The Importance of Model Studies in Computational Organic Synthesis

H. W. Whitlock

Contribution from the Department of Chemistry, University of Wisconsin, 1101 University Avenue, Madison, Wisconsin 53706

Received August 28, 2000. Revised Manuscript Received December 5, 2000

Abstract: We explore a model, the "synthesis engine", for synthesis of arbitrarily complex organic structures in the context of library construction of Grignard cycle compounds. We use the simplest possible governing logic and show that random synthesis produces an extremely uneven distribution of products over several target structure types. We show that the question of "synthetic power" may be addressed computationally in this model system.

Introduction

That "combinatorial synthesis" is affecting a sea change in synthetic organic chemistry is apparent.[1−5] In this paper we describe and examine the logical basis of an alternative to the underlying idea of combinatorics or parallel synthesis. We first define a serial device (the synthesis engine)[6] that is capable of sequential synthesis of libraries[7−9] of compounds of some complexity. We show that, applied to random library synthesis, the simplest combination of logic and hardware leads to an extremely uneven distribution of library members. There is a surprising bias toward certain target compounds that follows simply from the probabilistic nature of the engine studied. The concept of synthetic method "power" as used by the engine is shown to be computationally well defined, and the effect of adding reactions to the engine's armory can be predicted by simulation of small data sets. The fairly obvious conclusion is drawn, that "random synthesis" is not a very discriminating tool. But another not so obvious conclusion also follows, that the infinite mesh of synthetic transformations has repeated regions of local order. The *form* of organic synthesis is not hopelessly complicated.

The Synthesis Engine

The basic idea of the synthesis engine is that of an intelligent chemical reactor that makes, one after another, target chemical structures according to its reaction library. It is a combination of logic (intelligence) and hardware that by programming, either implicit or explicit, synthesizes a defined set of product types. Especially, it is a *serial* device; one compound at a time is prepared. It (presumably) competes with parallel synthesis by virtue of its speed and generality.[10,11] In particular, the limitation of split-and-pool techniques, that all beads must be chemically compatible with each other, is no longer valid.[1,12] We define it in the following manner.

1. It has a set of one or more specified starting materials from which it synthesizes things.

2. It knows[13] a (finite) set of chemical reactions. These are of two types: unary operators (FGS, or functional group switching reactions)[2,14] that simply change functionality, and binary operators, reactions that condense two structures to form larger ones.

3. It has "memory". It can remember what it has synthesized, and how. It can thus recognize whether a prospective reaction will form a previously made product and can thus draw upon previously made products for use in condensation reactions.

4. It is "programmable". We use the simplest possible programming (random synthesis, effectively no program at all) so as to expose its innate behavior. It is interesting to note, however, that programmability ultimately includes the ability to simulate itself, and thus to incorporate these results into its programmed behavior.[15,16]

Its principle of operation is as follows.

[1] Schreiber, S. L. *Science* **2000**, *287*, 1964−1969.

[2] Terms used: "FGS", a reaction that interconverts functional groups with no increase in molecular complexity; "terminating structure", a predicate (e.g., "10-carbon aldehyde") that identifies target compounds for synthesis; "Markov", "Markovian", etc., any stepwise process associated with exploring discrete graphs; "recursive synthetic method", a condensation reaction wherein the reactant functionality is (can be) also present in the product; "list", a term having a rather specialized meaning in the computer sense but corresponding closely to our usual idea of a linear list of objects.

[3] Dolle, R. E.; Nelson, K. H., Jr. *J. Comb. Chem.* **1999**, *1* (4), 235−282.

[4] Lam, K. S.; Lebl, M.; Krchnak, V. **1997**, *97*, 411−448.

[5] Fu, G. C. *Acc. Chem. Res.* **2000**, *33* (6), 412−420.

[6] We use the term "engine" rather than "machine" to capture the idea of an underlying driver rather than a physical tool.

[7] There are a large number of commercially available devices and systems for library synthesis. It is important to note in this respect that the *synthesis engine*, as we define it, is purely a theoretical construct. Whether and to what extent it could be physically constructed remains to be seen, but its behavior as discussed here will certainly be associated with any reduction to practice.

[8] Zhao, C.; Shi, S.; Mir, D.; Hurst, D.; Li, R.; Xiao, X.-Y.; Lillig, J.; Czarnik, A. W. *J. Comb. Chem.* **1999**, *1*, 91−95.

[9] Nicolaou, K. C.; Xiao, X.-Y.; Parandoosh, Z.; Senyai, A.; Nova, M. P.; *Angew. Chem., Int. Ed. Engl.* **1995**, *34*, 2289.

[10] Tan, D. S.; Foley, M. A.; Shair, M. D.; Schreiber, S. L. *J. Am. Chem. Soc.* **1998**, *120*, 8565−8566.

[11] Lee, D.; Sello, J. K.; Schreiber, S. L. *J. Am. Chem. Soc.* **1999**, *121*, 10648−10649.

[12] Spaller, M. R.; Burger, M. T.; Fardis, M.; Bartlett, P. A. *Curr. Opin. Chem. Biol.* **1997**, *1*, 47−53.

