SUPPORTING INFORMATION

for

Automated Lead Optimization of MMP-12 Inhibitors Using a Genetic Algorithm

Stephen D. Pickett[†*], Darren V.S. Green[†], David L. Hunt[§], David A. Pardoe[‡], Ian Hughes[‡]

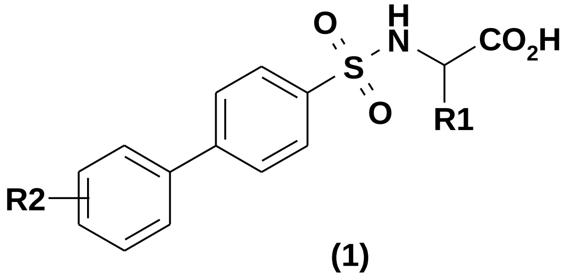
[†]GlaxoSmithKline Research and Development, Stevenage, Herts. SG1 2NY.

[‡]GlaxoSmithKline Research and Development, Harlow, Essex. CM19 5AW.

[§] Tessella plc, Stevenage, Herts. SG1 2EF.

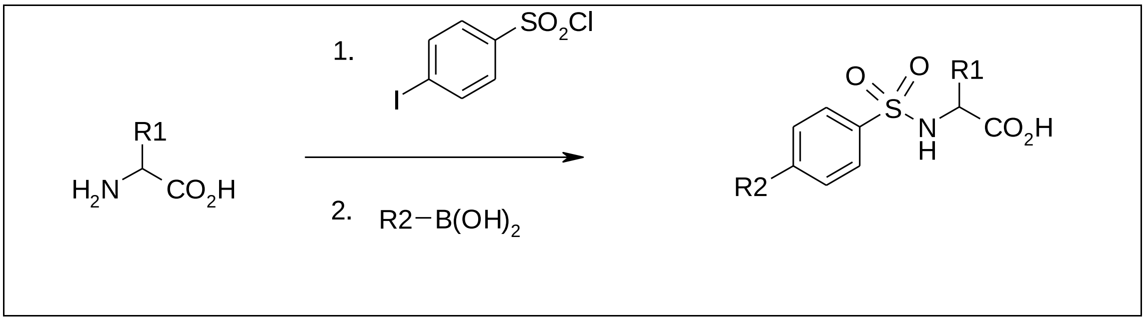
S1. Synthetic Chemistry

Existing in-house screening data had identified biaryl sulfonamides represented by **1** as a series with MMP-12 activity and amenable to chemistry from readily available starting materials, amino acids and boronic acids. A 50x50 array was synthesized as described below using the reagents shown in Table 1.



(1)

The majority of the products were obtained in a two step process from a series of amino acid and boronic acid building blocks as illustrated in Scheme 1. The amino acid was coupled with pipsyl chloride and the resulting iodide was reacted with the boronic acid under Suzuki conditions. In most cases the first step was outsourced to give suitable quantities of the sulfonamide to allow individual in-house Suzuki reactions with fifty boronic acids. All products were purified via preparative HPLC and characterized by NMR and MS.

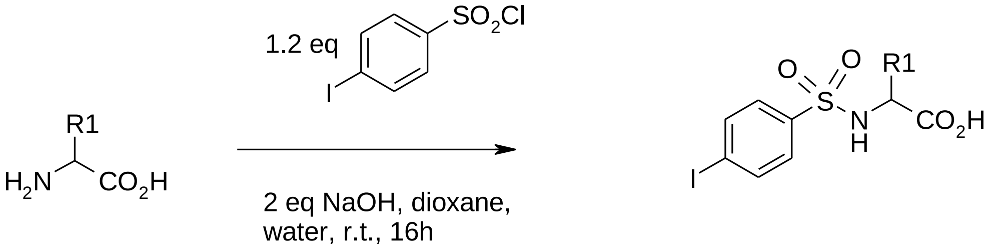


Scheme 1. Standard synthetic route for the biaryl sulfonamide array.

It is common practice to adopt a single route for parallel synthesis of arrays, for reasons of efficiency and economy. However, since the objective of this exercise was to obtain as diverse and complete an array as possible, variations of the standard route and conditions were required to avoid compromising the diversity of the final array. Despite best efforts, a number of the desired products could not be synthesized due to poor reactivity, unanticipated side reactions or product instability.

Synthesis of sulfonamides:

The standard route involved reaction of the amino acid with pipsyl chloride as shown in Scheme 2.



Scheme 2. Standard route for synthesis of sulfonamides

The majority were synthesized by this route, with the following exceptions:

A10: Not made

A12, A35: As standard route, but 0 C, 6h

A1. The phenol group required protection as the t-butyl ether, to prevent O-sulfonylation. The phenol was deprotected prior to the Suzuki coupling step.

