META. 3. A Genetic Algorithm for Metabolic Transform Priorities Optimization

Gilles Klopman,* Meihua Tu, and Joseph Talafous†

Chemistry Department, Case Western Reserve University, Cleveland, Ohio 44106

Received July 5, 1996⊗

META is a knowledge-based expert system that simulates the biotransformation of xenobiotics. It operates with the help of a dictionary (knowledge base) to seek target fragments in a compound and transform them to products. Here, a genetic algorithm is introduced to help build the knowledge base and optimize the performance of the methodology.

INTRODUCTION

The ever-increasing use of chemicals for industrial solvents, cosmetics, food additives and preservatives, herbicides and pesticides, drugs, and medicines has brought about a tremendous increase in the possible routes to exposure of humans and animals to xenobiotics. The effects of xenobiotics on the organism may be caused not only by the parent agent but also by its metabolites which are the result of biotransformations. As new chemical agents are designed for the variety of purposes alluded above, there is an increasing need to predict accurately and completely the nature and bioactivity of their metabolites. Hence, our laboratory has been developing a computer program (called META; the META program is distributed through MULTICASE Inc., P.O. Box 22517, Cleveland, OH 44122) that predicts the possible metabolic transformation products of xenobiotics along with the possible bioactivity of these metabolites.[1,2] This program employs principles somewhat related to computer-assisted organic synthesis methodologies described by other authors.[26−28]

Used with an appropriate dictionary that acts as the knowledge base, META recognizes chemical functional groups and applies chemical transformation rules to generate potential metabolites. Each transform is assigned a priority value according to the prevalence of the observed metabolites by experts. Even though this prioritization may by approximate, it is very important to arrange metabolites hierarchically in order to control the combinatorial explosion of potential metabolic pathways. Otherwise, investigators may become overwhelmed with the large numbers of irrelevant metabolic pathways that could be generated. As the number of transforms increases, it becomes impossible to assign priorities accurately by simple examination of the products. Moreover, as new transforms are added into the dictionary and because of the interdependency of the transforms, all the priorities need to be periodically reoptimized. Otherwise, the fragile balance among the priorities will be destroyed, and far-reaching deleterious consequences may result. So, developing a methodology to optimize the predictability of the program is necessary for building the dictionary. Unfortunately, the procedure is very complex as the learning set involves a large amount of sometimes conflicting information. Furthermore, the problem is not linear and cannot easily be represented as a continuous mathematical function. We therefore investigated a number of possible avenues[29] and report here the results of our implementation of a genetic algorithm to solve the problem.

The principles of genetic algorithm (GA) were first introduced by Holland.[3] The main idea of genetic algorithm is based upon the basic concepts of Darwinian principles of natural selection.[4] As an optimization method, GA has recently been widely applied to conformation searches.[5−9] In a typical implementation, a population of random bit strings is used as the starting population of solution trials. Each trial is encoded according to the particular application's representation. The quality of each solution generated by the algorithm is judged by a fitness criteria, which is used to select certain parents for succeeding generations. The better the solution, as compared with their competitors in the population, the higher the probability that their corresponding bit string is selected for survival and recombination. Further, a low mutational frequency is introduced when offspring strings are copied from parent strings. Succeeding generations should thus encode increasingly better solutions ("fitter population"), achieved through selection, mutation, and recombination. The process is repeated until a few populations dominate. The surviving individuals should represent a near-optimal solution.

In this paper, we use such a genetic algorithm to optimize the performance of the metabolism biotransform priorities.

METHODS

META Program. Basically, META operates from a dictionary of transforms, consisting of pairs of structural fragments known to describe a metabolic transformation. The program seeks the presence of a target structural fragment and transforms it into a product fragment. This searching process continues until no more targets are found. The program monitors and evaluates the stability of all the generated molecules by consulting a dictionary of spontaneous reaction that lists unstable structural fragments. Whenever a molecule is produced and found to contain such an unstable fragment, it is transformed into a stable product via an appropriate spontaneous reaction transform contained in the spontaneous reaction dictionary. A detailed description of the META algorithm was published previously.[1,2]

Dictionary. Since the utility of META is largely determined by the content of the dictionary, the philosophy of its design was carefully considered. Our current dictionary incorporates only well-established metabolic data from reviews, textbooks,[10−16] and monographs.[17−19] While there

† Current address: MDL Information Systems Inc., San Leandro, CA 945177.

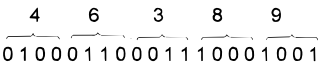
⊗ Abstract published in *Advance ACS Abstracts,* February 15, 1997.