[13] The word "know" alludes to its ability to apply both logic and chemical manipulation to reactions and reactants. It is a "robotic synthetic chemist" in an extremely limited sense.

[14] Whitlock, H. W. A Heuristic Solution to the Functional Group Switching Problem. *J. Am. Chem. Soc.* **1976**, *98*, 3225−3232.

[15] Any machine that is programmable in the Turing[16] sense must become theoretically intractable, simply because of its generality. Nonetheless, the underlying mesh (graph) of synthetic transformations remains unchanged, and it is ultimately *this* that defines the challenges of organic synthesis.

10.1021/ja003199o CCC: $20.00
Published on Web 02/15/2001

1. It examines the staring material and randomly selects one unused reaction (either a functional group switching[14] reaction, or a condensation) from its library. The reaction selected is marked “used”.

2. The reaction is carried out with one of two results: either it fails because of unexpected problems, or it succeeds. If the reaction fails, step 1 is repeated to select a new reaction, and the process repeats itself.

3. If the reaction succeeds, the product is examined. If it has been made in the current sequence of reactions (but *not* in a previous sequence), it is abandoned, and step 1 is repeated. If it is a new substance, it replaces the starting material and step 1 is repeated.

The engine stops under three different circumstances: there are no applicable reactions available that do not produce already-synthesized structures (the “no loop” requirement); mechanical or chemical failure of some sort occurs (a “bug” at some level); or an assigned limit of is reached, “no more than 12 carbons”, etc. (programmatic control). When a stopping state is reached, the engine saves and records the synthesis just completed and starts over again with its originally specified starting material. This process continues until user intervention occurs. It is useful (see below) to distinguish the compound last made when stopping occurs as a *terminating structure*.

As a tool for serial library construction, we define a set of chemical reactions, a set of starting materials, and a program that governs the engine’s behavior. The resulting set of terminating compounds constitutes the library. We consider here the simplest possible situation: the set of chemical reactions we refer to as the Grignard cycle; the simplest starting material, methanol; and the simplest possible program, a Markovian[2] search of synthesis space. The engine is logically both extremely simple and unintelligent and immensely powerful. It has no concept of goal achievement but mindlessly applies reactions until limits are reached. It has no concept of percent yield, although this can be added trivially. It represents the antithesis of the synthon[17] approach to synthesis. Curiously, the close coupling of reactions and logic suggests that the hardware implementation need not be chemically complex. While the chemical logic is simple, the compound types produced are not. That class of compounds produced by the Grignard cycle corresponds to a nested-parenthesis context-free[16] language and is thus unlimited in scope (correctly, countably infinite). This would seem to be a characteristic of that sparse set of synthetic methods that we can think of as “recursive”.[2] In itself this distinguishes the serial synthesis engine from parallel processes, which can form from a finite set of reactants only a finite set of products. In the context of current practice, the synthesis engine is probably closest to a diversity-oriented[1] or prospecting[12] tool.

Construction of the Synthesis Engine: YASS[18,19]

We simulate the synthesis engine by application of a programmable symbolic organic chemical synthesizer program, YASS. YASS is used to generate the recursive set of Grignard cycle structures, and these are then examined as random syntheses according to the above description of the synthesis engine. Standard counting observations are made of the nature of the (virtual) library produced.

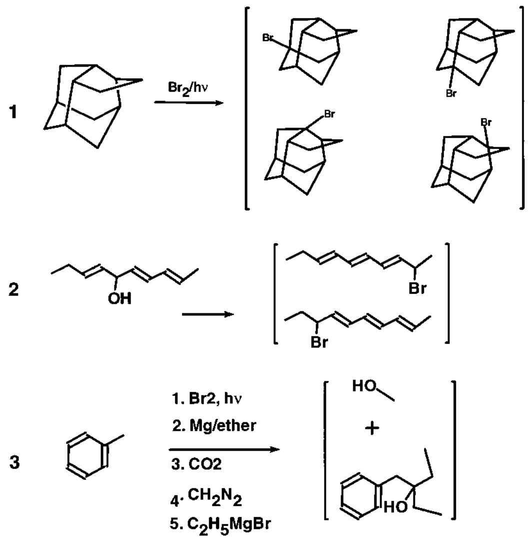


Figure 1. Typical YASS reactions. Case 1 illustrates structural isomer detection, case 2 thermodynamic capture, and case 3 sequential reactions with four unary and one binary operators.

Table 1. Functional Group Switching Reactions Used

reactant type	reagent	products
alcohol	HBr	alkyl halide[a]
alcohol	CrO_3/py	aldehyde or ketone
alkane	$Br_2/h\nu$	alkyl halide[b]
alkyl halide	Mg/ether	Grignard reagent
Grignard reagent	Br_2	alkyl halide
Grignard reagent	HBr, H_2O	alkane
Grignard reagent	CO_2	carboxylic acid
aldehyde or ketone	$NaBH_4$	alcohol
carboxylic acids	CH_2N_2	methyl ester

[a] Neopentyl alcohols are excluded. [b] Most substituted carbon is substituted.

Table 2. Condensation Reactions Used (the Carbonyl Reactant Is an Intermediate in Library Synthesis)

carbonyl structure	product formed
aldehyde or ketone	2° or 3° alcohol[a]
ester	3° alcohol
other	no product

[a] Rule of six applies.