A14. A milder base (Et3N) was used for the sulfonylation to prevent hydrolysis of the ethyl ester.

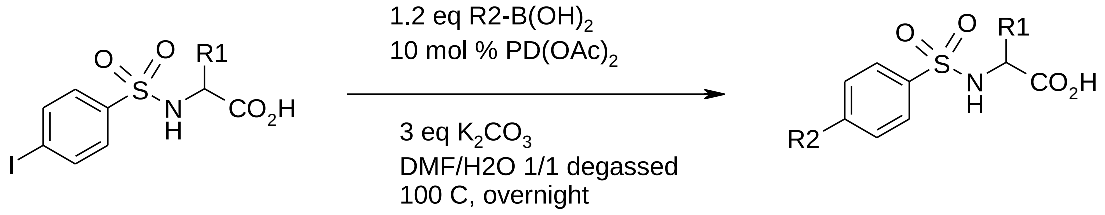
A15. The side-chain amino group of the amino acid was protected with a Cbz group during the sulfonylation and Suzuki steps, deprotected by hydrogenation, then trifluoroacetylated.

A22. The phenol was allowed to react with excess pipsyl chloride during the sulfonylation step and was cleaved by base hydrolysis.

A45. The Boc protected lysine methyl ester was sulfonylated then cleaved to the free amine which was coupled to pyridine-3-carboxylic acid before finally hydrolyzing the ester group.

Suzuki coupling reactions

In most cases the iodo sulfonamides were coupled to the boronic acid building blocks using Suzuki conditions as shown in Scheme 3.



Scheme 3. Standard route for Suzuki coupling

Although the majority of couplings were performed this way, the following were exceptions:

A2, A3, A12: used PPh_3 route (see Scheme 4)

A10: monomer was never made

A13, A23: failed due to cyclization reactions

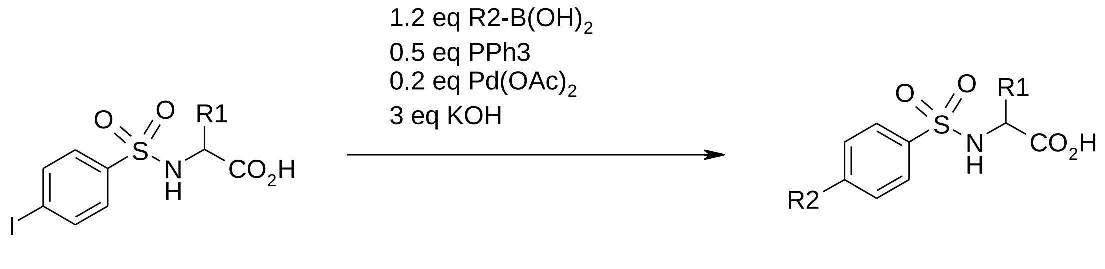
A14: heated for only 3 h to prevent ester hydrolysis

A15: a different route was required, as already described above

A20: failed to couple under a range of conditions

A26: The Suzuki reaction was performed on the methyl ester of the sulfonamide via the PPh_3 route (Scheme 4), and was hydrolyzed in the final step

A36: was unstable – dihydrobenzene ring oxidized readily



Scheme 4. Suzuki coupling using PPh_3

B9, B12, B28, B45: failed purification

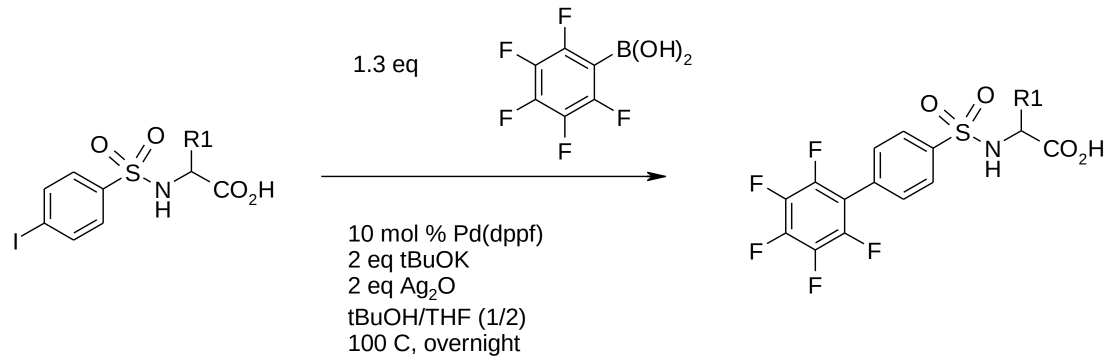
B2: Because this product possessed an aromatic bromo group which interfered in the Suzuki reaction, these products were made directly by coupling the bromo biphenyl sulfonyl chloride with the amino acid building blocks.

B26: Coupling of the perfluorophenyl boronic acid proved particularly troublesome, and the Suzuki conditions shown in Scheme 5 were found to be the most effective.