S0095-2338(96)00112-6 CCC: $14.00

are hundreds of transforms in the full dictionary, only 70 of them (Table 1) are included in the test dictionary described below, and 50 compounds (Table 2) served as a training set for refining the dictionary. The performance of individual transforms was determined by a comparison between the actual experimental metabolic status (observed vs not observed) and the META predictive status for each transform (hit vs miss).

Genetic Algorithm. In this section, we describe the details of the GA method as we applied it to the transform priority optimization problem. The goal of the program is to fit the META-calculated results to the experimentally observed products as closely as possible. Generally, there are four steps in a typical genetic algorithm: initialization, selection, crossover, and mutation.

(a) Initialization. At initialization, a GA population of size *N* is generated randomly. Each individual in this population contains *m* values, where *m* is the total number of transforms, and all the values are encoded into fixed-length binary form. For example, if there are a total of five transforms in the dictionary and the priority values for the transforms are 4, 6, 3, 8, and 9 , respectively, then the corresponding binary form for the individual will look like



Here, the length for each binary number is 4. In our case, the set of priorities assigned by the experts were encoded into binary form and served as one individual in the initial population; all the subsequent sets of priorities (individuals) were generated randomly.

To evaluate the fitness value of each individual (set of priorities), we submitted 50 test compounds to the META program operating with the initial expert-ranked priorities. An output file showed all possible transforms related to each test compound. The output was then compared to the experimental data, and the results were classified into categories. Four categories are possible:

Hit(unmasked): The metabolite observed in the experiment is predicted by the META program as a major product and is not masked, i.e., it either has the most favorable priority or is very close to that of the most favorable transform. In this case, the fitness value will increase.

Hit(masked): The metabolite observed experimentally is also predicted by the META program. However, due to the low priority of the transform, this metabolite will be masked by META, which means it will not be displayed as a major metabolite. In this case, the fitness value will decrease. The criteria used to decide whether to mask a metabolite in META is that the priority value of the corresponding transform is more than 2 units less than the highest priority value of all the transforms associated with the substrate.

Miss: The metabolite is observed experimentally but not predicted by META. In this case, the fitness value will not change, but a warning message will remind the knowledge-base developer to add an appropriate transform in the dictionary later so that the metabolite will be identified in the future.

Over: The metabolite is predicted by META but not observed experimentally. In this case, two steps will take place. First, the GA will reduce the priority of the corresponding transform so as to mask the metabolite. Second, a warning message will suggest to the knowledge-base developer the possible need to modify the transform (make it more specific) or delete it all together.

The raw fitness value of a given set of priorities is then calculated as in eq 1. The goal of this genetic algorithm optimization is to increase the number of hit(unmasked) while reducing the number of hit(mask) and over transforms.

$$\text{rawfit} = \sum \text{hit(unmasked)} - \sum \text{hit(mask)} - \sum \text{over} \quad (1)$$

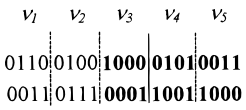
(b) Selection. In our GA program, the Boltzmann tournament selection[20] method was used to select the population's fittest individuals. Experimentation shows that the Boltzmann tournament selection converges faster than the roulette-wheel selection[21] or the tournament selection[21] in our specific application. In the Boltzmann tournament selection, the system anneals to a low final temperature from a high initial temperature, and the corresponding fitness value is the Boltzmann distribution form of the raw fitness (eq 2):

$$F(i) = \frac{e^{\text{rawfit}(i)/T}}{\sum_j e^{\text{rawfit}(j)/T}} \quad (2)$$

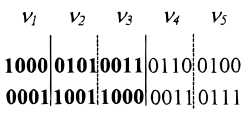
where *T* is the temperature of the current generation.

The actual tournament is based on the Boltzmann fitness value as calculated by eq 2. In one generation, a size *S* subpopulation (usually S is greater than 2 and less than 10) is picked randomly from the whole population. The fittest individual (the binary string with the largest fitness value) from the subpopulation is selected and reproduced. This step continues until a total of *N* individuals are selected. They form a new population.

(c) Crossover. The crossover step consists of a procedure that mates the binary string and exchange information. In our program we implemented a one-point crossover method (this point was generated randomly). For example, supposing we have a total of five transforms in the dictionary, a crossover point of 3 is generated in the program, and the length of each binary number is 4, then the following two binary strings



will produce two children such as:



(d) Mutation. Basically, mutation will mutate one bit in the binary string (from 0 to 1 or from 1 to 0) under a proper probability.