YASS operates on two-dimensional structural diagrams of the standard organic type. Three-dimensional information is incorporated as the standard notation (wedges, etc.) and is used in answering such questions as chirality, *R*/*S*, or *Z*/*E*, configuration, etc. These configuration-knowledge features were turned off for this work. All structures in this paper were drawn by YASS, although we occasionally correct by hand its penchant for overlapping part structures. Figure 1 illustrates some of YASS’s abilities.

We equipped YASS with the functional group switching reactions shown in Table 1 and the condensation reactions shown in Table 2. Two condensation reactions, those of Grignard reagents with formaldehyde and ethylene oxide, were inserted as functional group switchers rather than the more general condensation reactions. Tables 1 and 2 comprise the unadorned[20] Grignard cycle of reactions familiar to all students of organic chemistry. These reactions are a built-in part of the

[16] For a discussion of theoretical issues surrounding “programmability”, see: Salomaa, A. *Formal Languages*; Academic Press: New York, 1973; pp 26−41 and Chapter 5.

[17] Corey, E. J.; Cheng, X.-M. *The Logic of Chemical Synthesis*; John Wiley: New York, 1989; Chapter 1.4

[18] Standing for “yet another symbolic synthesizer”.

[19] While YASS is at present incomplete, its internal structure representation is available as C++ source in the Supporting Information.

[20] This set of reactions was chosen since it defines a powerful but minimal Grignard reagent tool.

system; most of the interesting parts of the program were turned off in order to keep the chemistry and computing time manageable.

YASS is equipped with a collection of useful set-theoretic operations on sets of structures such as *union*, *intersection*, and *difference* functions, which produce results as lists of structures (it is a LISP-C++ hybrid). These operations are necessary to deal with the rather unwieldy numbers of structures produced.

We added several routines to YASS to enumerate all structures accessible from the designated starting material by application of the Grignard FGS[2] and condensation reactions (Tables 1 and 2). Each enumeration cycle of calculation had an upper limit of molecular size (carbon count) imposed. Chemical heuristics (see below) were necessary, since without these, many bizarrely hindered structures were produced. These heuristics are part of the underlying reaction machinery and were not added specially.

The set of all synthesizable structures was then converted to the equivalent Markov graph. This was then subjected to analysis by an iterative process that counts all possible syntheses together with their probability. This is, in a sense, the "point" of this paper, since it is *this* step that directly simulates the synthesis engine. We chose to break the structure generation (by YASS) and Markov calculations into two separate phases. The Markov path analysis could have been carried out simultaneously with structure synthesis, but the two-step procedure is both computationally more efficient and easier to understand. Otherwise the two procedures are equivalent.

Chemical Considerations

YASS has no intelligence per se. The first of its two types of reactions, functional group switchers, was implemented as a straightforward pattern match−replacement process. Without heuristics this produces an undesirable number of highly hindered structures. This was corrected for in the following manner. Reaction of alcohols with hydrogen bromide is essentially a textbook-level affair: primary and unhindered secondary alcohols give the corresponding bromides, as do tertiary alcohols. However, *neo*pentyl alcohols are made to fail.[21] Alkanes may be "radical-brominated", but this was limited to monobromination. The normal reactivity order was taken: benzylic > tertiary > secondary > primary (see Figure 1). Cases (e.g., *n*-pentane) that give multiple products were accepted on a "separate and purify" basis. Alkyl halides may be converted into generic Grignard reagents without limit. Grignard and organolithium reagents are not distinguished. Grignard reagents react with either water or HBr to form alkanes, with Br_2 to form alkyl bromides, with CO_2 to form carboxylic acids, and with the reagents formaldehyde and oxirane to form the expected primary alcohols. Reactions with other carbonyl compounds were treated as condensations. Aldehydes and ketones may be reduced with generic sodium borohydride, and carboxylic acids may be esterified with "diazomethane".

Condensation reactions were dealt with by classifying compounds as "reactive" or not. Only Grignard reagents are reactive in this set of compound types. Their chemical complement ("accepting") is the carbonyl group (aldehydes, ketones, and esters).[22] We incorporated Newman's rule of six[23] for condensation of Grignard reagents with aldehydes and ketones to keep things as reasonable as possible.

The structures produced by the above enumeration constitute a subset of the following compounds:

1. Acyclic alkanes.
2. Acyclic saturated primary, secondary, and tertiary alcohols.
3. Acyclic saturated primary, secondary, and tertiary alkyl halides. No distinction is made between the various halogens: all are "Br".
4. Acyclic saturated primary, secondary, and tertiary Grignard reagents, "RMgBr".
5. Acyclic saturated aldehydes and ketones.
6. Acyclic saturated carboxylic acids and their methyl esters.

Esters other than methyl esters are not included. Only aldehydes, ketones, and esters condense with Grignard reagents. Elimination reactions were excluded. Chirality was ignored, as were stereochemical consequences of the various reactions.[24] The set is a *subset* because of the chemical restrictions (*neo*pentyl alcohols, Newman's rule) placed on the reactions. That it is an infinite set follows from the observation that synthesis of primary alcohols RCH_2OH is in close correspondence to the corresponding context-free grammar for generation of R groups. This is an important feature of the synthesis engine and is responsible for its distinction from parallel library machines.