B30: The nitro functionality was introduced at the final step by condensation of nitromethane with the biphenyl aldehyde.

B41, B47: only 3 h heating to stop ester hydrolysis

B50: never coupled successfully



Scheme 5. Route for B26

Reagent selection

At the time that this experiment was started there were about 450 commercially available boronic acids and 680 commercially available α-substituted amino acids. These lists were reduced to give a 50x50 array using our in-house library design system, ADEPT[1]. A diverse set of reagents was selected using clustering and similarity analysis with Daylight fingerprints.[2]

Full details of the chemical structures (SMILES strings[2]), QC data from LC/MS and biological data in the MMP-12 assay are given in the excel file included with the supporting information. In addition NMR data on several of the most active compounds are given below.

All nmr data were measured at 400 Mhz on a Bruker DPX400.

A18B18 [pIC50 = 7.4]

^{1}H NMR (CD_3OD) δ 0.96 (3 H, t, *J* = 8 Hz), 1.40 (9 H, s), 1.67 (2 H, m), 2.63 (2 H, t, *J* = 8 Hz), 2.60 (1 H, dd, *J* = 6, 16 Hz), 2.68 (1 H, dd, *J* = 6, 16 Hz), 4.21 (1 H, t, *J* = 6 Hz), 7.28 (2 H, d, *J* = 8 Hz), 7.60 (2 H, d, *J* = 8 Hz), 7.77 (2 H, d, *J* = 9 Hz), 7.92 (2 H, *J* = 9 Hz)

A01B30 [pIC50 = 7.3]

^{1}H NMR (CD_3OD) δ 2.73 (1 H, dd, *J* = 8, 14 Hz), 2.97 (1 H, dd, *J* = 6, 14 Hz), 3.97 (1 H, dd, *J* = 6, 8 Hz), 6.58 (2 H, d, *J* = 9 Hz), 6.93 (2 H, d, *J* = 9 Hz), 7.67 (4 H, m), 7.78 (4 H, m), 7.94 (1 H, *J* = 15 Hz), 8.10 (1 H, d, *J* = 15 Hz)

A04B04 [pIC50 = 7.2]

^{1}H NMR ($CDCl_3$) δ 0.79 (3 H, d, *J* = 8 Hz), 0.85 (3 H, d, *J* = 8 Hz), 1.38 (2 H, m), 1.72 (1 H, m), 2.40 (3 H, s), 3.90 (1 H, m), 5.08 (1 H, br.d, *J* = 9 Hz), 6.10 (1 H, br.s), 7.28 (2 H, d, *J* = 9 Hz), 7.47 (2 H, d, *J* = 9 Hz), 7.63 (2 H, d, *J* = 9 Hz), 7.82 (2 H, d, *J* = 9 Hz)

A09B48 [pIC50 = 6.1]

^{1}H NMR (d_6DMSO) δ 2.27 (3 H, s), 3.08 (1 H, m), 3.24 (1 H, m), 3.73 (1 H, m), 5.57 (2 H, s), 6.03 (1 H, t, *J* = 7 Hz), 7.38 (1 H, t, *J* = 9 Hz), 7.48 (1 H, dd, *J* = 2, 8 Hz), 7.53 (1 H, dd, *J* = 2, 11 Hz), 7.79 (2 H, d, *J* = 9 Hz), 7.85 (2 H, d, *J* = 9 Hz), 8.15 (1 H, br.s), 12.8 (1 H, br.s)

A24B15 [pIC50 = 5]

^{1}H NMR (CD_3OD) δ 2.04 (1 H, m), 2.25 (1 H, m), 2.60 (3 H, d, *J* = 7 Hz), 2.80 (1 H, m), 2.92 (1 H, m), 4.03 (1 H, m), 7.25 (2 H, m), 7.43 (1 H, m), 7.53 (1 H, td, *J* = 8, 2 Hz), 7.72 (2 H, d, *J* = 8 Hz), 7.94 (2 H, d, *J* = 8 Hz)

A04B07 [pIC50 = 7.8]

^{1}H NMR (d_6DMSO) δ 0.68 (3 H, d, *J* = 7 Hz), 0.78 (3 H, d, *J* = 7 Hz), 1.37 (2 H, m), 1.57 (1 H, m), 2.49 (3 H, s), 3.64 (1 H, m), 7.36 (2 H, d, *J* = 9 Hz), 7.67 (2 H, d, *J* = 9 Hz), 7.78 (2 H, d, *J* = 9 Hz), 7.83 (2 H, d, *J* = 9 Hz), 8.08 (1 H, br.s), 12.70 (1 H, br.s)

A25B19 [pIC50 = 7.5]

