Inverse Design of Large Molecules using Linear Diophantine Equations

Shawn Martin, W. Michael Brown, Jean-Loup Faulon
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0310
{smartin,wmbrown,jfaulon}@sandia.gov

Derick Weis, Donald Visco
Tennessee Technological University
Department of Chemical Engineering
Cookeville, TN, 38505
{DVisco,dcweis21}@tntech.edu

John Kenneke
Environmental Protection Agency
Ecosystems Research Division
960 College Station Road
Athens, GA, 30605
kenneke.john@epa.gov

Abstract

We have previously developed a method [1] for the inverse design of small ligands. This method can be used to design novel compounds with optimized properties (such as drugs) and has been applied successfully to the design of small peptide antagonists to leukocyte functional antigen-1 (LFA-1) and its intercellular adhesion molecule (ICAM-1). A key step in our method involves computing the Hilbert basis of a system of linear Diophantine equations. In our previous application, the ligands considered were small peptide rings, so that the resulting system of Diophantine equations was relatively small and easy to solve. When considering larger molecules, however, the Diophantine system is larger and more difficult to solve. In this work we present a method for reducing the system of Diophantine equations before they are solved, allowing the inverse design of larger compounds. We present this reduction on our original LFA-1/ICAM-1 dataset, where we were able to reduce a system with 24 equations and 49 variables to an equivalent system with 11 equations and 34 variables, giving a 10 times speedup in performance. We also present the results of our reduction on two new datasets, neither of which we could solve previously: a set of 27 conazole fungicides and a set of 61 γ-secretase inhibitors.

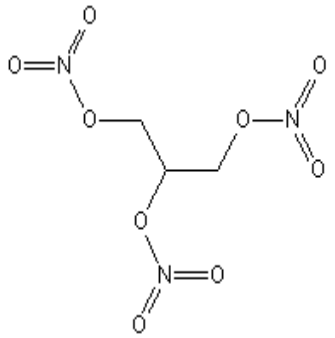
1. Introduction

In previous work [1], we proposed a method for the inverse design of small molecules. This method is based on a fragmental descriptor called *signature*. Signature encodes molecular structure by counting the occurences fragments in a molecule. As an example, we show the molecular signature encoding of nitroglycerin in Figure 1. Further details on signature can be found in [3], where the signature encoding is used in the calculation of quantitative structure-activity relationships (QSARS) and is related to other topological descriptors.

Signature can also be used to reverse engineer molecular structures. This is done by deriving constraint relations that must be present between fragments in order that the fragments may be combined to form a molecule [1]. These constraints consist of one *graphicality* equation and multiple *consistency* equations. The graphicality equation assures that the molecular fragments can be combined to form a connected molecular graph and assumes the form

$$\sum_{i\geq 2}(i-2)n_i - n_1 + 2 = 2z,$$

where n_i is the number of vertices of degree i (number of atoms connected to i other atoms), and z is a non-negative integer. The consistency equations assure that the molecular fragments can be re-connected such that the molecular bonds are consistent. In Figure 1 we show a consistency equation which guarantees that the number of bonds of type $O \rightarrow C$ must be equal to the number of bonds of type $C \rightarrow O$. Any molecular signature will satisfy the constraint equations, and conversely, any signature that satisfies the constraint equations will correspond to some molecule. By solving the constraint equations, we can obtain novel molecular structures which can then be screened for certain properties (e.g. drug activity using a QSAR).



Nitroglycerin

3 o_(n_c_)
6 o=(n_)
3 n_(o_o=o=)
5 h_(c_)
2 c_(o_h_h_c_)
1 c_(o_h_c_c_)

3 o_(n_c_) = 2 c_(o_h_h_c_) + 1 c_(o_h_c_c_)

Figure 1. Signature encoding of Nitroglycerin. Shown here is the molecular graph of Nitroglycerin and a corresponding signature encoding (to the right). Also shown (bottom center) is the consistency equation for the $C \leftrightarrow O$ bond.

However, the constraint equations make up a linear system of Diophantine equations, and solutions must consists of non-negative integer coefficients. This type of system is very difficult to solve (NP-hard) and the best known solver [2] seems to be limited to about 50 or 60 variables for our problem. In this paper, we present a series of simple transformations which reduce the constraint equations. We show that our reductions improve solution time and allow us to provide at least partial solutions to previously unsolved problems.

2. Methods

We reduce the constraint equations using three simple linear transformations. To describe these transformations, suppose we have m equations and n variables. We write our Diophantine system as $A^0\mathbf{x}^0 = \mathbf{b}$, where $A^0_{m\times n} = (a^0_{ij}), \mathbf{x}^0_{n\times 1} = (x^0_j), \mathbf{b}_{m\times 1} = (b_i)$, with a^0_{ij}, b_i integer and x^0_j non-negative integer. We use the superscript notation to denote steps in our reduction, never exponentiation.