After each cycle of GA operation (selection, crossover, and mutation), the fitness of the new population is evaluated again by assigning a fitness value to each set of priorities, and another series of GA operations will take place until convergence is achieved or the stop condition is satisfied. The criteria to determine convergence is that the root mean square deviation of the fitness values of two adjacent

Table 1. 70 Transforms[a] in the Dictionary and Molecules of the Training Set That Contain Them

reaction	transform	compound
hydroxy replacement of activated α-carbonyl halide	CO −CQ −X CO −CQ −OH	**14**
hydrolysis of imines	C@ ═N −C# − C@ NH_2−C# −⟨1═O⟩	**18**
N-acylurea hydrolysis to *N*-urea and acid	NQ −CO −N! −CO − NQ −CO −N+ CO −⟨4−OH⟩	**44**
dehalogenation	C″ −CQ −C@ −Cl ⟨2-C″⟩ C″ −C− ═C@ Cl_⟨2-C″⟩	**17**
aliphatic hydroxylation at penultimate carbon	CH_3−CQ − CH_3−C− −⟨2-OH⟩	**3**, **19**, **28**, **24**, **38**, **40**, **41**
aliphatic hydroxylation at terminal carbon	CH_3−C@ − CH_2−C@ −⟨1-OH⟩	**3**, **24**, **28**
aliphatic hydroxylation of secondary carbon	CH_2−CQ −C@ − CH_2−C− −C@ −⟨2-OH⟩	**12**, **14**
aliphatic hydroxylation of halogenated alkanes	CW −X C− −X ⟨1-OH⟩	**26**, **31**
aliphatic hydroxylation of aryl/vinyl methyl group	CH_3−C! ═ CH_2−C! ═⟨1-OH⟩	**5**, **15**, **19**
hydroxylation of aromatic/alkene bond	CQ ═CQ − CQ ═C− −⟨2-OH⟩	**4**, **9**, **14**, **16**, **18**, **27**, **30**, **42**, **3**, **40**, **50**
hydroxylation of non-tert aliphatic carbon adjacent to oxygen	O! −CW − O! −C− −⟨2-OH⟩	**14, 20, 38, 41**
hydroxylation of non-tert alkyl in aryl/vinyl alkyl ether	C!″−O −CW − C!″−O −C− −⟨3-OH⟩	**5, 41**
hydroxylation of non-tert aliphatic carbon adjacent to sulfur	S! −CW − S! −C− −⟨2-OH⟩	**12**, **15**
C-hydroxylation of alkyl primary amine, α to N	NH_2−CW − NH_2−C− −⟨2-OH⟩	**3**
C-hydroxylation of alkyl sec, tert amine, α to N	C# −N −CW − C# −N −C− −⟨3-OH⟩	**3**, **18**, **22**, **24**, **30**, **35**, **39**
C-hydroxylation of aryl/vinyl alkyl amine	C″ −N! −CW − C″ −N! −C− −⟨3−OH⟩	**11**
N-hydroxylation of (form)amide	NH −CO − N −CO −⟨1-OH⟩	**2**, **14**, **41**
N-hydroxylation of aryl/vinyl amine	NQ −C ═ N− −C ═⟨1-OH⟩	**2**, **4**, **27**, **41**, **45**
hydroxylation of sp^3 adjacent to nitrogen in *N*-nitrosoamines	NO −N! −CW NO −N! −C− ⟨3-OH⟩	**22**
C-hydroxylation next to amides	CO −N! −CW − CO −N! −C− −⟨3-OH⟩	**18**
desulfurization of organophosphosulfur compound	O −PS −O O −PO −O	**38**
epoxidation of double bond	CQ ═CQ − CQ −CQ −⟨1-O-2⟩	**7**, **8**, **34**, **42**, **46**, **47**, **48**
N-oxidation of aryl/vinyl imine	C! ═N − C! ═NO −	**18**, **19**, **35**, **46**
oxidative dehalogenation	CO −CQ −X CO −C− ═O	**14**
P-oxidation	C″ −P −C″ C″ −PO −C″	**23**
N-hydroxylation of prim, sec aliphatic amines	NQ −C# − N− −C# −⟨1-OH⟩	**3**, **14**, **24**
N-oxidation of tertiary nitrogen	C# −N −C# −⟨2-C −⟩ C# −NO −C# −⟨2-C −⟩	**35**
S-oxidation of aliphatic sulfhydryl to sulfenic acid	C# −SH C# −S ⟨2-OH⟩	**12**
S-oxidation of aliphatic sulfhydryl to sulfenic acid	C# −S − C# −SO −	**15**
