible over (x_i, x_j) which makes it a reliable interpolator. Each of the component functions $f_{ij}(x_i, x_j)$ will have its own unique ordering of site moieties even if another function $f_{ik}(x_i, x_k)$ involves the same site i (i.e., each pair of variables has its own unique contribution to the property F).

As with low dimensional scaffold-based libraries [19,20,24], the low dimensional component functions $f_i(x_i)$, $f_{ij}(x_i, x_j)$, etc. may be fitted with basis functions. This

work utilizes a basis of cubic B-spline functions; other choices could be employed including simple nearest neighbor interpolation. The details of the initial approximation of the component functions, the reordering operation, and the fitting procedure are given in Appendix A.

2.3 HDMR integrated with substituent reordering

In any particular application involving a library of compounds with a large value for N, not all of the first and second order HDMR functions are likely to be physically significant (i.e., certain substituents and their interactions may dominate over others). Retaining only the significant component functions is important to avoid overfitting and thereby enhancing the quality of the property estimation. A statistical F-test [4] with 99 % confidence level was used to determine whether a component function is actually significant in the SR-HDMR expansion.

In this fashion, the component functions are incorporated into the SR-HDMR approximation one at a time. After a new component function has been added, the component functions that were already included are recalculated. The new component functions are affected by the component functions which are already present in the HDMR expansion (and *vice versa*; see Eqs. (2)a–c) so this recalculation enhances estimation quality by improving the accuracy of the previous and newly included component functions. The cycle of backfitting (i.e., recalculating the component functions already included in the HDMR) with inclusion and exclusion of component functions based on their significance continues until an adequate fidelity for the training set is reached or until the SR-HDMR $\boldsymbol{x} \rightarrow F$ map converges. If the map has converged, but the estimation quality is inadequate, then the training set size T can be increased and the process repeated. Figure 2 and the accompanying caption give an overall summary of the SR-HDMR algorithm for property estimation over high dimensional libraries, and a more detailed explanation of the inclusion/exclusion process and backfitting is given in Appendix B. The means for testing the significance of the component functions is explained in Appendix C.

3 Illustrations

The SR-HDMR procedure described in Sect. 2, the Appendices and Fig. 2 was tested in two applications with laboratory data as well as with simulated data. With laboratory data SR-HDMR was used to (1) estimate C^{13}-NMR shifts of a trisubstituted benzene library (three scaffold sites) and (2) describe the interaction between a mutant lac repressor and the lac operon (four scaffold sites). Since these cases have relatively low dimensional libraries, we also performed simulations of SR-HDMR for libraries with larger numbers of substituents and substitution sites (i.e., libraries with up to 20^{20} members corresponding to 20 sites and 20 moieties per site).

In all of the illustrations, the reliability of the SR-HDMR procedure is quantified by calculating the squared correlation coefficient r^2 between the true and estimated property values. In practical laboratory operation SR-HDMR would be executed iteratively as indicated in Fig. 2. In this fashion the r^2 value would be available upon cyclic

operation of property estimation and testing with newly synthesized compounds. In the present work we employ existing libraries of laboratory data or simulated data libraries.

The examples in Sects. 3.1 and 3.2 on C^{13} NMR shifts and lac operon repression, respectively, naturally include experimental noise from the laboratory data. The high dimensional model cases in Sect. 3.3 will first be tested with no noise to better understand the scaling behavior of the SR-HDMR procedure with respect to dimension N and training set size T. Section 3.3.3 presents an assessment of the impact of noise in high dimensional cases.

3.1 Estimation of C^{13} NMR chemical shifts

Using an available database [1], we collected the C^{13} NMR shifts for the 1-carbon in a series of 1, 2, 4-trisubstituted benzene molecules. At each of the three substitution sites there are ten possible functional groups: –H, $–CH_3$, $–NH_2$, –OH, $–COCH_3$, –CHO, –Br, –Cl, –F, $–OCH_3$. The shifts were measured in $CHCl_3$ solvent with a TMS standard. Only those samples with an assigned carbon shift for the 1-carbon were considered. When multiple entries were found for the C^{13} shifts in a single compound, their chemical shifts were averaged together. Additional steps were taken if multiple entries existed with different peak assignments for the 1-carbon (i.e., indicating a mistake in peak assignment in at least one of the entries). In these instances, the 1-carbon peak assignment was based on the plurality of entries whose 1-carbon shifts were given a single value. Samples without a plurality of 1-carbon assignments were not considered. Not all of the possible 1,000 library compounds were present in the database, but using these criteria, 428 compounds were available from the database. The maximum number of B-spline knots permitted for the component functions was $m = 5$ (See Appendix A).

Given the limited number of samples, we performed five trials using randomly chosen training sets with each having 200 samples. The average r^2 was 0.991 ± 0.006. As a comparison, we repeated the HDMR calculations where the ordering of the substituents for each HDMR component function was *randomly* assigned. The average r^2 for these latter trials was $r^2 = 0.853 \pm 0.129$. The predictive results for a typical test set and training set with optimal and random ordering trials is shown in the truth plots of Fig. 3. The figure clearly shows a significant improvement in property estimation quality when the optimal ordering is used in SR-HDMR. The truth plot achieved using the optimal ordering has very high predictive quality for both the training and test sets.

3.2 Estimation of lac operon repression

A library of modified lac operons and repressors is available [14] where the amino acid residues 1 and 2 in the repressor recognition helix and base pairs 4 and 5 of the symmetric operon were varied. Out of a possible $20 \times 20 \times 4 \times 4 = 6{,}400$ combinations, 1,288 were experimentally tested. The repression value, given as $1 + K(C)$ for the equilibrium binding constant K (as a function of the repressor concentration C), was determined by measuring the concentration of β-galactosidase synthesized using