Operation of the Synthesis Engine

For a specified starting material (methanol), the synthesis engine prepares all accessible compounds having no more than a maximum number of carbons.[25] The result is a collection of compounds of varying size and type from which is picked the library. The simulation accomplishes this by a two-step affair: first we exhaustively generate all possible structures within the constraints set, and then we use this set to determine the probability of making any given compound or set of compounds.

Structure generation proceeds according to the following procedure. Starting with methanol, we generate all possible products by direct application of FGS reactions. We then carry out all possible condensation reactions with previously synthesized compounds. This process is continued until a previously specified maximum carbon count is reached. The synthesizable one-carbon structures produced were methanol, bromomethane, formaldehyde, methylmagnesium bromide, and methane. Condensation, followed by another round of functional group switching, gives the two-carbon set, which includes ethanol, bromoethane, acetaldehyde, ethylmagnesium bromide, acetic acid, and ethane. Similarly, another round of switching and condensation affords all three-carbon compounds.[26] The set of all 79 synthesizable compounds having five[27] or fewer carbons is shown in Figure 2.

Each synthesized compound has an associated set of synthetic connections. Figure 3 illustrates this for several Grignard reagents.[28] The behavior of compound **34** (*tert*-pentylmagnesium

[21] For forward synthesis "to fail" simply means "no product is possible".

[22] This is an attempt to minimize the combinatorial problems associated with binary reactions. As another example, dienophiles are "reactive", and dienes are "accepting" toward them.

[23] Newman's rule of six: "In reactions involving addition to an unsaturated function containing a double bond, the greater the number of atoms in the six position the greater will be the steric effect." *Steric Effects in Organic Chemistry*; Newman, M. S., Ed.; John Wiley and Sons: New York, 1956; p 206.

[24] For example, reduction of ketones with "NaBH4" gives a single secondary alcohol with no stereochemical properties.

[25] The limit of 12 or fewer carbons was chosen as a compromise between completeness and computational limitations on the machines available.

[26] Source code is available in the Supporting Information.

[27] The six- and eight-carbon sets of Grignard library structures are available in the Supporting Information. The latter, however, is somewhat artistically challenged (overlapping carbons).

[28] Output listing is available in the Supporting Information.

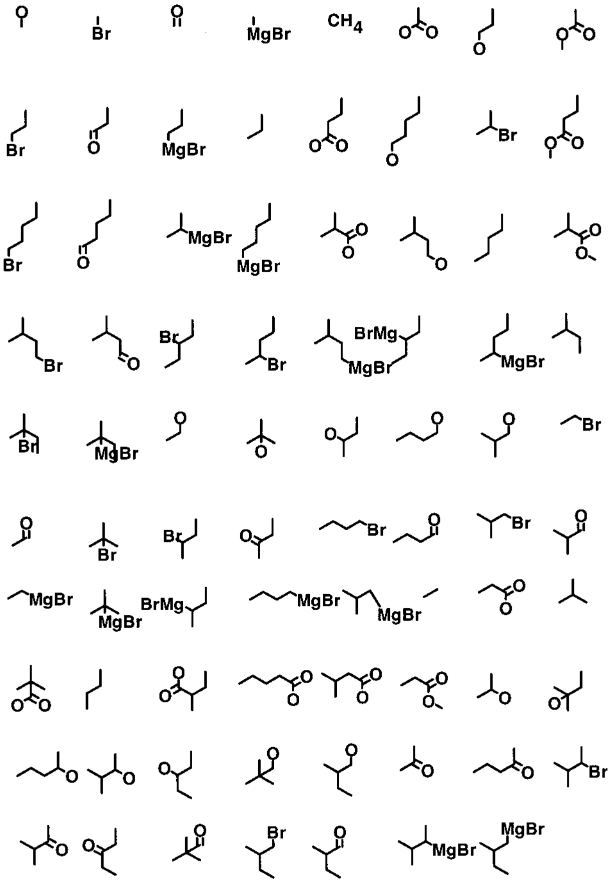


Figure 2. Five-carbon or less structures synthesized from methanol.

```
> Struc#33  C3H7MgBr
>   CondenseTo #11  #47
>   CondenseTo #3  #46
>   GoTo #37  H2O
>   GoTo #38  CO2
>   GoTo #37  HBr_Ionic
>   GoTo #25  Br2_CHCl3
>
> Struc#34  C5H11MgBr
>   GoTo #39  H2O
>   GoTo #39  HBr_Ionic
>   GoTo #27  Br2_CHCl3
>
```

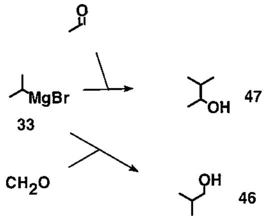


Figure 3. Partial output from structure enumeration, five or fewer carbons.

bromide) is truncated because these data were generated under the "five carbon or less" rule (see below).