^{1}H NMR (CD_3OD) δ 0.98 (9 H, s), 1.25 (3 H, t, *J* = 8 Hz), 2.68 (2 H, q, *J* = 8 Hz), 3.30 (1 H, m), 7.30 (2 H, d, *J* = 9 Hz), 7.58 (2 H, d, *J* = 9 Hz), 7.74 (2 H, d, *J* = 9 Hz), 7.88 (2 H, d, *J* = 9 Hz)

A04B18 [pIC50 = 7.2]

^{1}H NMR ($CDCl_3$) δ 0.82 (3 H, d, *J* = 7 Hz), 0.88 (3 H, d, *J* = 7 Hz), 0.98 (3 H, t, *J* = 8 Hz), 1.42 (2 H, m), 1.70 (2 H, m), 1.73 (1 H, m), 2.65 (2 H, t, *J* = 8 Hz), 3.94 (1 H, m), 4.50 (1 H, br.s), 5.05 (1 H, d, *J* = 9 Hz), 7.28 (2 H, d, *J* = 9 Hz), 7.50 (2 H, d, *J* = 9 Hz), 7.67 (2 H, d, *J* = 9 Hz), 7.84 (2 H, d, *J* = 9 Hz)

A49B04 [pIC50 = 6.8]

^{1}H NMR (CD_3OD) δ 2.37 (3 H, s), 3.35 (1 H, dd, *J* = 8, 15 Hz), 3.60 (1 H, dd, *J* = 6, 15 Hz), 4.27 (1 H, m), 7.28 (2 H, d, *J* = 8 Hz), 7.41 (1 H, d, *J* = 4 Hz), 7.53 (2 H, d, *J* = 8 Hz), 7.63 (1 H, d, *J* = 4 Hz), 7.68 (2 H, d, *J* = 9 Hz), 7.84 (2 H, d, *J* = 9 Hz)

A50B18 [pIC50 = 6.8]

^{1}H NMR (d_6DMSO) δ 0.89 (3 H, t, *J* = 8 Hz), 1.60 (2 H, m), 2.58 (2 H, t, *J* = 8 Hz), 2.78 (1 H, dd, *J* = 7, 14 Hz), 2.98 (1 H, dd, *J* = 6, 14 Hz), 3.74 (1 H, m), 6.99 (2 H, m), 7.31 (2 H, d, *J* = 9 Hz), 7.60 (1 H, m), 7.63 (2 H, d, *J* = 9 Hz), 7.74 (2 H, d, *J* = 8 Hz), 7.79 (2 H, d, *J* = 8 Hz), 8.33 (1 H, br.s), 13.00 (1 H, br.s)

S2. MMP-12 assay

The MMP-12 assay was performed on recombinant human MMP-12 catalytic domain, residues G106-N268. The recombinant protein was synthesized in-house. A standard FRET based assay was employed with 6nM MMP-12 and 2μM of a FAM-TAMRA substrate FAM-Thr-Pro-Gly-Pro-Leu-Gly-

Leu-X-Ala-Arg-Arg-Lys-TAMRA synthesized in-house. X is 3-[(2,4-dinitrophenyl)amino]-L-alanine. The assay buffer contained 50mM HEPES, 150mM NaCl, 10mM $CaCl_2$, 1µM $ZnAc_2$ and 600µM CHAPS (all purchased from commercial sources); pH7.5. The assay was incubated at room temperature for 2 hours, in the dark, and the reaction terminated by the addition of 100mM EDTA. Plates were read with a PheraStar (BMG Labtech) platereader (Ex of 485nm, Em of 530nm, Dichoic Filter of 505nm).

All results are an average of at least 2 determinations.

S3. Optimisation Algorithm

We made the decision to use a genetic algorithm based optimizer to drive each iteration cycle, similar in principle to previous publications.[3, 4] However, the algorithm incorporates a number of important additional features. In the language of the GA, each reagent is an allele in a specific gene (reaction position). The complete library, or genome, is the collection of genes. The population is the set of molecules (individuals) that have been or are to be made. To start the process the user provides the algorithm with the genome and various optimization parameters defined below. The algorithm generates the starting population (each molecule being a combination of alleles from each gene) and the compounds are synthesized and tested. The population is updated with the screening results (scores) and a new population generated using genetic operators. The process is repeated until convergence or a user-defined number of iterations has been reached.

Genes, Genomes and Population

Each gene represents a position in the combinatorial library. Each reagent at this position (allele) is represented by some unique label, e.g. compound identifier. In this implementation the genes are files and the genome is simply a list of these files. The population is a list of specific individuals from the genome. So for a two component library with reagents A1 to A10 and B1 to B10 a valid individual is A3-B5. No attempt is made to generate a combinatorial subpopulation at each iteration.