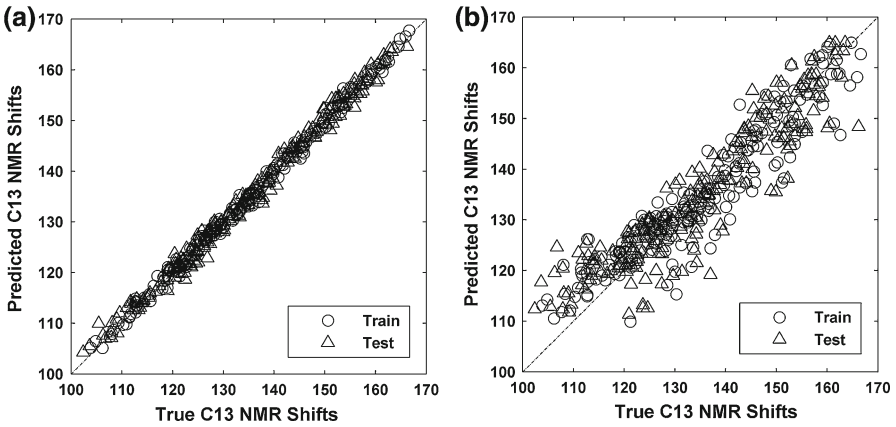

Fig. 3 **a** The truth plot of the test and training sets for estimation of C^{13} NMR shifts determined with *optimal* substituent ordering using SR-HDMR. **b** The truth plot of the test and training set estimations of C^{13} NMR shifts determined with *random* substituent ordering. The HDMR functions for both cases were determined using a training set of 200 samples and a test set of 228. The importance of using proper ordering in SR-HDMR is clear upon comparing (**a**) and (**b**)

the lac Z reporter gene. Repression values below 4 are the least accurate and the β-galactosidase measurements are estimated to have a relative error of $\pm 20\,\%$ [15]. The laboratory sampling was also not random; while all combinations of nucleotides are represented, many amino acid combinations are not. Certain amino acid residues are more frequently represented for a given site, while others are rarely sampled. Although this data set is not ideal for SR-HDMR due to its under-represented, non-random sampling and low data accuracy, we still performed an analysis to evaluate the capabilities of the integrated technique in such adverse conditions.

A slight modification of the normal SR-HDMR procedure was required. In the previous C^{13}-NMR example, each site had the same number of possible substituents. However, in this case there are different numbers of substituents for the four sites. The number of spline function knots for the nucleotide sites (only 4 substituents) is set to $m = 1$ because there are not enough substituents to use more knots. The number of knots for the amino acid sites ranges from $m = 1$ to $m = 5$. For the second order HDMR component functions involving nucleotide and amino acids sites, the spline functions are also similarly adjusted; rather than utilizing $m \times m$ splines, we employed $1 \times m$ splines.

Because the training set was so sparsely and unevenly sampled, a different reordering algorithm was used for the second order component functions. A genetic-algorithm was employed to determine the order as described previously [24]. In this process, a set of 100 random pairs of i and j orderings was generated; for each pair of ordered variables, a component function was generated. The i, j orderings that gave the 10 worst component functions (i.e. the ones that fit the property values the worst) were discarded and replaced with new orderings generated from the 90 remaining ordering pairs. This was done by weighted random selection of two of the 90 pairs to generate each new ordering. The weights were given as the inverse RMS distance between the

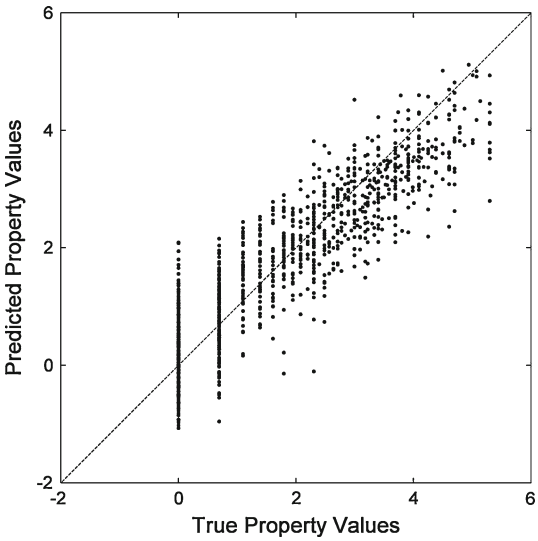

Fig. 4 The truth plot of the estimated ln(repression) values for combinations of lac operons and repressors which were each varied at two sites. The plot shows the quality of SR-HDMR function for regenerating the ln(repression) values which were used to create it. The results are quite satisfactory given the dynamic range of the data and its significant error

true property values and the property values were estimated by a spline fitting function. The new i and j orderings are randomly taken from either parent. With a 0.1 probability for either of these orderings, a mutation occurs [24] to change the order slightly. The process of removing the poorest 10 ordering pairs and generating new orderings continues until the algorithm fails to improve on the best possible ordering pair for 500 generations. This best ordering of a pair of variables is utilized in the component function that is ultimately considered for the HDMR model.

The output was taken as $F = \ln(1 + K(C))$, and Fig. 4 shows a plot comparing the true values with the estimated values. This plot uses SR-HDMR to estimate the values of the same samples which were used to generate the HDMR (i.e., the training set). Trials run without reordering failed to identify any significant component functions (all compounds had an estimated value equal to f_0) and therefore could not be used for property estimation (i.e., describing the training set in this circumstance). It is clear from the results that in this case the estimation quality for the training set is lower than the estimation quality in the NMR case. However, the results are quite good considering the high uncertainty of the input values and the biased non-random sampling.

3.3 High dimensional simulated data sets

To better evaluate the scalability of the SR-HDMR algorithm, simulated high-dimensional data sets were generated using multi-variable input $\rightarrow$ output functions $H(\boldsymbol{x})$; the simulated trial and test data are denoted as H to distinguish from the SR-HDMR estimate F. Two types of libraries were used to illustrate the SR-HDMR procedure with property estimation under various conditions. In the first library, the property function landscape $H(\boldsymbol{x})$ has a Gaussian form in the N variables. Such simply

Table 1 Average r^2 for the Gaussian property training sets

Training set size N	1,000	2,000	3,000	5,000
8	0.994 ± 0.001	0.998 ± 0.001	0.999 ± 0.000	–
12	0.996 ± 0.001	0.997 ± 0.000	0.998 ± 0.000	–
16	–	–	–	1.00 ± 0.00

structured landscapes are only coarsely sampled in SR-HDMR and lie in high dimensions. More complex landscapes were also considered in a family of cases where the property function $H(\boldsymbol{x})$ is in the form of an HDMR consisting of first and second order terms whose polynomial functions were generated independently of each other. The landscapes in the latter cases are smooth (i.e., with properly ordered variables for each component function) but highly complex over $\boldsymbol{x}$. In all cases the substituent ordering was randomized *before* the data was used for testing. Consequently, the SR-HDMR algorithm is not "aware of" the true underlying functions, even when these functions were generated by another HDMR expression. The initial tests in Sects. 3.3.1 and 3.3.2 will be free of input noise, while Sect. 3.3.3 will explore the impact of noise on the estimation quality.