In our first reduction, we eliminate equations of the form

$$x^0_j = \sum_{k\neq j} a^0_{ik} x^0_k, \tag{1}$$

where $a^0_{ik} \geq 0$ for $k \neq j$. To eliminate an equation of this form, we replace any occurence of x^0_j in $A^0\mathbf{x}^0 = \mathbf{b}$ with the corresponding sum $\sum_{k\neq j} a^0_{ik}x^0_k$. We can then eliminate both the variable x^0_j and the equation $x^0_j = \sum_{k\neq j} a^0_{ik}x^0_k$ to obtain a reduced system $A^1\mathbf{x}^1 = \mathbf{b}$. Note that the condition $a^0_{ik} \geq 0$ is necessary to ensure that $x^0_j \geq 0$. Further, we obtain a linear transformation

$$\mathbf{x}^0 = T_1\mathbf{x}^1, \tag{2}$$

where T_1 has n rows and $n-1$ columns, and the jth row is given by $\sum_{k\neq j} a^0_{ik}x^0_k$. By repeating this process, we obtain a sequence $T_1, T_2, \ldots, T_p$ of transformations such that we can obtain our original variables from our reduced variables by

$$\mathbf{x}^0 = T_1T_2\cdots T_p\mathbf{x}^p, \tag{3}$$

where $A^p\mathbf{x}^p = \mathbf{b}$ represents our equations after p reductions.

Our next transformation is achieved by considering equations of the form

$$2x^p_j = \sum_{k\neq j} a^p_{ik}x^p_k, \tag{4}$$

where $a^p_{ik} \geq 0$ for $k \neq j$. In this case, we observe that $a^p_{ik} > 1$ can be replaced by the remainder of a^p_{ik} divided by 2, provided that x^p_j is adjusted appropriately. Consider, for example, the equation $2x^p_1 = 3x^p_2 + x^p_3 = 2x^p_2 + x^p_2 + x^p_3$. Here $2(x^p_1 - x^p_2) = x^p_2 + x^p_3$ so that we can replace x^p_1 by a new variables $x^{p+1}_1 = x^p_1 - x^p_2$. Since $2x^p_1 = 3x^p_2 + x^p_3$ we know that $x^p_1 \geq \frac{3}{2}x^p_2 \geq x^p_2$ so that $x^{p+1}_1 = x^p_1 - x^p_2 \geq 0$. In addition, the original variable x^p can be recovered from x^{p+1} by using the relation $x^p_1 = x^{p+1}_1 + x^p_2$. Thus equations of the form (4) again yield a sequence of transformations $M_{p+1}, M_{p+2}, \ldots, M_q$ so that

$$\mathbf{x}^0 = T_1T_2\cdots T_pM_{p+1}M_{p+2}\cdots M_q\mathbf{x}^q, \tag{5}$$

where our further reduced system is now given by $A^q\mathbf{x}^q = \mathbf{b}$.

Finally, it often occurs that A^q has a few identically zero columns after the previous reductions, and even some repeated columns. Identically zero columns represent free variables, which can be removed, and repeated columns represent groups of variables that occur together in every equations. These variable groups can be replaced by single variables and recovered later by solving equations with the form

$$\sum_{i_c} x^q_{i_c} = x^{q+1}_j, \tag{6}$$

where the sum is over the only the indices i_c corresponding to a specific set of repeated columns. These substitutions do not yield transformations of the type in (5), but they are nevertheless easy to solve at a later stage. Upon removal of identically zero and repeated columns, we obtain our fully reduced system $A^r\mathbf{x}^r = \mathbf{b}$, where $r \geq q$.

To solve the reduced system $A^r\mathbf{x}^r = \mathbf{b}$ we use the Diophantine solver in [2]. This solver produces a *Hilbert basis* H^r for the system $A^r\mathbf{x}^r = \mathbf{b}$. This basis consists of a minimal set of solutions to $A^r\mathbf{x}^r = \mathbf{b}$ such that any other solution can be obtained via non-negative integer linear combinations of the solutions in H^r. To obtain the basis H for the original system $A^0\mathbf{x}^0 = \mathbf{b}$, we add unit vectors for any free variable previously eliminated as well as new minimal solutions for any repeated columns that were removed. The minimal solutions for the repeated columns are obtained by solving the equations of the type found in (6) and replacing the variables x_j^{q+1} with the various possibilities for $\{x_{i_c}^q\}_{i_c}$. Finally, the full basis H for the original system $A^0\mathbf{x}^0 = \mathbf{b}$ is obtained using the transformation in (5).

3. Results

We first applied our algebraic reduction to the constraint equations previously derived for peptide rings in an LFA-1/ICAM-1 study [1]. For this problem, we obtained 24 equations with 49 variables. The Diophantine solver identified 2222 minimal solutions in the Hilbert basis. The same basis was found using our reduced system but the Diophantine solver was 10x faster, as shown in Table 1. To the the effects of the different reductions, we also timed the solver at different stages of the reduction process, shown in Figure 2. The most noticable improvement was seen using the reduction equations in (1), giving a 5x speedup; the reductions equations in (4) had no effect on this problem; and the reduction equations in (6) gave an additional 2x speedup.