Figure 4 portrays the intergenerational recursive nature of the structure generation. Functional group switching interconverts members of any generation; entry to the set is by condensation of members of a smaller set. Grignard acceptors are available for condensation with later generations. It is this feature that generates all possible R groups of any given carbon count. Table 3 shows the composition of the various libraries synthesized up through 12 or less carbons. Two columns are present: the first is the compound count using the Grignard cycle discussed above, and the second is the count when the reaction between Grignard reagents and oxirane is added to the reaction dictionary. Starting at six-carbon libraries, oxirane chemistry enhances the library composition (see below).

The synthesis graph is constructed from the data in Figure 3.[26] Each node of the graph corresponds to a structure, and two types of edges connect structure nodes: a functional group switching ("GoTo") edge is an ordered pair associated with an FGS reaction name. For example the pair {CH_4 CH_3Br} has the name "radical bromination". Two nodes may be connected by more than one directed edge if more than one reagent effects the conversion (RMgBr and RH are connected by both "HBr_Ionic" and "H_2O" edges.) Condensation is represented by a "CondenseTo" edge, which is an ordered triplet. The pair {2-propylmagnesium bromide, formaldehyde} is connected by a directed "CondenseTo" edge to isobutyl alcohol (see Figure 3).

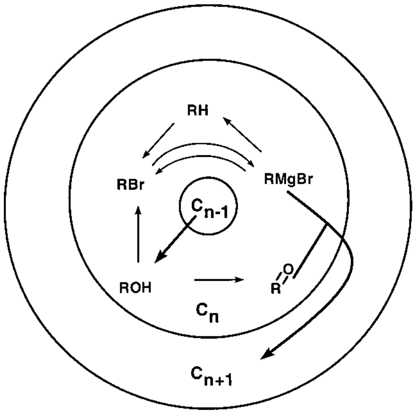


Figure 4. Illustrating the recursive nature of the Grignard cycle of reactions. The middle circle has *n* carbons; the inner circle *n* − 1; the outer *n* + *x* carbons. Light arrows are FGS reactions. Darker arrows are condensation reactions.

Table 3. Library Composition (Number of Unique Compounds Prepared) of Synthesizable Structures, Starting from Methanol and Having the Number Indicated or Fewer Carbons[a]

max carbons	library size (no oxirane)	library size (with oxirane)
1	5	5
2	11	11
3	22	22
4	42	42
5	79	79
6	151	158
7	299	333
8	605	721
9	1 256	1 594
10	2 660	3 581
11	5 727	8 146
12	12 498	18 766

[a] The column labeled "no oxirane" is for the reaction set shown in Tables 1 and 2. The column labeled "with oxirane" has the Grignard reagent−oxirane FGS reaction added.

A modified Markov[29,30] path algorithm accomplishes simulation of library construction:

a. We start with methanol as the initial node and with an empty synthetic path.

b. We create (by look-up) a list (**List1**) of reaction possibilities for the current node. These are the "GoTo" and "CondenseTo" entries shown in Figure 3.

c. A list[2] (**List2**) of candidates is constructed from **List1**. For each reaction on **List1**, if it is a condensation and the partner has been made, or if it is an FGS reaction and its product is not on the current synthetic path, it is added to **List2**. Otherwise, it is rejected.

d. If **List2** is not empty, one of its members, either a condensation or a switch, is chosen randomly, applied, and removed from **List2**. If the reaction fails, **List2** is reexamined. A reaction can fail by virtue of the rule of six, etc., or if the

[29] Norris, J. R. *Markov Chains*; Cambridge Series in Statistical and Probabilistic Mathematics; Cambridge University Press: Cambridge, UK, 1997.

[30] These are not properly Markov paths since state memory and "CondenseTo" nodes are present.

product has already been made in this cycle. All synthetic yields are either 0% or quantitative.

e. If the reaction succeeds, the product is added to the stored synthesis path. It becomes the current node, and the path growth process is repeated starting at step a.

Repeated application of this selection−application process is carried out until the process fails for one of several reasons:

1. No applicable reactions are possible (**List2** becomes empty at step d).

2. No condensations are possible, except they produce products that are “too big” by the carbon count restriction.

3. The user intervenes or other external restraints are applied.

At this point path growth ends, the end synthetic target is added to the library, the synthetic path found is recorded[31] in a dictionary, and the synthesis cycle restarts at step a to make a new random target. This iterative process is repeated until 50 000 consecutive cycles of terminating synthesis produce no change in the relative frequency of compound formation.[32] In the case of C-12 chemistry, termination occurred after approximately 5 × 10^8 iterations. At that time the dictionary contained 94 137 distinct synthesis paths.

Results of Emulation

Each cycle of the above iteration constitutes a “synthesis” and gives the following:

1. A list (synthesis path) of compounds prepared in that cycle. The first is the starting material methanol, and the last is the terminating structure for that cycle. A typical example is the following: “**1**,**2**,**4**,**21**,**33**,**44**,**50**”, a six-step synthesis of propane involving two condensations. This particular chain terminates because **33** is 2-bromopropane. Any path is stored in a dictionary[31] together with the number of times it is found. The full synthesis tree[17] for any target is constructed by string matching techniques as desired.

2. Each compound prepared has a counter (“FINDCOUNT”), which is the number of times it has been made.

3. Each compound also has a counter (“TERMCOUNT”), which is the number of times it appears in a synthesis path as the terminal target.