3.3.1 Simply structured property landscape

This case tests the effectiveness of SR-HDMR on simulated high dimensional data sets when the property landscapes $H(\boldsymbol{x})$ has a simple structural form. Importantly, landscape simplicity or monotonicity itself is not a requirement for SR-HDMR. The coarsely sampled and randomly ordered input data nominally hides the original regularity, and these cases provide a test of SR-HDMR's ability to identify the true simple character of the $\boldsymbol{x} \rightarrow H$ relationship.

In this case, the property function is generated as an N-dimensional Gaussian $H(\boldsymbol{x}) = \exp(-\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x})$ where $\boldsymbol{A}$ is a random $N \times N$ positive definite matrix with the largest eigenvalue being 0.5. Each of the N sites had 20 substituents and their "true" orderings corresponded to the variables being equally spaced and sequentially labeled on $[0, d]$ where d is a constant specifying the size of the hypercube $[0, d]^N$. For the $N = 8, 12$, and 16-D Gaussians, d was set to 0.50, 0.475, and 0.45, respectively. These values were chosen to ensure that essentially the same range of property values was considered for the different dimensional Gaussians in order to make comparative tests with respect to N.

For each value of N, three random matrices $\boldsymbol{A}$ were chosen. Each $\boldsymbol{A}$ was used to generate a training set of size $T = 1{,}000, 2{,}000$, and 3,000 and a test set of size 10,000 (there were three training and test sets for each N and T). The $N = 16$ case was tested for three random matrices $\boldsymbol{A}$ which were used to generate training sets of size $T = 5{,}000$. The average r^2 values for the training sets are given in Table 1 with standard deviations determined from the multiple data sets.

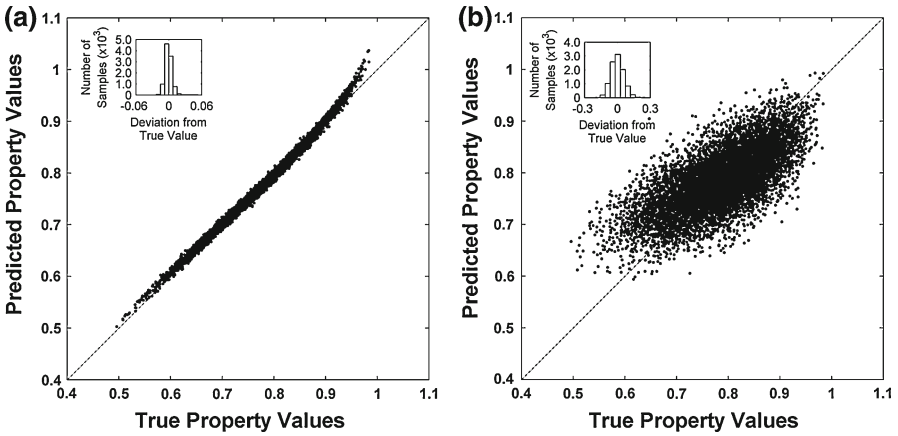

Fig. 5 The truth plot of property estimation using a 8-D Gaussian function to represent a scaffold with 8 sites and 20 substituents per site. **a** The truth plot of property estimate for optimally ordered substituents in SR-HDMR. **b** The truth plot for random orderings of the substituents. Inserts show the distribution of the differences between the true and estimated property values; note the change in scale along the abscissa. These results represent the estimation quality for the test set when a training set of $T = 1{,}000$ is used to determine the HDMR function. The SR-HDMR in (**a**) with reordering shows dramatically improved estimation quality over the case of random ordering in (**b**)

As a typical illustration, Fig. 5 shows the truth plots for a test set with and without reordering created for a $N = 8$ case and a training set of $T = 1{,}000$. The estimation quality is excellent and the r^2 for the optimally ordered trial is 0.994 while for the randomly ordered trial it is 0.483. One noticeable feature is a slight tendency to give overestimated property values for $H(\boldsymbol{x}) \gtrsim 0.9$. Samples with property values greater than $\sim$0.9 make up less than 1 % of the library, and the modest number of random samples did not cover this region very thoroughly.

3.3.2 Highly structured property landscapes

While the Gaussian examples above generated a globally regular property function upon proper ordering of the variables, the SR-HDMR method only requires smooth overall behavior of $H(\boldsymbol{x})$. Thus, as a further test we chose property functions $H(\boldsymbol{x})$, as a sum of randomly generated component functions, whose overall behavior could be quite complex. Each property function had the form $H(\boldsymbol{x}) = h_0 + \sum_{i=1}^{n} h_i(x_i) + \sum_{1 \le i < j \le n} h_{ij}(x_i, x_j)$ where $h_0 = 0$, $h_i(x_i) = a_i \varphi(z_i)$, $h_{ij}(x_i, x_j) = b_{ij} \varphi(z_i)\varphi(z_j)$. Here, the relation between x_i and z_i is through their ordering, as explained below. The function φ is a shifted Legendre polynomial on [0, 1]

$$\varphi(z_i) = \sqrt{3}(2z_i - 1)$$

with weights a_i and b_{ij} prescribing the significance of the different contributions of the component functions. The z-values are discrete and equally spaced between 0 and 1, inclusive for each component function. To form the substituent values for x_i in each component function the order of z_i is randomized. Although this property function has a seemingly simple form, globally $H(\boldsymbol{x})$ is quite complex and not separable in

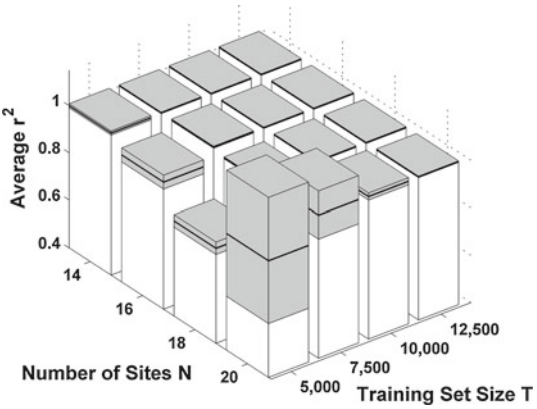

Fig. 6 The property estimation quality (expressed in terms of r^2) as a function of the number of substituent sites N and the training set size T. The two *shaded bars* stacked on top of each *white bar* represent ±two standard deviations for the average r^2 for the given training set size and number of sites. The average r^2 is shown as a *solid line* between the two *shaded bars*. Each average r^2 was generated using a trial from three different libraries. The average r^2 significantly improves (i.e., gets larger) as the number of samples increases for any number of sites. Very small training set sizes of ≈12,500 give excellent property estimation quality for libraries up to 20^{20} members

the contributions of any of the variables. The choice of component functions satisfies the definitions given in Eqs. (3)a–c, thus ensuring that h_0 actually corresponds to the average property value of the entire library.