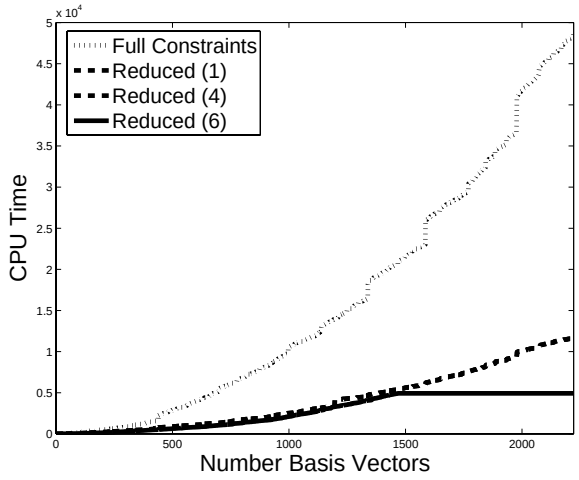


Figure 2. Effect of Algebraic Reductions. In this plot we see the effects of the algebraic reductions in (1), (4), and (6) on the performance of the Diophantine solver.

	Original	Reduced	% Reduction
Peptides			
vars	49	34	30.6
eqs	24	11	54.2
cpu time	–	–	90.0
γ-secretase			
vars	68	49	27.9
eqs	21	9	57.1
cpu time	–	–	98.0
Conazoles			
vars	91	64	29.7
eqs	29	15	48.2
cpu time	–	–	94.1

Table 1. Performance Statistics. Here we compare the use of the reduced constraint equations with the original equations in terms of number of variables, equations, and percent reduction in cpu time.

We next applied our reduction to the inverse design of γ-secretase inhibitors for Alzheimer's disease. This dataset was obtained from [4] and consisted of 61 compounds with varying IC_{50} values. For this dataset we derived 21 constraint equations with 68 variables. The overall speedup for this problem was approximately 51x. In fact, we could only obtain one minimal solution using the full system of constraints. In the same time that this one solution was found, we found 51 solutions using the reduced equations. Further, we found 900 solutions after running the solver for 24 hours (using the reduced system). These 900 minimal solutions yield an arbitrary number of new γ-secretase inhibitors via linear combinations. We computed over 700,000 new compounds to start. A summary of the performance of our method in this problem is also shown in Table 1.

Finally, we applied our reduction to the design of non-toxic but still effective conazole fungicides. These 27 fungicides were obtained from the Enivronmental Protection Agency's Persistent, Bioaccumulative, and Toxic (PBT) Profiler database (www.pbtprofiler.net), each with a corresponding fish chronic toxicity value (ChV). For this dataset, we obtained 29 constraint equations with 91 variables. Using the full set of constraints, the Diophantine solver was able to obtain only 4 minimal solutions. Using the reduced equations, we obtained 68 solutions in the same time, giving a 17x speedup. We used the solutions to compute more than 500,000 new conazole fungicides. Performance statistics are again shown in Table 1.

4. Discussion

We have proposed a simple method for reducing a linear system of homogeneous equations when using the signature molecular descriptor for inverse design of chemicals. We have tested the reduction on three datasets, including a set of ICAM-1 inhibitory peptides, a set of γ-secretase inhibitors, and a set of conazole fungicides. On these three datasets we achieved an average reduction of 29.4% in the number of variables and 53.2% in the number of equations, resulting in an average reduction in computation time of 94.0%. This increase in efficiency allows us to use the signature descriptor to design large molecules, previously impossible with our technique.

5. Acknowledgements

This work was funded by interagency agreement (IAG) DW89921601 between the Environmental Protection Agency (EPA) and Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] C. J. Churchwell, M. D. Rintoul, S. Martin, D. P. Visco, A. Kotu, R. S. Larson, L. O. Sillerud, D. C. Brown, and J.-L. Faulon. The signature molecular descriptor. 3. Inverse quantitative structure-activity relationship of ICAM-1 inhibitory peptides. *Journal of Molecular Graphics and Modelling*, 22(4):263–273, 2004.

[2] E. Contejean and H. Devie. An efficient incremental algorithm for solving systems of linear diophantine equations. *Information and Computation*, 113(1):143–172, 1994.

[3] J.-L. Faulon, D. P. Visco, and R. S. Pophale. The signature molecular descriptor. 1. Using extended valence sequences in QSAR and QSPR studies. *Journal of Chemical Information and Computer Science*, 43(3):707–720, 2003.

[4] C. V. C. Prasad, J. W. Noonan, C. P. Sloan, W. Lau, S. Vig, M. F. Parker, D. W. Smith, S. B. Hansel, C. T. Polson, D. M. Barten, K. M. Felsenstein, and S. B. Roberts. Hydroxytriamides as potent g-secretase inhibitors. *Bioorganc and Medicinal Chemistry Letters*, 14:1917–1921, 2004.