For any particular compound, the inequality [VISITCOUNT ≥ TERMCOUNT] holds, since preparation of a given compound may or may not terminate a synthesis. For example, in C-5 chemistry, the value of termcount is always zero for alkyl bromides. Five-carbon ketones have the two variables equal, since ketones are always made from the alcohol. Both butanes have different nonzero vales for the two counters since whether isobutene can go further depends on how it was made. No structures have both counters equal to zero (sanity check).

The interesting cases are those where [TERMCOUNT = VISITCOUNT]: all syntheses of a compound are terminal. These compounds, with suitable filtering out of undesired targets, constitute our library. Our initial thinking was that the distribution of libraries of a particular compound type (say, C-12 ketones) should prove to be uniform, since one ketone would be as likely to be made as another. But this is not the case. Quite the opposite is found, as the distribution of terminal targets of a particular type is extremely uneven.

Table 4. Six-Carbon Aldehyde−Ketone Library Composition Arranged in Decreasing Order of Frequency of Synthesis[a]

compound	TERMCOUNT	% of library
pinacolone	220 120	24.7
2,2-dimethylbutanal	155 343	17.4
3-methyl-2-pentanone	64 097	7.19
2-hexanone	60 516	6.78
4-methyl-2-pentanone	60 198	6.75
2-ethylbutanal	50 372	5.64
2-methylpentanal	50 168	5.62
3-methylpentanal	46 542	5.22
4-methylpentanal	46 450	5.21
hexanal	46 169	5.18
2,3-dimethylbutanal	46 186	5.18
2-methyl-3-pentanone	23 051	2.58
3-hexanone	22 855	2.56

[a] The oxirane-free Grignard chemistry (Table 1) was used.

Table 5. Six-Carbon Alcohol Library Composition Arranged in Decreasing Order of Frequency of Synthesis[a]

alcohol	VISITCOUNT	% of library
pinacolol	220 120	11.5
2,3-dimethyl-2-butanol	186 963	9.80
2-methyl-2-pentanol	183 026	9.60
2,2-dimethyl-1-pentanol	155 343	8.15
3-methyl-3-pentanol	129 638	6.80
3-methyl-2-pentanol	128 091	6.72
2-hexanol	120 711	6.33
4-methyl-2-pentanol	120 303	6.31
2-ethyl-1-butanol	100 393	5.26
2-methyl-1-pentanol	100 199	5.25
3-methyl-1-pentanol	93 144	4.88
4-methyl-1-pentanol	92 727	4.86
2,3-dimethyl-1-butanol	92 530	4.85
1-hexanol	92 323	4.84
2-methyl-3-pentanol	46 335	2.43
3-hexanol	45 326	2.38

[a] The oxirane-free Grignard chemistry of Table 1 was used.

Consider the C-6 case: there are 151 compounds having six or fewer carbons synthesized using the above Grignard cycle with the six-carbon restriction. The Markov exploration of the synthesis graph converges after 2.4 × 10^6 iterations, resulting in 689 different synthesis strings. Of the 151 terminal compounds, 13 are aldehydes and ketones (Table 4). As a group, they account for 37% of all syntheses.

Directly analogous results are obtained for C-6 alcohols (Table 5), except that none of these are terminal products. Because of the reactions used in the Grignard cycle, all primary and secondary alcohol products can be oxidized to the corresponding carbonyls. We see that the results in the two tables are quite similar. However, both library distributions are extremely uneven. A factor of 10 separates the probability of making pinacolone from 3-hexanone (Table 4), and a factor of 4.8 separates the corresponding alcohols. Clearly, as a tool for library construction, the random Markovian synthesis of targets is unacceptable.

A similar disparity in synthesis probability was found for the library of 12-carbon aldehydes. The most frequently synthesized aldehyde (148 255 times) was 2,2-dimethyldecanal, while the least frequently made was 2-(1-butyl)-3,4-dimethyl-hexanal (37 026 times), a ratio of most to least frequent of 100:25. Figures 5−8 summarize these results and show the four most and least likely synthesized 12-carbon aldehydes and ketones.

The case of 12-carbon compounds is similar. There are 12 498 distinct compounds prepared from methanol. Of these, 736 are ketones, and 598 are aldehydes. Markov path generation

[31] An “associative array” is used to record the synthetic sequence. This is a dictionary whose keys are synthetic reaction sequences, and values to be retrieved are frequency of application.

[32] Less than 0.1% change in the probability of finding any synthetic path (the ratio of the ct variable to the total number of iterations) after 50 000 iterations. The exact number of iterations varies according to seeding the random number generator involved.

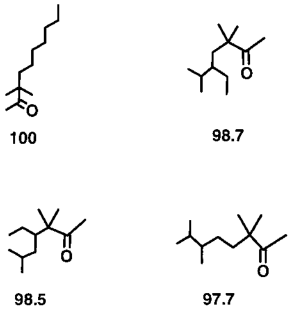


Figure 5. Four most likely 12-carbon ketones. The relative probability is shown.