As specified above, a random ordering was assigned to the substituents for each site in any component function. Thus, the original ordering of the substituents for a given site is different for each component function. Physically, this reflects the fact that the substituents at site i may contribute differently to the overall property function when acting alone versus when they are jointly contributing with another site j. Therefore, the optimal ordering at site i depends on which component function is being considered. This behavior is accounted for in the SR-HDMR algorithm of Fig. 2.

The a_i coefficient for each first order component function $h_i(x_i)$ is a pseudorandom number generated from a Gaussian distribution centered at zero with a standard deviation of 3, and the b_{ij} coefficient for each second order component function $h_{ij}(x_i, x_j)$ is similarly generated with a standard deviation of 1. Only a_i's and b_{ij}'s between -10 and 10 were allowed. This choice of standard deviations simulates the general expectation that (i) first order component functions will likely contribute more significantly than second order component functions, (ii) most components contribute only in a small way to the overall property value, and (iii) there are a few sites or site pairs that can significantly affect the property.

The number of substituents M at each site was set to 20 for all the simulations and the following number of sites were considered: $N = 4$, 8, 12, 14, 16, 18, and 20. For each library of dimension N, training sets of size $T = 5{,}000$, 7,500, 10,000, 12,500, and 15,000 were used and test sets were always of size 10,000. In this collective fashion a broad variety of library and training set sizes were generated to assess the quality of the SR-HDMR procedure. The average r^2 results of simulations with $N = 12$, 14, 16, and 18 sites are plotted in Fig. 6. As expected, the larger training sets increase the

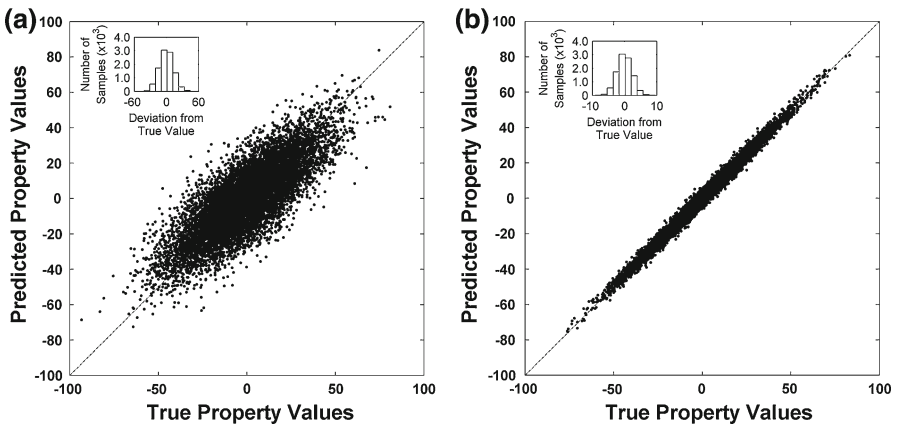

Fig. 7 The truth plots of property estimation for a test set from a library of samples functionalized at 20 sites with 20 substituents per site. Panel (**a**) uses a training set of $T = 5{,}000$ and panel (**b**) uses a training set of $T = 10{,}000$. The inserts show the distribution of the differences between the true and estimated property values; note the change in scale along the abscissa. A dramatic improvement occurs for a modest increase in the training set size, especially considering that the overall library has 20^{20} members

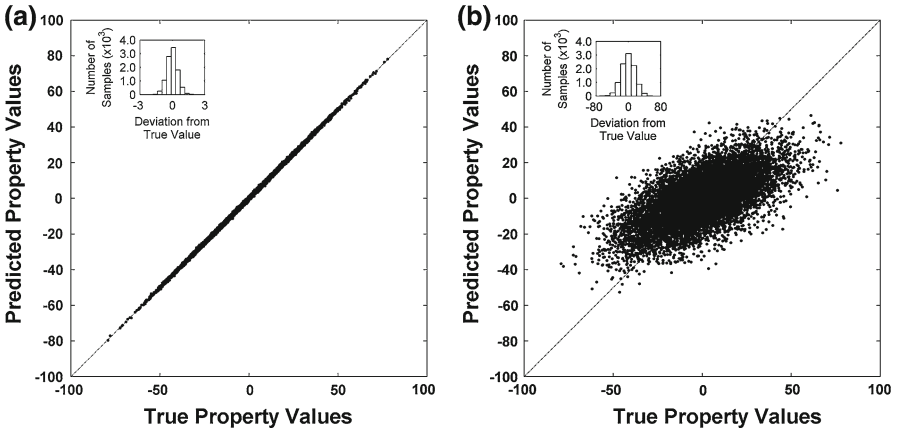

Fig. 8 The truth plots of property estimation for a test set using a training set of $T = 12{,}500$ sampless from a library of members functionalized at 20 sites with 20 substituents per site (**a**) with optimal ordering and (**b**) with random ordering. The significance of employing optimal ordering is clear upon comparing (**a**) and (**b**)

property estimation quality. The truth plots for estimation quality for 20 sites and training set sizes $T = 5{,}000$ and $T = 10{,}000$ are given in Fig. 7. The estimation quality improves greatly for the modest increase in sample size compared to the overall library size of 20^{20}. Figure 8 shows the estimation quality for 20 sites with a training set size $T = 12{,}500$ created with and without substituent reordering. This result clearly demonstrates the necessity of reordering to obtain good quality property estimation. The histogram inserts in Figs. 7 and 8 show the distribution of differences between the

Table 2 Average $r^2_{\text{noisy-estimated}}$ for noisy data

Training set size $r^2_{\text{exact-noisy}}$	5,000	10,000	15,000	20,000	25,000
0.9	0.606 ± 0.056	0.770 ± 0.003	0.820 ± 0.004	0.840 ± 0.000	0.858 ± 0.011
0.95	0.710 ± 0.036	0.866 ± 0.003	0.902 ± 0.007	0.919 ± 0.002	0.926 ± 0.003

Table 3 Average $r^2_{\text{exact-estimated}}$ for noisy data

Training set size $r^2_{\text{exact-noisy}}$	5,000	10,000	15,000	20,000	25,000
0.9	0.666 ± 0.062	0.847 ± 0.010	0.900 ± 0.005	0.925 ± 0.004	0.943 ± 0.005
0.95	0.746 ± 0.041	0.910 ± 0.006	0.947 ± 0.006	0.964 ± 0.003	0.973 ± 0.003

real and estimated property values indicating the dramatic improvement with sample size and reordering, respectively.