S-oxidation of aryl/vinyl sulfhydryl to sulfenic acid	C″ −SH C″ −S ⟨2-OH⟩	**49**
dehydration of aliphatic alcohols	OH −C@ −CW − OH_2 C@ ═C− −	**14**
reduction of aryl/vinyl nitro to nitroso	NO_2−C ═ NO −C ═	**14**, **36**, **38**
reduction of terminal aliphatic trihalides	X −C −X ⟨2-F⟩ X −CH X _⟨2−F⟩	**13**, **17**
reduction of di(aryl/vinyl) azo compounds	C! ═C −N ═N −C ═ C! ═C −NH_2 NH_2−C ═	**45**
reduction of disulfides	CS −S −S −CS CS −SH SH −CS	**24**
reductive dehalogenation	F −C −CH −Br ⟨3-Cl−⟩ F −C −CH_2 Br_⟨3-Cl−⟩	**24**
reduction of sulfoxide	C# −SO −C# C# −S −C#	**24**
hydrolysis of aryl/vinyl ester	CO −O −C! ═ CO −OH C! ═⟨3-OH⟩	**5**
hydrolysis of any amide	N@ −CO − N+ CO −⟨2-OH⟩	**1**, **2**, **30**, **14**, **29**, **44**
glycination of pri, sec, tert aliphatic acids	OH −CO −C# − NH −CO −C# −⟨1-CH_2−CO −OH⟩	**32**, **12**, **43**
glycination of aryl/vinyl carboxylic acids	OH −CO −C ═ NH −CO −C ═⟨1-CH_2−CO −OH⟩	**10**
glutamation of pri, sec, tert aliphatic acids	OH −CO −C# − NH −CO −C# −⟨1-Z_2⟩	**12**, **32**, **43**
glutamation of aryl/vinyl carboxylic acids	OH −CO −C ═ NH −CO −C ═⟨1-Z_2⟩	**10**
glutamation of benzylic/allylic acids	OH−CO−C@−C ═ NH−CO−C@−C ═ ⟨1-Z_2⟩	**43**
GSH conjugation to isocyanate	N! ═CO N+ −CO ⟨2-Z_3⟩	**33**
GSH conjugation of sulfhydryl	C# −SH C# −S ⟨2-Z_3⟩	**12**
glucuronic O-conjugation of carboxylic acid	OH −CO − O −CO −⟨1-Z_4⟩	**10**, **12**, **43**
glucuronic acid O-conjugation of alkyl alcohol	OH −C# − O −C# −⟨1-Z_4⟩	**14**
glucoronic acid O-conjugation of aryl/vinyl alcohols	OH −C! ═ O −C! ═⟨1-Z_4⟩	**1**, **42**, **49**
glucoronic acid N-conjugation of aryl/vinyl primary amine	NH_2−C ═ NH −C ═⟨1-Z_4⟩	**4**, **45**
O-sulfation of aliphatic alcohols	OH −C# − O −C# −⟨1-SO_2−OH⟩	**14**
O-sulfation of aryl/vinyl alcohols	OH −C ═ O −C ═⟨1-SO_2−OH⟩	**1**, **42**, **49**
N-sulfation of aryl/vinyl amines	NH_2−C ═ NH −C ═⟨1-SO_2−OH⟩	**4**, **45**
N-methylation of aliphatic primary amines	NH_2−C# − NH −C# −⟨1-CH_3⟩	**3**
O-methylation of aryl/vinyl alcohols	C″ −OH C″ −O ⟨2-CH_3⟩	**42**
S-methylation of aryl/vinyl sulfhydryl	SH −C″ S −C″ ⟨1-CH_3⟩	**49**
thiol-*S*-methyltransferase	C# −SH C# −S ⟨2-CH_3⟩	**12**
N-methylation of *N*-heterocycles	C! ═N −C! ═ C! ═N+ −C! ═⟨2-CH_3⟩	**19**, **29**, **46**
N-acetylation of primary aliphatic amines	NH_2−C# − NH −C# −⟨1-CO −CH_3⟩	**9**
N-acetylation of aryl/vinyl amines	NQ −C! ═ N− −C! ═⟨1-CO −CH_3⟩	**4**, **27**, **28**, **9**
N-acetylation of hydrazine derivatives	NH_2−N@ − NH −N@ −⟨1-CO −CH_3⟩	**27**, **29**
N-acetylation of sulfonamide	SO_2−NH_2 SO_2−NH −CO −CH_3	**45**
oxidation of primary aliphatic alcohol	OH −CH_2− O ═CH −	**14**, **25**
oxidation of secondary aliphatic alcohol	OH −CH − O ═C −	**14**
oxidation of purines	N ═CH −N! − N ═C −N! −⟨2-OH⟩	**11**
hydrolysis of dihalide	X −C −X X_C ═O	**14**, **17**
dehydrogenation of prim or sec amines	C# −NH −CQ − C# −N ═C− −	**3**
oxidation of tert amines	C″ −CQ −N −C# − C″ −C− ═N+ −C# −	**39**
more specific MAO reaction for neurotransmitters	NH_2−CQ −C ═ COH−CQ −C ═	**3**
β-oxidation of long chain hydrocarbons	CH_2−CH2−CO −OH CO CH_3−CO −OH ⟨1-OH⟩	**32**