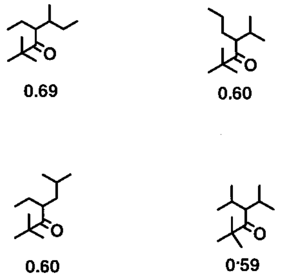


Figure 6. Least likely 12-carbon ketones. Their relative frequency of synthesis is shown.

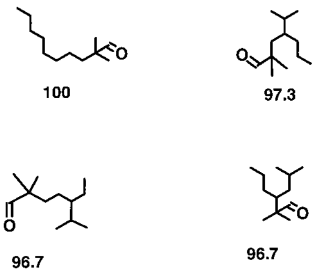


Figure 7. Most likely 12-carbon aldehydes, with relative frequency.

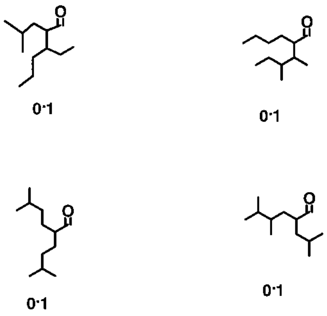


Figure 8. Least likely 12-carbon aldehydes with relative frequency of synthesis.

converged after approximately 2 × 10^8 syntheses. As discussed above, formation of a 12-carbon aldehyde or ketone leads to termination of the synthesis in this case. The most frequently synthesized ketone was found to be 3,3-dimethyl-2-decanone. It was made 223 440 times. At the other extreme was the least frequently synthesized ketone, 2,2,5-trimethyl-4-isopropyl-3-hexanone. It was made 1325 times. Thus, the ratio of most to least frequently synthesized ketones is the astounding (and potentially catastrophic) 100:0.59.

A similar wide disparity in synthesis probability was found for 12-carbon aldehydes. The most frequently synthesized aldehyde, made 148 255 times, was 2,2-dimethyldecanal, while the least frequently made was 2-(1-butyl)-3,4-dimethylhexanal, made 37 026 times for a ratio of most to least frequent of 100:25. Figures 5−8 summarize these results and show the four most and least likely synthesized 12-carbon aldehydes and ketones.

Table 6. Composition of Five-Carbon Carbonyl Library without (Column 2) and with (Column 3) Added Oxirane Grignard Reagent Reaction

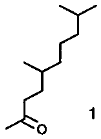
	% of library	
C-5 carbonyl	no oxirane	with oxirane
pivaldehyde	32.22	30.47
3-methyl-2-butanone	15.4	14.0
2-pentanone	14.7	14.6
2-methylbutanal	12.2	12.2
3-methylbutanal	10.9	11.8
pentanal	10.8	11.9
3-pentanone	3.8	4.9

Table 7. Synthesis Strings Leading to the More Probable Pivaldehyde (**32**) and the Least Probable 3-Pentanone (**52**)[a]

Synthesis Paths Leading to Pivaldehyde (**32**)
1, **2**, **4**, **20**, **25**, **33**, **46**, **53**, **64**, **17**, **12**, **14**, **24**, **32**
1, **2**, **4**, **20**, **26**, **9**, **12**, **14**, **24**, **32**
1, **2**, **4**, **6**, **7**, **9**, **12**, **14**, **24**, **32**
1, **2**, **4**, **8**, **10**, **13**, **22**, **28**, **35**, **37**, **25**, **33**, **46**, **53**, **64**, **17**, **12**, **14**, **24**, **32**
1, **2**, **4**, **8**, **11**, **20**, **25**, **33**, **46**, **53**, **64**, **17**, **12**, **14**, **24**, **32**
1, **2**, **4**, **8**, **11**, **20**, **26**, **9**, **12**, **14**, **24**, **32**
1, **2**, **4**, **9**, **12**, **14**, **24**, **32**
1, **3**, **22**, **28**, **35**, **37**, **25**, **33**, **46**, **53**, **64**, **17**, **12**, **14**, **24**, **32**
1, **3**, **24**, **32**
1, **3**, **46**, **53**, **64**, **17**, **12**, **14**, **24**, **32**
1, **3**, **8**, **10**, **13**, **22**, **28**, **35**, **37**, **25**, **33**, **46**, **53**, **64**, **17**, **12**, **14**, **24**, **32**
1, **3**, **8**, **11**, **20**, **25**, **33**, **46**, **53**, **64**, **17**, **12**, **14**, **24**, **32**
1, **3**, **8**, **11**, **20**, **26**, **9**, **12**, **14**, **24**, **32**
Synthesis Paths Leading to 3-Pentanone (**52**)
1, **2**, **4**, **8**, **10**, **13**, **22**, **29**, **45**, **52**
1, **2**, **4**, **8**, **10**, **13**, **45**, **52**
1, **3**, **22**, **29**, **45**, **52**
1, **3**, **8**, **10**, **13**, **22**, **29**, **45**, **52**
1, **3**, **8**, **10**, **13**, **45**, **52**

[a] The numbers correspond to structures shown in Figure 2.

The unbranched ketone 2-dodecanone is made with a middling relative frequency of 25, neither large nor small, as was the recognizable terpenoid, tetrahydrogeranylmethyl ketone (**1**), synthesized with a relative frequency also of ∼25.