3.3.3 Highly structured property landscapes with Gaussian noise

The trials in Sect. 3.3.2 only considered data without noise. When Gaussian noise is added to the data, the estimation quality improves more slowly with increasing training set size than without added error. In order to simulate equivalent levels of Gaussian noise in different libraries, the standard deviation of the noise added to the library was selected such that the value of $r^2_{\text{exact-noisy}}$ between the initial exact data and noisy data was approximately constant over a series of simulations. This method was used to produce six noisy $N = 16$ libraries from the three libraries used in Sect. 3.3.2. For each of the exact libraries, one noisy library with $r^2_{\text{exact-noisy}} = 0.95$ and one noisy library with $r^2_{\text{exact-noisy}} = 0.90$ was created. For each new library, training sets of size $T = 5{,}000$, 10,000, 15,000, 20,000, and 25,000 were generated. Test sets of 10,000 were utilized for each training set. The training sets were used to generate a SR-HDMR, which was employed to estimate the property value for the test set.

Property estimation quality was measured in two different ways. The squared correlation coefficient $r^2_{\text{noisy - estimated}}$ was calculated to assess the relationship between the true and estimated property values. Additionally, $r^2_{\text{exact-estimated}}$ was calculated to assess the relationship between the initial exact data and the estimated property values. This latter statistical measure is important because the upper bound for this value is 1.0. The SR-HDMR procedure, should ideally *not* fit the random noise, and thus the best possible estimated property values would be equal to the noise-free exact property values. $r^2_{\text{noisy-estimated}}$ alone does not give information about whether a better HDMR is theoretically possible. Tables 2 and 3 show the average $r^2_{\text{noisy-estimated}}$ and $r^2_{\text{exact-estimated}}$.

As expected, for each test case a comparison between Tables 2 and 3 shows that $r^2_{\text{noisy-estimated}}$ is less than $r^2_{\text{exact-estimated}}$. Thus, the SR-HDMR procedure is effectively filtering out some level of noise in processing the training set data. A comparison of the

behavior in Tables 2 and 3 with the noise-free case of $N = 16$ in Fig. 6 illustrates the need for larger T to attain comparable results when noise is significant. Nevertheless, the training set sizes required for noisy and idealized data are similar, so the increase is modest to accommodate noisy data. For example, a training set of size $T \sim 20{,}000$ is large, but it is a small fraction of the total library of size 20^{16}. The results indicate that the SR-HDMR algorithm can be applied to high dimensional libraries with reasonable levels of measurement uncertainty.

4 Conclusion

This paper combined the HDMR technique with optimal substituent reordering (SR) to provide a robust SR-HDMR algorithm for molecular property estimation in libraries with a common scaffold and multiple substitution sites. SR-HDMR decomposes the overall property into a hierarchy of many low dimensional contributions which represent the individual and cooperative roles of the substituents located at different scaffold sites. The simulations in Sect. 3.3 indicate that the required sampling effort of SR-HDMR scales very favorably with respect to the number of the substitution sites and overall library size.

The results utilizing laboratory data sets were of good quality for two diverse properties (NMR shifts and protein repression). The treatment of the repression data showed that the quality of the training set data and the presence of biased non-uniform sampling of the library can impact SR-HDMR estimation fidelity. These laboratory data tests were only 3- and 4-D, so simulated data for higher dimensions was used to better assess the scalability of the SR-HDMR algorithm. While the library sizes ranged from 20^{12} to 20^{20}, the size of the training set necessary to attain excellent estimation quality remained at $\sim$15,000 or smaller if the data had a low level of noise. The SR-HDMR algorithm appears promising for estimating diverse properties in large libraries, especially when combined with automated synthesis and property measurement procedures.

This paper focused on introducing the SR-HDMR algorithm and its operational components. In practical applications the extracted HDMR component functions $f_i(x_i)$, $f_{ij}(x_i, x_j)$, etc. can be further analyzed to gain physical/chemical insights on the roles played by the substituents acting independently or cooperatively to influence the property values. In this regard the behavior of these functions, upon optimal ordering of their variables, could provide valuable information. Even the norms $||f_i(x_i)||$, $||f_{ij}(x_i, x_j)||$, etc. could give a simple measure of the relative roles of the variables [16].

Further enhancement of the SR-HDMR algorithm may be considered to improve property estimation quality. For example, the component functions may be represented with bases, other than cubic B-splines, that are tailored to the particular application. The current method of simply increasing the size of the training set to enhance estimation quality can also be improved in some circumstances. In particular, rather than performing a larger random sampling over the whole library, certain portions of the library with inadequate estimation quality could be given a sampling bias, or targeted sampling could favor domains which are found to contain compounds with desired

property values. The first case aims at efficiently achieving good quality property estimation for the full library, while the second case seeks a focused library. These options will be especially important when multiple properties of the same library are considered for optimization. The SR-HDMR capability of estimating each individual property over the entire library can be especially valuable with several properties to balance. In such demanding circumstances, intelligent sampling procedures guided by multi-objective optimization algorithms [23] may increase the estimation quality while decreasing the total amount of required sampling. Finally, the SR-HDMR algorithm can potentially be expanded beyond strictly defined libraries of functionalized molecular scaffolds. For example, by treating molecular fragments as the independent variables [10], SR-HDMR has the capability of being utilized to estimate the property values for libraries of diverse constituents.

Acknowledgments The authors acknowledge support form NSF and DARPA QuBE.

Appendices

A Determination of individual component functions

The process of determining the component functions within SR-HDMR has two steps (a) reordering and (b) data fitting with a basis function expansion. Here, we present these processes for first and second order component functions. Application to higher order component functions is performed in a similar manner.