[a] Special characters in the transforms are as follows: ⟨ ⟩, substituent attached at indicated position; ″, an sp^2 carbon; X, a halogen; !, the presence of 0 or 1 hydrogen atom; @, 0−2 hydrogens; #, 0−3 hydrogens; Q, 1 or 2 hydrogens; W, 1−3 hydrogens; _, a negative charge; + and −, directly following a carbon atom, add or remove a hydrogen atom, respectively.

Table 2. 50 Training Set Compounds

no.	name	no.	name
1	acetaminophen	**26**	halothane
2	2-(acetylamino)fluorene	**27**	hydralazine
3	amphetamine	**28**	ibuprofen
4	aniline	**29**	isoniazid
5	aspirin	**30**	lidocaine
6	BCNU (carmustine)	**31**	lindane (gamma)
7	benzo[*a*]pyrene	**32**	methotrexate
8	benzene	**33**	methyl isocyanate
9	benzidine	**34**	naphthalene
10	benzoic acid	**35**	nicotine
11	caffeine	**36**	nitrobenzene
12	captopril	**37**	nitroglycerin
13	carbon tetrachloride	**38**	parathion
14	chloramphenicol	**39**	pargyline (eutonyl)
15	cimetadine	**40**	pentobarbital
16	coumarin	**41**	phenacetin
17	DDT (chlorphenothane)	**42**	phenol
18	diazepam	**43**	phenylacetic acid
19	diazinon	**44**	phenytoin
20	dichlorvos	**45**	prontosil (sulfamidochrysoidine)
21	dioxin (TCDD)	**46**	pyridine
22	dimethylnitrosamine	**47**	styrene
23	diphenylmethylphosphine	**48**	thiophene
24	disulfiram	**49**	thiouracil
25	ethanol	**50**	toluene

generations is less than 10^{-4}. The best individual set of priorities for each generation is stored, and the best overall set will be the final result of the program.

RESULTS AND DISCUSSION

Because of the complexity of our problem (we have more than 200 variables or transforms to be optimized in a full dictionary), it is important to choose appropriate parameters and methods to increase the convergence rate and reduce the CPU time while avoiding the local minimas. As stated before, the Boltzmann selection method of the genetic algorithm converges faster than the roulette wheel selection and the tournament selection in our specific application. The annealing rate we chose starts at 1000 K, then maintains this temperature for 50 generations, and finally drops the temperature 4 K/generation until convergence is achieved or the final temperature reaches 200 K. This annealing procedure seems to work best in our situation. Figure 1 lists a comparison of the convergence rate between the three selection methods. It is worth noting that all three methods have advantages and drawbacks, and each may work best in specific applications. Readers may wish to check other books and articles for more detailed comparisons.[20,21]

In Table 3, we show how the number of hit(unmasked), hit(mask), and over transforms for the 50 training set compounds changed after optimization of the priorities has taken place. It is obvious that the number of correctly predicted transforms has increased after optimization while the number of incorrectly masked and over predicted transforms has decreased.

After optimization of the biotransform priorities, we needed to test the new dictionary. The chief objective of our test is to see if the META program can predict the experimentally observed metabolites and mask the unobserved ones. For the metabolites that were missed, we might have to add appropriate transforms and restart the optimization process.