Parent selection

A key part of the genetic algorithm is the way in which it chooses which individuals from the population of tested data to use as parents for generating the next generation of test data. This process involves choosing individuals at random but with a probability that is weighted towards the fittest. In this application we use a roulette wheel with the proportion of the wheel based on an individual's fitness (activity). This is done in two stages. Two roulette wheels are generated, one for the latest test generation ("new") (note: the definition of last test generation is user controllable, by defining the number of previous generations to be considered as part of the latest generation. This is done so that small numbers can be tested at each generation, e.g. 1, but without affecting the ability to have a large population "exploring" the search-space) and one for the individuals tested in the generations prior to current generation ("old").

The roulette of "old" individuals then removes all individuals except the N fittest, where N is controlled by a user-defined proportion times the size of the "new" roulette wheel. This provides a strong pressure to search known maxima. This is not done to the "new" roulette, as the "new" individuals are more diverse in order to ensure that there is a robust search of the space for undiscovered maxima.

Both the “old” and the “new” roulette wheels are then “normalized” so that the probability of the fittest being selected is greater than the probability of the weakest being selected by a user-defined amount. The probability for the other individuals is a linear function of fitness – not rank order. This is done in the expectation of finding islands of activity, forcing the algorithm to explore these islands requires a strong selection pressure relative to the surrounding inactive region.

Once the roulette wheels have been set up, the second stage of the selection process can occur. When a parent is required, an individual is selected randomly from either the old or the new roulette wheel, again governed by a user-defined probability. Once a wheel is chosen, an individual is chosen by “spinning” the wheel to select a member of the population.

Genetic operators

The GAO library has been written to be problem agnostic, that is it can be applied in principle to any optimization problem. For example, several genetic operators have been implemented for numerical optimization problems. In this section we describe the operators relevant to this particular application.

Cross-over takes two parents and creates a child with a mixture of the parents’ genetic information. For each gene in turn the child allele is chosen at random to be either of the parent alleles.

Mutation takes one parent, where each gene either randomly mutates or takes the parent’s value. The probability of the gene mutating is such that on average, only one gene in the individual will change.

Random generates an individual at random.

The relative proportion of cross-over, mutation and random is user controlled.

Choosing the next test set

The next generation is chosen in the following manner

1. The size of the next generation, nNew, is kept constant as defined by the initial population size, set to a user-defined value or can be variable. In the latter case the generation size is set to the size of the previous generation, nOld multiplied by the sum of the proportion of cross-over, mutation and random.

2. A roulette wheel is generated with segments for cross-over, mutation and random. The weight of each segment is set to the proportion of the next generation to be generated by the corresponding method. The wheel is “spun” nNew times, each time a child is chosen, then a child is generated by cross-over, etc. This is a rather complex way of choosing the

method of generating the next generation. However, the use of a roulette wheel allows the size of each generation to be as low as one.

3. The parents for the cross-over and mutation are selected as described above.

4. Each individual chosen by the above methods is a new individual in the population. When an individual is chosen that is already in the population, then the process is repeated with a different set of parents. Once every possible individual has been added to the population, then the process ends.

   When trying to create an individual by cross-over, if after a large number of attempts it is not possible to generate a new individual, then an individual is created by localized mutation.

Novel aspects of the algorithm

Retesting individuals

Any experimental process, be that synthesis or assay will be prone to errors. Similarly, certain molecules may not be synthesizable. Thus a novel aspect of the algorithm described here is to allow for these potential errors in the optimization process.

At each iteration, those individuals that were selected to be a parent are considered for re-testing. If the parent has been tested less than a user-defined number of times then it will be flagged for re-testing.

Seek mutation

A proportion of random individuals can be a good way of maintaining diversity within the population. Within the context of a combinatorial library it is possible to identify regions of the space that have been poorly sampled. Thus, as an alternative to the random operator, a seek operator has been implemented that identifies such regions and focuses the random generation in this region.

NULL individuals

If GAO requests a test and there are no results available for the test, then the user can do one of three things;

1. Add no entry for the test in to the fitness file

2. Add an entry with fitness data

3. Add an entry with "<NULL_FITNESS>" as the fitness entry.

In all cases, GAO will keep the individual for retesting and will request that it be tested again at some future generation.

All untested individuals will be stored in the population file as individuals without fitness data.

If the user wishes to indicate that the test cannot be performed and that GAO should not request a re-test then the user marks the individual with "<NULL_INDIV>" as the fitness.

Similarity and Grouping

As described above the algorithm knows nothing about the chemical composition of the individual reagents (alleles) or the products. Thus we have implemented the ability to define similarity between alleles. The similarity is defined in terms of distance and need not be symmetric. In addition alleles may be grouped. Two alleles that are not in the same group have a similarity of zero. An allele may belong to more than one group. There are many ways of defining similarity between reagents and clustering or grouping compounds and the implementation leaves it up to the user to decide on the most appropriate criteria to use, based upon the problem at hand. For instance, a group of reagents may all contain a particular functional group four bonds from the attachment point or may be in the same cluster according to a particular set of chemical descriptors.