The HDMR expansion can compactly be written as

$$F(\boldsymbol{x}) = f_0 + \sum_{p=1}^{n_p} g_p + \epsilon \tag{4}$$

where g_p is a first or higher order component function (e.g., $f_i(x_i)$, $f_{ij}(x_i, x_j)$, etc.) which depends on its associated variables from $\boldsymbol{x}$. The number of such component functions is n_p, and ϵ is the residual error. The error may arise from a host of factors including operation with an insufficient number of component functions, the approximation of the component functions, data noise, substituent ordering issues, and possibly overfitting. The error will be referred to as the residual property value which has not been adequately explained by the existing HDMR expansion. Adding a new component function g_k into the SR-HDMR expansion is intended to encompass some portion of the residual property value and reduce the norm of ϵ (i.e., ideally while not overfitting). The method of adding a component function for this purpose is explained in the following subsections.

A.1 First order component functions

Each substituent x_i at site i can be assigned a value representing its contribution to the overall property F. These values are only a preliminary estimate of the contribution of x_i to $f_i(x_i)$. The final estimate is determined by reordering the substituents and then representing the function in a basis set.

A.1.1 Reordering for first order component functions

The average residual property value is used as a preliminary estimate of the contribution of the kth substituent, x_i^k, at site i to the overall property value. The average residual property value, $A_i(x_i^k)$, is specified as:

$$A_i(x_i^k) = \frac{1}{T_i(x_i^k)} \sum_{s=1}^{T_i(x_i^k)} \left[F(\boldsymbol{x}^s | x_i^k) - f_0 - \sum_{p=1, g_p \neq f_i}^{n_p} g_p(\boldsymbol{x}^s | x_i^k) \right] \tag{5}$$

where $T_i(x_i^k)$ is the number of samples in the training set with x_i^k at site i, while f_0 and g_p are the component functions already included in the HDMR expansion. If f_i is already *included* in the expansion, then it is *excluded* from $\sum g_p$. This procedure is done in order to recalculate an included function as part of the backfitting process, which will be described in more detail in Appendix B.

A random order gives a landscape $A_i(x_i)$, which likely does not have any recognizable pattern (i.e., analogous to the 2-D case shown in Fig. 1a). A function fitted to such irregularly scattered data points will have poor predictive quality. For first order HDMR functions, the substituents x_i are reordered so that the values of $A_i(x_i)$ are in increasing or decreasing order. This process ensures that the substituents which behave similarly (i.e., with regard to their contribution to the overall property value) will be grouped together, thereby enabling a reliable representation of $A_i(x_i)$

A.1.2 First order component function representation

Smoothing cubic B splines [6,18,22] were used as basis functions to represent (i.e., fit) the reordered average residual property values. A function on the interval of [0, 1] is approximated using $m+3$ cubic B splines ($B_k(x)$ for $k = -1, 0, \ldots, m+1$) where m, the number of "knots", determines how many spline functions are used on the interval. A first order component function is expressed as:

$$f_i(x_i) \approx \sum_{r=-1}^{m+1} \alpha_r^i B_r(x_i) \tag{6}$$

The splines provide a basis to represent $f_i(x_i)$ while filtering irregularities from the property landscape (details given in [6]). The coefficients are determined using least squares regression. The degree of filtering or smoothness is captured by the regularization parameter λ; increasing the value of λ places more weight on optimizing

smoothness and decreasing it places more weight on optimizing the fit. The λ parameter was set to 10 in all trials using smoothing splines [6]. The size of m is determined using significance testing, as described in Appendix C.

A.2 Second order component functions

The pair of substituents (x_i^k, x_j^l) at sites (i, j) can be assigned a value representing their joint contribution to the overall property F. As with the first order component functions, preliminary estimates of the joint property contributions are used to prescribe the optimal order of the substituents, and the function is then represented in a set of basis functions to provide refined estimates of the joint property contribution due to (x_i, x_j).

A.2.1 Reordering for second order component functions

For a second order component function the average residual property value $A_{ij}(x_i^k, x_j^l)$ is expressed as:

$$A_{ij}(x_i^k, x_j^l) = \frac{1}{T_{ij}(x_i^k, x_j^l)} \sum_{s=1}^{T_{ij}(x_i^k, x_j^l)} \left(F(\boldsymbol{x}^s | x_i^k, x_j^l) - f_0 - \sum_{p=1, g_p \neq f_{ij}}^{n_p} g_p(\boldsymbol{x}^s | x_i^k, x_j^l) \right) \tag{7}$$

where $T_{ij}(x_i^k, x_j^l)$ is the number of training samples with x_i^k at site i and x_j^l at site j. These average values form an $M_i \times M_j$ matrix of preliminary property values:

$$\begin{pmatrix} A(x_i^1, x_j^1) & A(x_i^1, x_j^2) & \cdots & A(x_i^1, x_j^{M_j}) \\ A(x_i^2, x_j^1) & A(x_i^2, x_j^2) & \cdots & A(x_i^2, x_j^{M_j}) \\ \vdots & \vdots & \ddots & \vdots \\ A(x_i^{M_i}, x_j^1) & A(x_i^{M_i}, x_j^2) & \cdots & A(x_i^{M_i}, x_j^{M_j}) \end{pmatrix} \tag{8}$$

Here the x_i's in each column take on M_i different values (and similarly for x_j taking on M_j values in the rows). Figure 1a, b corresponds to the likely behavior of such a matrix before and after reordering, respectively. The similarity between two substituents (i.e., defined in terms of their like role upon the property value) at a given site is determined by the similarity between two rows or two columns of the matrix. The similarity can be represented as a "distance" between two rows or two columns. The goal is to reorder and group together substituents that behave similarly. The matrix is generally dense (i.e., few if any entries are unspecified), because the T samples will likely have residual property values projected throughout the 2-D space of the second order component functions. Reordering for dense matrices can be accomplished by a variety of methods and here we use the "bond energy algorithm" [11]. In this fashion the rows and columns of the matrix are separately reordered so as to maximize their associated similarity. The distance (difference) $d_{i'i''}$ between two rows i' and i''

is given by $d_{i'i''} = \left[\frac{\sum_{l=1}^{M_j}(A(x_{i'},x_j^l)-A(x_{i''},x_j^l))^2}{M_j}\right]^{1/2}$. An analogous expression can be written for the distances between the columns of A. If $A(x_{i'}, x_j)$ and/or $A(x_{i''}, x_j)$ are not specified due to insufficient sampling (i.e., the training set does not contain any samples that have a particular pair of substituents at sites i and j), the pair is left out and M_j is adjusted accordingly.