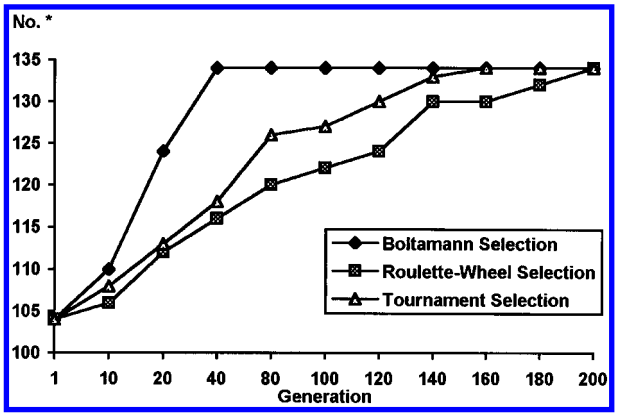


Figure 1. Convergence rate of the different selection methods. All three methods used a mutation rate of 0.6 and a crossover rate of 0.05. No. means the number of transforms correctly predicted. Because the three selection methods use different fitness value scales, it is better to evaluate their convergence rate by comparing their correctly predicted transforms.

Table 3. Comparison of the Predictions for the 50 Molecules in the Training Set as Found by the Use of the Expert-Ranked Dictionary and the GA-Optimized Dictionary

	expert-ranked dictionary	GA-optimized dictionary
hit(unmasked)[a]	103	134
hit(mask)	45	14
over	28	18
miss	0	0

[a] The total number of experimental transforms associated with the 50 compounds is 148.

The first exercise was to evaluate the ability of our program to identify the observed metabolites of compounds unknown to our system. We arbitrarily selected 30 common xenobiotics from D. R. Hawkins' book[22] and B. J. William's book.[23] None of these compounds except amphetamine, isoniazid, and phenylacetic acid were included in our training set of 50 compounds. We first ran the META program with a dictionary using the priorities suggested by our experts and then with the dictionary optimized by the genetic algorithm. Our results show that the META program using the GA-optimized dictionary made better predictions than that using the expert-ranked dictionary. Indeed, the META program with the GA-optimized dictionary predicted all the main biotransforms of the tested xenobiotics while producing a lesser number of irrelevant metabolites; when used with the expert-ranked dictionary, it missed major metabolites for seven of the compounds and generated a greater number of unnecessary metabolites (see Table 4).

Our second controlled experiment consisted of testing the ability of the genetic algorithm to rebuild the dictionary once we add new transforms. First, we selected a compound whose biotransformations would not be predicted correctly by our program with its current dictionary because a needed transform was missing. The compound was 4-(methylnitrosamino)-1-(3-pyridyl)butan-1-one whose metabolism in rats was reported as shown in Figure 2.[24]

Metabolites **2**−**4** were predicted by the META program. **2** and **4** were produced by the transform "hydroxylation of sp^3 carbon adjacent to nitrogen in *N*-nitrosoamines" followed by spontaneous reactions, and **3** was produced by the transform "N-oxidation of aryl/vinyl imine". Metabolite **1** was missed by META, so we needed to add the transform (reduction of ketones: C# −CO −C# → C# −CH −C#

Table 4. Evaluation of 30 Test Compounds.

compound	no. (expt)	META (expert) hum	hm	over	META (GA) hum	hm	over
cyclohexylamine	3	3	0	2	3	0	1
2-isopropylnaphthalene	4	3	1	3	4	0	1
isoprene	2	2	0	1	2	0	0
valproic acid	4	3	1	3	4	0	2
4-nitroanisole	2	2	0	2	2	0	0
3-methyl-4-nitroanisole	4	2	2	2	4	0	1
amphetamine	3	3	0	3	3	0	2
meperidine	4	2	2	3	4	0	3
chlorobenzene	2	2	0	0	2	0	0
cyclohexane	1	1	0	1	1	0	0
phenylacetamide	2	2	0	2	2	0	0
isoniazid	1	0	1	3	1	0	1
phenetidine	3	3	0	2	3	0	0
p-hydroxylbenzyaldehyde	2	2	0	3	2	0	1
theophylline	2	2	0	2	2	0	1
N,*N*-dimethylaniline	2	2	0	1	2	0	0
pentan-2-ol	2	2	0	1	2	0	0
propan-2-ol	2	2	0	0	2	0	0
N-propylamine	1	1	0	3	1	0	1
adrenaline	3	2	1	2	3	0	1
2,5-dimethylfuran	2	2	0	0	2	0	0
p-xylene	2	2	0	2	2	0	1
indene	2	2	0	1	2	0	0
propylcyanide	2	2	0	1	2	0	0
p-tolunitrile	2	2	0	1	2	0	0
p-nitrophenol	4	4	0	2	4	0	1
N-methylaniline	1	1	0	2	1	0	0
p-aminobenzoic acid	3	2	1	3	3	0	2
2,4-dioxoimidazolidine	3	3	0	2	3	0	1
hetrazan	2	2	0	2	2	0	1
phenylacetic acid	3	3	0	1	3	0	0