When enabled similarity and grouping leads to modified selection conditions for cross-over and mutation.

For cross-over the child allele is an allele that is similar to both parent alleles. A child allele is chosen at random from all the alleles that satisfy the following conditions;

Sim (P1,C) >= Sim (P1,P2)

Sim (P2,C) >= Sim (P2,P1)

Sim (P1,C) > 0

Sim (P2,C) > 0

In addition, when grouping is enabled:

- if one parent is not in a group or if they are not in a common group, then the child allele can only be one of the parent alleles.

- if the parent alleles are in the same one group that is not overlapped, then either parent allele or alleles in the that group could be chosen

- where groups are overlapping, the situation is more complicated as the similarity between alleles from grouping is additive.

In the case of mutation each gene is considered in turn. The new allele is an allele that is similar to the parent allele. The new allele is chosen by jumping a distance from the parent allele. The maximum jump distance is given by the formula;

MaxJumpDistance = $N_{alleles}$ * Factor

$N_{alleles}$ is the number of alleles in the gene.

Factor = JumpFactor * AnnealingRate $^{generation-1}$

JumpFactor is a scaling factor to increase the range of the distance jumped. AnealingRate specifies the rate at which the jump distance will decay with each generation focusing the search to a smaller region as the search progresses.

A jump allele is chosen at random from all the alleles that are within the specified distance of the parent allele (given the relationship between similarity and distance above).

If no similarity is defined the child allele is set to a random value with a probability of 1/N_{genes}, where N_{genes} is the number of genes in the genome so that on average only one allele within an individual generated by the mutation mechanism will change randomly.

It is possible to specify the similarity between products rather than between the individual alleles. Normally the similarity between products is implied by the constituent alleles that define the product. For instance, for a set of genes A and B with unordered alleles;

<A1|B1> is similar to <A1|B2> because they share the same A allele.

<A1|B1> is not similar to <A2|B2> because they do not share a common allele.

However, the user can specify product similarity explicitly. This would allow the user to specify that <A1|B1> and <A2|B2> are similar.

In order to do this, the user must choose whether they wish to keep the implicit similarity of the underlying gene structure.

Product similarity can be specified in two ways;

1) Grouping, products can be grouped in the same way as alleles.

2) Similarity between products can be specified in the same way as alleles, e.g.;

<Similar;A1|B1;A2|B2,0.5>

<Similar;A2|B2;A1|B1,0.5>

Both “similar” and “group” can be used together. It should be noted that the similarity specified by these commands is added to the implied similarity. It does not over-write it.

Learning the SAR

If the user has defined an initial set of groupings or similarities how do these relate to the actual biological data is obtained? Is it possible to learn appropriate groupings and thus generate an implied SAR from the data? In order to answer these questions options have been added to re-evaluate the similarity levels associated within the groups at regular intervals during the optimization. The algorithm uses the standard deviation of fitness within the group to redefine the group similarity.

References

1. Leach, A. R.; Bradshaw, J.; Green, D. V. S.; Hann, M. M.; Delany, J. J. Implementation of a system for reagent selection and library enumeration, profiling and design. *J. Chem. Inf. Comput. Sci.* **1999,** *39*, 1161-1172.

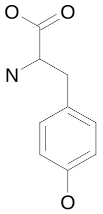
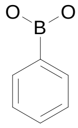
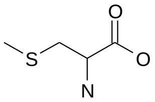
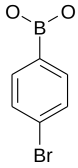
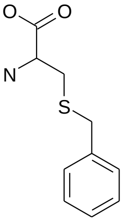
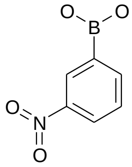
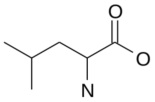
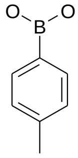
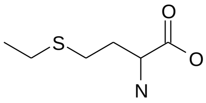
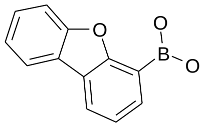
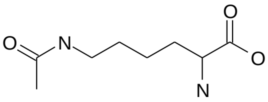
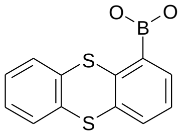
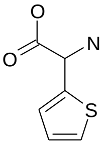
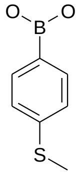
2. James, C. A.; Weininger, D.; Delaney, J. *Daylight Theory Manual, Daylight Chemical Information Systems.* http://www.daylight.com/dayhtml/doc/theory/index.html **2005**.