Traveling across the rows or columns of the A matrix has a distance which is incremented upon going from one row or column to the next. The goal is to rearrange the rows and columns to minimize the total distance needed to travel across the matrix (i.e., across all the rows and across all the columns). This task is similar to the traveling salesman problem, which attempts to determine the shortest possible "tour" to pass through a set of "cities" while visiting each exactly once. The rows or columns are the cities and the "tour" through all the rows or columns is their order. However, the situation here differs from the traveling salesman problem because it lacks the condition that the tour must return to the starting city.

Many fast heuristic methods are available to solve for the near-optimal tour which can be modified for our problem. A near-optimal solution may be sufficient for problems where other sources of error, such as the quality of the input data, interfere with ideal reordering. We selected two methods to find the proper ordering (both modified to find the shortest path without returning to the city of origin): dynamic programming [9] and mixed integer programming with subtour elimination [21]. The illustrations in this paper primarily used the latter method and reverted to dynamic programming when the subtour elimination method was too slow. The linear programming code is given in [12].

A.2.2 Second order component function representation

Once the ordering of the rows and columns that optimally decreases the inter-row and inter-column differences is found, the matrix A can be represented (i.e., fitted). The use of B-spline or other basis functions for filtering is necessary even when all the cells in this matrix are filled from the original data. If the filled cells are used as a simple value table of property contributions, the resulting HDMR function generally gives poor estimates for the test set unless a very large training set size T is used. The error is decreased by the smoothing splines to allow for accurate property estimation with a relatively low value of T.

A second order function is represented in the following way:

$$f_{ij}(x_i, x_j) \approx \sum_{p=-1}^{n+1}\sum_{q=-1}^{m+1} \beta_{pq}^{ij} B_p(x_i) B_q(x_j) \tag{9}$$

As in Appendix A.1.2, the coefficients β_{pq}^{ij} are determined by using least squares fitting. To simplify the computations and because we mostly dealt with datasets that had the same number of substituents at each site, we set $m = n$ (an exception is in

Sect. 3.2). The significance testing procedure in Appendix C is used to determine the minimum number of spline functions needed to give a good approximation.

B Inclusion and processing of component functions in HDMR

In this work each of the HDMR component functions is labeled as either *include* or *exclude* based on whether it is employed in the HDMR expansion. Initially all of the component functions except f_0 are excluded. At any point in the process, the HDMR expansion with the included component functions may be expressed as Eq. (4). The residual error comes from a host of factors including an insufficient number or type of component functions, the approximation of the component functions, data noise, substituent ordering issues, and possible overfitting. As in Appendix A, the error ϵ will be referred to as the residual property value of a sample. Each new component function h_k is intended to encompass some of the residual property value and reduce the magnitude of ϵ. Introducing more component functions is done cautiously to avoid overfitting.

If we wish to include a new component function g_k in the HDMR expansion, this new member is determined by fitting it to the residual property value:

$$\epsilon = F(\boldsymbol{x}) - f_0 - \sum_{\substack{p=1 \\ p \neq k}}^{n_p} g_p \tag{10}$$

This equation may be used to determine if each of the excluded first order functions is significant (see Appendix C). Then, the most significant one is included and all of the now-included g_p's undergo an adjustment that is described later in this section. If there are no significant first order component functions, a similar process is performed on the excluded second order component functions. If a significant second order component function is found, then it is included; after the latter step the included first order functions are again adjusted, and the search restarts with the first order excluded functions. If no significant first order functions are found, the included second order functions are also adjusted before the search for second order functions restarts. An overview of this process is given in Fig. 2.

This procedure ensures that the first order contributions are accounted for before the second order contributions are considered. If this is not done, the higher order component functions will not be determined correctly. For example, if the component functions f_i and f_j are not determined first (i.e., see Eq. (2c)) then the component function f_{ij} will improperly contain both the joint and individual contributions from x_i and x_j. A second order component function can still be considered for inclusion if one or both of the corresponding first order component functions are not significant.

Each component function may be affected by the presence of other component functions in the HDMR expansion since the set of g_p component functions are also subtracted from $F(\boldsymbol{x})$ when calculating $f_{ij}(x_i, x_j)$ (i.e., see Eq. (2c)). If a component function g_p is not correlated with the $f_i(x_i)$, $f_j(x_j)$, and $f_{ij}(x_i, x_j)$ terms (i.e.,

g_p does not involve x_i or x_j), then it should not effect the computation of g_{ij}. In practice, there may be some residual correlation due to various factors. However, this circumstance does not prevent the procedure from functioning; the HDMR expansion includes only statistically significant component functions (see Appendix C), and the redundant correlated component functions are excluded.

As a consequence of the aforementioned interdependence, all of the included HDMR component functions are adjusted whenever a new component function is included in the expansion. This adjustment is typically slight, aming to reduce error due to imperfect sampling, component function fitting, and the sequential determination process. This step is done using backfitting [8]: removing a component function from the HDMR expansion and then recalculating it based on the residual property values, which generally will be slightly different from what they were when the component function was initially determined.

The component functions in the HDMR expansion are processed under backfitting based on their order and sequence incorporated in the HMDR. A first order component function under consideration is recalculated and remains in the HDMR expansion, provided that it is still significant. The backfitting process is then repeated for the remainder of the included first order component functions. The second order component functions are processed in the same way with the following sequence of operations: (i) backfit the included first order component functions, (ii) search for significant component functions among the excluded first order functions, (iii) backfit the included second order component functions, and (iv)search for significant component functions among the excluded second order functions. This cycle continues in steps C, D and E of Fig. 2 until the estimation quality for the training set converges or reaches the maximum possible r^2 value.