[a] The number of metabolites found experimentally and by META; no. (expt), the number of metabolites observed experimentally; hum, the number of transforms that are observed experimentally and predicted by META without being masked; hm, the number of transforms that are observed experimentally, predicted by META, and masked by META; over, the number of transforms that are not observed experimentally but are predicted by META.

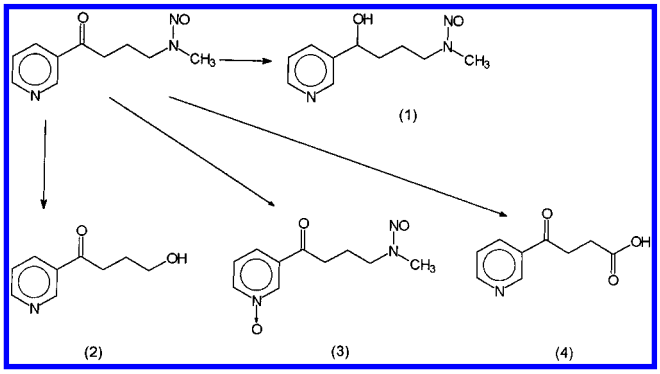


Figure 2. Metabolites of 4-(methylnitrosamino)-1-(3-pyridyl)butan-1-one in rats.

−⟨2-OH⟩) to the dictionary. We added the new compound to our training set and reoptimized the new dictionary with our genetic algorithm. We then submitted both 4-(methylnitrosamino)-1-(3-pyridyl)butan-1-one and (*R*)-(+)-pulegone,[25] which was experimentally also known to involve the reduction of a carbonyl group, to the META program with the upgraded dictionary. It predicted correctly all the transforms observed experimentally for both compounds (Figures 2 and 3).

Metabolite **5** was produced by the transform "reduction of ketones", metabolite **6** by the transform "epoxidation of double bond", and metabolite **7** by the transform "aliphatic hydroxylation of aryl/vinyl methyl group".

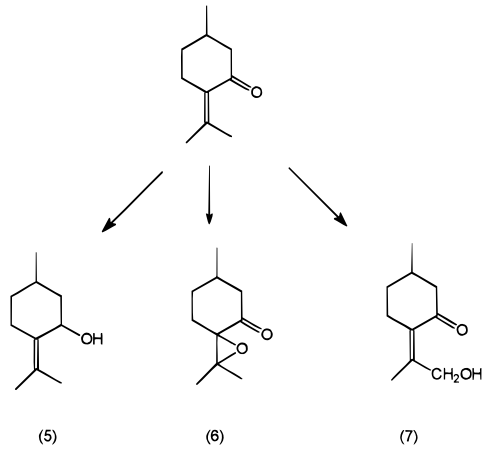


Figure 3. Metabolites of (*R*)-(+)-pulegone in rats.

CONCLUSION

We have successfully developed a genetic algorithm to optimize metabolic transform priorities and demonstrated that the algorithm is useful in dictionary development. The dictionary optimized by our genetic algorithm has proven to rarely miss the prediction of possible metabolites and does not generate spurious predictions. Future research will focus on widening the scope of the program and on further refinement of the dictionary. With the availability of this genetic algorithm program, future dictionaries could be made species-specific, and other factors that are known to affect metabolism, such as sex, age, and exposure history, may also be considered in determining the priority of the transforms.