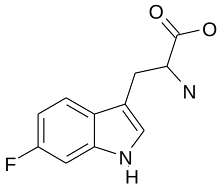
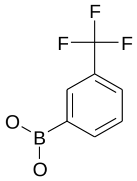
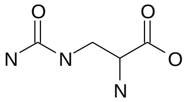
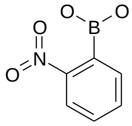
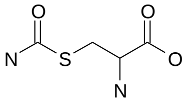
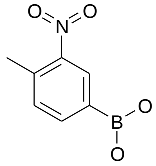
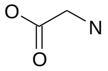
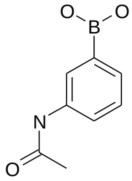
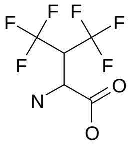
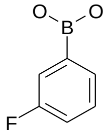
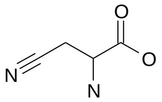
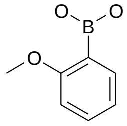
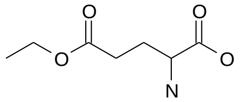
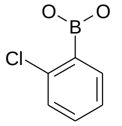
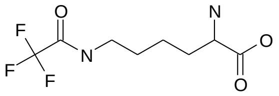
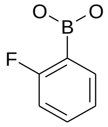
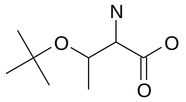
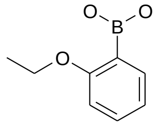
3. Weber, L.; Wallbaum, S.; Broger, C.; Gubernator, K. Optimization of the Biological Activity of Combinatorial Compound Libraries by a Genetic Algorithm. *Angew. Chem. , Int. Ed. Engl.* **1995,** *34*, 2280-2282.

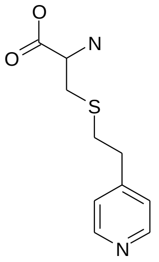
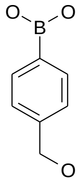
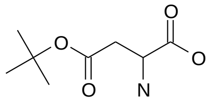
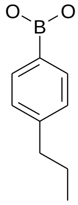
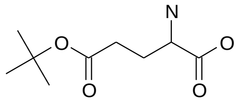
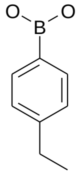
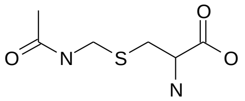
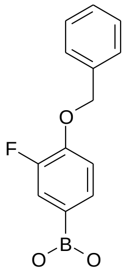
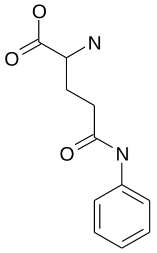
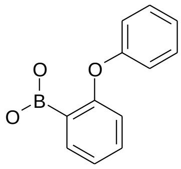
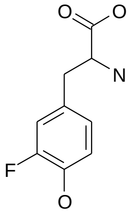
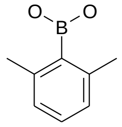
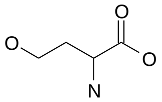
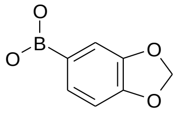
4. Singh, J.; Ator, M. A.; Jaeger, E. P.; Allen, M. P.; Whipple, D. A.; Soloweij, J. E.; Chowdhary, S.; Treasurywala, A. M. Application of Genetic Algorithms to Combinatorial Synthesis: A Computational Approach to Lead Identification and Lead Optimization. *J. Am. Chem. Soc.* **1996,** *118*, 1669-1676.

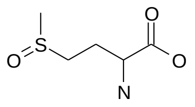
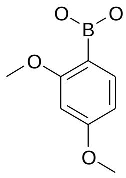
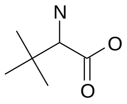
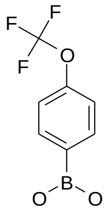
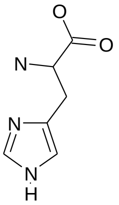
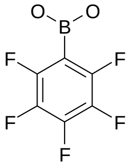
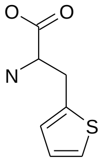
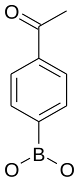
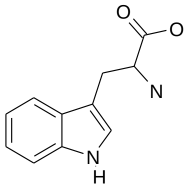
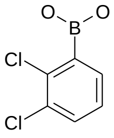
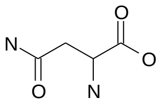
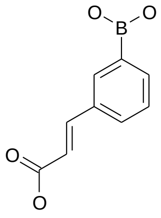
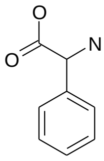
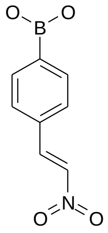
Tables