C Significance testing for the overall HDMR and individual component functions

The F-test was used to determine whether an extra component function or fitting parameter significantly improved the estimation quality [4] of the HDMR. The F-test compares the squared differences between the values given by the original and extended approximations. The squared differences are given as:

$$S_O = \sum_{r=1}^{R} \left[F(\boldsymbol{x}^{(r)}) - f_O(\boldsymbol{x}^{(r)}) \right]^2 \tag{11}$$

$$S_E = \sum_{r=1}^{R} \left[F(\boldsymbol{x}^{(r)}) - f_E(\boldsymbol{x}^{(r)}) \right]^2 \tag{12}$$

where $f_O(\boldsymbol{x})$ is the original approximation for $F(\boldsymbol{x})$ that is currently in use and $f_E(\boldsymbol{x})$ is the extended approximation that is being considered to replace $f_O(\boldsymbol{x})$. These equations are also characterized by the number of parameters employed to approximate them: p_O is the number of parameters used to specify f_O, and p_E is the number of

parameters used to specify f_E. The F-value is given by:

$$F_{\text{val}} = \frac{[S_O - S_E]/(p_E - p_O)}{S_E/(N - p_E)} \tag{13}$$

The degrees of freedom $(p_E - p_O)$ and $(N - p_E)$ are used to calculate an F-distribution [2], and a threshold F-value for the 99 % confidence level is determined. If a calculated F-value is greater than this threshold F-value, the new approximation is adopted, as it is considered significantly better than the original one. In this work the F-test was used to determine (i) whether the HDMR approximation needs another component function or (ii) whether more knots are needed for the spline function approximation of a component function which is considered for inclusion.

When dealing with individual component functions we start with one knot. An F-test can be used to determine whether an extra knot significantly increases the estimation quality. If it does, then a new F-test is performed to see whether this approximation can be significantly improved by adding another knot, etc., until the maximum number of allowed knots is reached (i.e., at most $m + 3$). If the addition of an extra knot is not significant, the component function remains unchanged.

When dealing with significance testing for the entire HDMR expansion, the F-test was used to pick out the significant component functions. In this case the S_O term utilizes all of the included component functions while the S_E term includes one additional component function. In this circumstance p_O is the sum of all the knots in all of the included component functions and the p_E is the sum of p_O and the number of knots in the extra function. A first order component function with k knots means that $(p_E - p_O) = m + 3$ and a second order component function with m knots means that $(p_E - p_O) = (m + 3)^2$.

References

1. KnowItAll Informatics System 8.0, KnowItAll U Edition. Published by the Informatics Division of Bio-Rad Laboratories, Inc
2. JSci—A Science API for Java (2009). http://jsci.sourceforge.net/
3. J. Bicerano, *Prediction of Polymer Properties* (Marcel Dekker, New York, NY, 2002)
4. S. Chatterjee, A.S. Hadi, *Sensitivity Analysis in Linear Regression* (Wiley, New York, NY, 1988)
5. M. Clark, Generalized fragment-substructure based property prediction method. J. Chem. Inf. Model. **45**(1), 30–38 (2005)
6. P. Eilers, B.D. Marx, Flexible smoothing with b-splines and penalties. Stat. Sci. **11**(2), 89–121 (1996)
7. J. Gasteiger, T. Engel (eds.), *Chemoinformatics* (Wiley-VCH, Weinheim, 2003)
8. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning* (Springer, New York, NY, 2009)
9. M. Held, R.M. Karp, A dynamic programming approach to sequencing problems. in *Proceedings of the 1961 16th ACM National Meeting* (ACM, New York, NY, USA, 1961), pp. 71.201–71.204. doi:10.1145/800029.808532
10. W.L. Jorgensen, The many roles of computation in drug discovery. Science **303**(5665), 1813–1818 (2004)
11. P.A. DiMaggio Jr., S.R. McAllister, C.A. Floudas, X.J. Feng, J.D. Rabinowitz, H.A. Rabitz, Optimal methods for re-ordering data matrices in systems biology and drug discovery applications. in *BIOMAT 2007: International Symposium on Mathematical and Computational Biology*, (2008)

12. H.T. Lau, *A Java Library of Graph Algorithms and Optimization. Discrete Mathematics and its Applications* (Chapman & Hall, CRC, London, 2007)
13. A.R. Leach, V.J. Gillet, *An Introduction to Chemoinformatics* (Springer, The Netherlands, 2007)
14. N. Lehming, Regeln fur protein/dna-erkennung. PhD thesis, Universitat zu Koln (1990)
15. N. Lehming, J. Sartorius, B. Kisters-Woike, B. von Wilcken-Bergmann, B. Muller-Hiller, Mutant lac repressors with new specificities hint at rules for protein—dna recognition. EMBO J. **9**(3), 615–621 (1990)
16. G. Li, H. Rabitz, P.E. Yelvington, O.O. Oluwole, F. Bacon, C.E. Kolb, J. Schoendorf, Global sensitivity analysis for systems with independent and/or correlated inputs. J. Phys. Chem. A **114**(19), 6022–6032 (2010). doi:10.1021/jp9096919. http://pubs.acs.org/doi/abs/10.1021/jp9096919
17. G. Li, C. Rosenthal, H. Rabitz, High dimensional model representation. J. Phys. Chem. A **105**(33), 7765–7777 (2001)
18. G. Li, S.W. Wang, H. Rabitz, Practical approaches to construct RS-HDMR component functions. J. Phys. Chem. A **106**, 8721–8733 (2002)
19. F. Liang, X.J. Feng, M. Lowry, H. Rabitz, Maximal use of minimal libraries through the adaptive substituent reordering algorithm. J. Phys. Chem. B **109**, 5842–5854 (2003)
20. S.R. McAllister, X.J. Feng Jr., P.A. DiMaggio, C.A. Floudas, J.D. Rabinowitz, H. Rabitz, Descriptor-free molecular discovery in large libraries by adaptive substituent reordering. Bioorg. Med. Chem. Lett. **18**(22), 5967–5970 (2008)
21. M.W. Padberg, M. Grotschel, in *Polyhedral computations*, ed. by E.L. Lawler, J.K. Lenstra, A.H.G.R. Kan, D.B. Shmoys The Traveling Salesman Problem (Wiley, Chichester, 1985)
22. P.M. Prenter, *Splines and Variational Methods* (Wiley, New York, NY, 1975)
23. J.L. Ringuest, *Multiobjective Optimization: Behavioral and Computational Considerations* (Kluwer, Boston, MA, 1992)
24. N. Shenvi, J.M. Geremia, H. Rabitz, Substituent ordering and interpolation in moleular library optimization. J. Phys. Chem. A **107**(12), 2066–2074 (2003)
25. J.A. Shorter, P.C. Ip, H. Rabitz, An efficient chemical kinetics solver using high dimensional model representation. J. Phys. Chem. A **103**, 7192–7198 (1999)
26. S. Wang, P.R. Jaffe, G. Li, S.W. Wang, H.A. Rabitz, Simulating bioremediation of uranium-contaminated aquifers; uncertainty assessment of model parameters. J. Contam. Hydrol. **64**(3–4), 283–307 (2003)