1

That an even distribution of reaction path choices should lead to such an *uneven* distribution of target formation is not surprising. There are two important factors in determining whether a particular target is synthesized. First, a bushy synthesis tree has more synthesis paths from starting leaf (methanol) to target root and thus is relatively favored over one with fewer paths. Second, we must recognize that a synthesis tree in the context of enumeration is somewhat more complex than the common concept; it is an AND/OR tree, where OR nodes are alternative syntheses of a target or intermediate, and AND nodes are condensation reactions. The strong preference for formation of pivaldehyde over 3-pentanone (Table 6) is a reflection of there being 13 synthesis paths from methanol to pivaldehyde, but only five for 3-pentanone (Table 7). The intermediate 2-bromopropane may be produced from either 2-propanol or propane, and

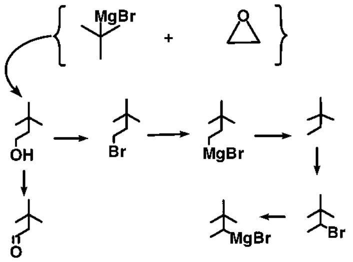


Figure 9. Seven six-carbon compounds uniquely synthesized when oxirane−Grignard condensation is present.

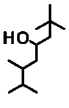
tert-butyl bromide from either *tert*-butyl alcohol or isobutane. It is not to say that making 2-bromopropane from propane is chemically smart, but that without added intelligence this will be tried. Lest one dismiss a reaction sequence such as {RBr → RMgBr → RH → R′Br} as redundant and dismissible, one should examine its power, as shown in Figure 9. We thus conclude that *an even distribution of selected synthetic path choices need not produce an even distribution of synthetic target formation.*

Computational Synthetic Power

We were curious about the effect of increasing the power of the synthetic methodology, so we added to the reaction dictionary that of oxirane with Grignard reagents to form β-substituted ethanols. Since the oxirane condensation is equivalent to two consecutive one-carbon extensions, we initially thought this would effect the proportioning of the library, but not its size. As expected, the wide disparities of probability evened out somewhat (Table 6), but we also find that the oxirane condensation permits synthesis of an entirely new set of terminal products. Without the oxirane reaction there are 151 different six-carbon compounds made; with the oxirane reaction there are 158. The seven newly accessible targets are shown in Figure 9, together with the synthetic interconversions involved.

Since our synthesis model has the Grignard condensation as the only carbon−carbon bond-forming reaction, it can make neopentyl alcohols, but further growth is blocked by its (imposed) inability to convert neopentyl alcohols into the corresponding bromides (see above). The "power" of the oxirane condensation can be removed by either removing the prohibition or adding a new FGS reaction ("TOSCl/NaBr/DMF"). The synthesis of 2-bromo-3,3-dimethylbutane in Figure 9 illustrates neatly the ease with which enumeration can be confused with creativity, since this is a very clever way to get to this bromide.

Great amplification of this effect is seen in 12-carbon library construction (see Table 3). With the oxirane condensation, there are now 18 766 compounds made by the synthesis engine (vs 12 498 when the oxirane reaction is absent.) Of these, 2087 are aldehydes or ketones, 753 of which are not synthesizable without the oxirane condensation reaction. Typical of the newly synthesizable structures is **2**. This has considerable import with respect to using the synthesis engine as a model for serial library construction, but its real value lies in the clear demonstration that addition of the oxirane condensation to the Grignard cycle of reactions is *not* just "two formaldehydes in a row". It increases the power of the Grignard cycle since with it compounds can be made that are inaccessible in its absence. Moreover, this computable power of the oxirane condensation does not require the large 12-carbon set to appear but is visible in much simpler compound sets, as demonstrated by simulation of model (e.g., six-carbon) compound sets. How can this be, when the synthesis mesh or graph is infinitely large? Infinite it may be, but local regions of order must occur and reoccur. This has to do with the finite structural context of reactions which, translated into graph terminology, corresponds to a repeated occurrence of local regions of order in an infinite graph.



2

Conclusions

The synthesis engine as described above is a rather minimal affair. It has memory and can thus determine whether it is about to prepare a known compound, but it is essentially a simple "finite-state"[16] machine constructed in a chemical context. The above analysis demonstrates that it has no "intelligence" as a synthesizing tool. With its use we have shown that some of the basic properties of random library construction must be surmounted. But on the other hand, intelligent design in the form of evaluation and selection functions is easily added since our simulation is exactly how the machine would work. The only added step needed in its construction is the ability to carry out the transformations. We suggest that the underlying synthesis engine is of some interest as a simple and tractable mathematical model of the organic synthesis of complex organic compounds.

The importance of model reactions in this study is both surprising and gratifying: surprising because of its close similarity to our chemical emphasis of model studies; gratifying because exhaustive calculations such as those described above cannot be carried out on arbitrarily large and complex compound sets.

Particularly interesting is the observation that the oxirane condensation (in the chemical context used) unequivocally enhances the power of the Grignard cycle of reactions. The extent that this technique can be used in a real non-toy case will rest on the issues of computational power and validity of our suggested importance of model studies.

Supporting Information Available: C++ source of the YASS internal structure representation; source code; six- and eight-carbon sets of Grignard library structures; and output listing (PDF). This material is available free of charge via the Internet at http://pubs.acs.org.

JA003199O