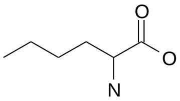
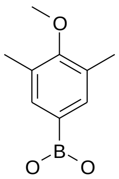
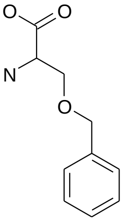
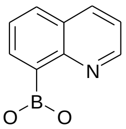
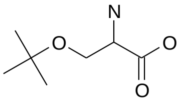
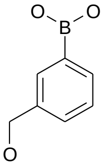
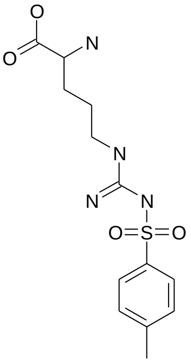
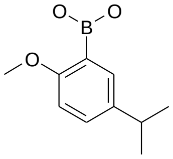
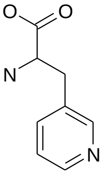
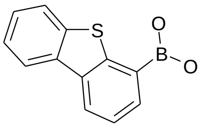
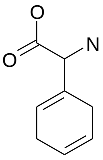
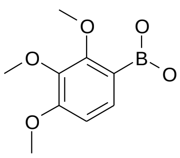
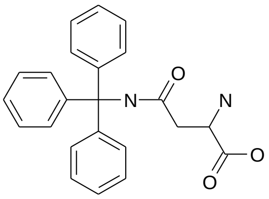
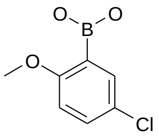
Table 1. 50 Amino acids (Ann) and 50 Boronic acids (Bnn) selected for inclusion in the 50x50 array based on chemotype 1.

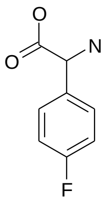
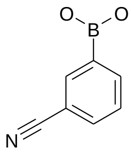
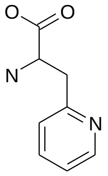
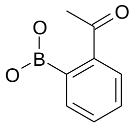
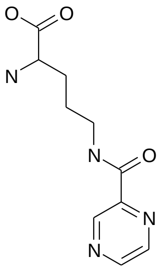
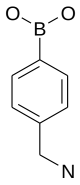
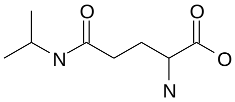
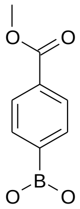
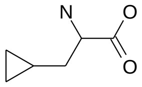
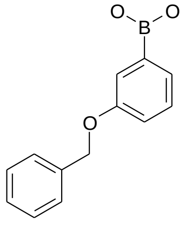
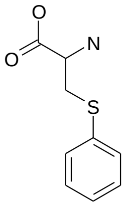
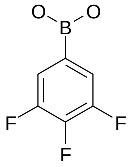
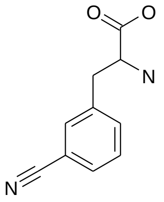
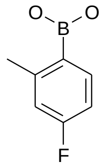
Reagent A	Structure	Reagent B	Structure
A01		B01	
A02		B02	
A03		B03	
A04		B04	
A05		B05	
A06		B06	
A07		B07	

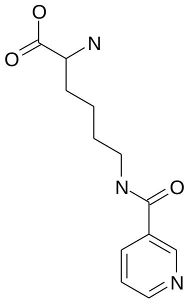
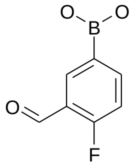
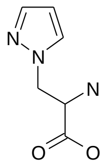
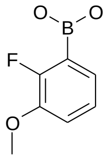
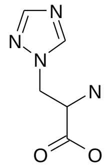
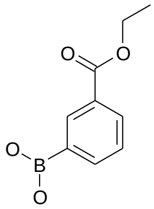
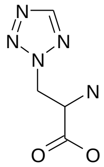
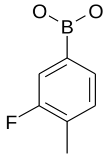
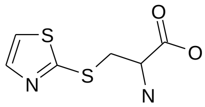
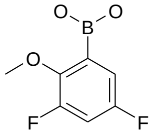
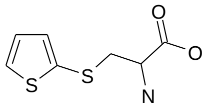
Reagent A	**Structure**	**Reagent B**	**Structure**
A08		B08	
A09		B09	
A10		B10	
A11		B11	
A12		B12	
A13		B13	
A14		B14	
A15		B15	
A16		B16	

Reagent A	Structure	Reagent B	Structure
A17		B17	
A18		B18	
A19		B19	
A20		B20	
A21		B21	
A22		B22	
A23		B23	

Reagent A	**Structure**	**Reagent B**	**Structure**
A24		B24	
A25		B25	
A26		B26	
A27		B27	
A28		B28	
A29		B29	
A30		B30	

Reagent A	Structure	Reagent B	Structure
A31		B31	
A32		B32	
A33		B33	
A34		B34	
A35		B35	
A36		B36	
A37		B37	

Reagent A	**Structure**	**Reagent B**	**Structure**
A38		B38	
A39		B39	
A40		B40	
A41		B41	
A42		B42	
A43		B43	
A44		B44	

Reagent A	**Structure**	**Reagent B**	**Structure**
A45		B45	
A46		B46	
A47		B47	
A48		B48	
A49		B49	
A50		B50