REFERENCES AND NOTES

(1) Klopman, G.; Dimayuga, M.; Talafous, J. META. 1. A Program for the Evaluation of Metabolic Transformation of Chemicals. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1320−1325.
(2) Talafous, J.; Sayre, L. M.; Mieyal, J. J.; Klopman, G. META. 2. A Dictionary Model of Mammalian Xenobiotic Metabolism. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1326−1333.
(3) Holland, J. H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66−73.
(4) Antonoff, M. Genetic Algorithms: Software By Natural Selection. *Popular Sci.* **1991**, October, 70−74.
(5) Blommers, M. J. J.; Lucasius, C. B.; Kateman, G.; Kaptein, R. Conformational Analysis of a Dinucleotide Photodimer with the Aid of the Genetic Algorithm. *Biopolymers* **1992**, *32*, 45−52.
(6) McGarrah, D. B.; Judson, R. S. Analysis of the Genetic Algorithm Method of Molecular Conformation Determination. *J. Comput. Chem.* **1993**, *14* (11), 1385−1395.
(7) Payne, A. W. R.; Glen, R. C. Molecular Recognition Using a Binary Genetic Search Algorithm. *J. Mol. Graph.* **1993**, *11*, 74−91.
(8) Clark, D. E.; Jones, G.; Willett, P. Pharmacophoric Pattern Matching in Files of Three-Dimensional Chemical Structures: Comparision of Conformational-Searching Algorithms for Flexible Searching. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 197−206.
(9) Unger, R.; Moult, J. Genetic Algorithms for Protein Folding Simulations. *J. Mol. Biol.* **1993**, *231*, 75−81.
(10) Hathaway, D. E. *Foreign Compound Metabolism in Mammals*; The Chemical Society: London, 1975; Vol. 3.
(11) Testa, B.; Jenner, P. *Drug Metabolism: Chemical and Biochemical Aspects*; Marcel Dekker: New York, 1976.
(12) Paulson, G. D.; Caldwell, J.; Hutson, D.; Menn, J. J.; *Xenobiotic Conjugation Chemistry*; American Chemical Society: Washington, DC, 1986; p 299.
(13) Williams, D. A. In *Principles of Medicinal Chemistry*, 3rd ed.; Foye, W. O., Ed.; Lea & Febiger: Philadelphia, PA, 1989; pp 79−117.
(14) Gilman, A. G.; Rall, T. W.; Nies, A. S.; Taylor, P. *The pharmacologi-*

cal Basis of Therapeutics, 8th ed.; MacMillian Publishing Co.: New York, 1990.
(15) Alvares, A. P.; Pratt, W. B. *Principles of Drug Action*, 3rd ed.; Churchill Livingstone: New York, 1990.
(16) Timbrell, J. A. *Principles of Biochemical Toxicology*, 2nd ed.; Taylor & Francis: London, 1991.
(17) Jakoby, W. B. *Enzymatic Basis of Detoxification*; Academic Press: New York, 1980; p 1.
(18) Aizawa, H. *Metabolic Maps of Pesticides*; Academic Press: New York, 1982.
(19) Anders, M. W. *Bioactivation of Foreign Compounds*; Academic Press: New York, 1985.
(20) Baeck, T. A note on Boltzmann Tournament Selection. *Complex Syst.* **1990**, *4*, 445−460.
(21) Michalewicz, Z. *Genetic algorithms + data structures = evolution programs*; Springer-Verlag: Berlin, New York, 1996.
(22) Hawkins, D. R. *Biotransformations: A Survey of the Biotransformations of Drugs and Chemicals in Animals*; The Royal Society of Chemistry: London, 1991; Vol. 1−3.
(23) William, B. J. *Enzymatic Basis of Detoxication*; Academic Press: New York, 1980.
(24) Belinsky, S. A.; White, C. M.; Trushin, N.; Hecht, S. S. Cell specificity for the pulmonary metabolism of tobacco-specific nitrosamines in the Fischer rat. *Carcinogenesis* **1989**, *10*, 2269−2274.
(25) Moorthy, B.; Madyastha, P.; Madyastha, K. M. Metabolism of a monoterpene ketone, R-(+)-pulegone --- a hepatotoxin. *Xenobiotica* **1989**, *19*, 217−223.
(26) Corey, E. J.; Wipke, W. T. Computer-Assisted Design of Complex Organic Synthesis. *Science* **1969**, *166*, 178−192.
(27) Gelernter, H.; Sanders, A. F.; Larsen, D. L.; Ararwal, K. K.; Boive, R. H.; Spritzer, G. A.; Searleman, J. E. Empirical Explorations of SYNCHEM. *Science* **1977**, *197*, 1041−1049.
(28) Varkony, T. H.; Smith, D. H.; Djerassi, C. Computer-Assisted Structure Manipulation Studies in the Biosynthesis of Natural Products. *Tetrahedron* **1978**, *34*, 841−852.
(29) Talafous, J. Studies in Computational Biochemistry: Computer Prediction of Xenobiotic Metabolism and the Three-Dimension Solution Structure of Residues 1−28 of the Alzheimer's Disease Amyloid Beta Peptide. Ph.D. Thesis, Case Western Reserve University, Cleveland, OH, 1995.

CI9601123