J Comput Aided Mol Des (2007) 21:327–340
DOI 10.1007/s10822-007-9121-3

Exploring fragment spaces under multiple physicochemical constraints

Juri Pärn · Jörg Degen · Matthias Rarey

Received: 7 February 2007 / Accepted: 8 May 2007 / Published online: 28 June 2007

Abstract We present a new algorithm for the enumeration of chemical fragment spaces under constraints. Fragment spaces consist of a set of molecular fragments and a set of rules that specifies how fragments can be combined. Although fragment spaces typically cover an infinite number of molecules, they can be enumerated in case that a physicochemical profile of the requested compounds is given. By using min–max ranges for a number of corresponding properties, our algorithm is able to enumerate all molecules which obey these properties. To speed up the calculation, the given ranges are used directly during the build-up process to guide the selection of fragments. Furthermore, a topology based fragment filter is used to skip most of the redundant fragment combinations. We applied the algorithm to 40 different target classes. For each of these, we generated tailored fragment spaces from sets of known inhibitors and additionally derived ranges for several physicochemical properties. We characterized the target-specific fragment spaces and were able to enumerate the complete chemical subspaces for most of the targets.

Keywords Chemical fragment spaces · Enumeration · De novo design · Library design · Ligand-based design · Computer chemistry

Abbreviations

ACE	Angiotensin-converting enzyme
AChE	Acetylcholinesterase
ADA	Adenosine deaminase
ALR2	Aldose reductase
AmpC	AmpC β-lactamase
AR	Androgen receptor
CDK2	Cyclin-dependent kinase 2
COMT	Catechol *O*-methyltransferase
COX-1	Cyclooxygenase-1
COX-2	Cyclooxygenase-2
DHFR	Dihydrofolate reductase
EGFr	Epidermal growth factor receptor
ER	Estrogen receptor
FGFr1	Fibroblast growth factor receptor kinase
FXa	Factor Xa
GART	Glycinamide ribonucleotide transformylase
GPB	Glycogen phosphorylase β
GR	Glucocorticoid receptor
HIVPR	HIV protease
HIVRT	HIV reverse transcriptase
HMGR	Hydroxymethylglutaryl-CoA reductase
HSP90	Human heat shock protein 90
InhA	Enoyl ACP reductase
MR	Mineralocorticoid receptor
NA	Neuraminidase
P38 MAP	P38 mitogen activated protein
PARP	Poly(ADP-ribose) polymerase
PDE5	Phosphodiesterase 5
PDGFrb	Platelet derived growth factor receptor kinase
PNP	Purine nucleoside phosphorylase
PPARg	Peroxisome proliferator activated receptor γ
PR	Progesterone receptor
RXRa	Retinoic X receptor α
SAHH	*S*-adenosyl-homocysteine hydrolase
SRC	Tyrosine kinase SRC
TK	Thymidine kinase
VEGFr2	Vascular endothelial growth factor receptor

J. Pärn · J. Degen · M. Rarey (✉)
Center for Bioinformatics, Bundesstrasse 43,
20146 Hamburg, Germany
e-mail: rarey@zbh.uni-hamburg.de

Introduction

In the beginning of every drug discovery process, the question arises which part of the chemical space should be considered. Due to its enormous size, the usage of the entire chemical universe is not feasible [1]. Oprea et al. showed that the whole chemical space does not have to be considered necessarily for applications in pharmaceutical research since drug-like compounds seem to cluster and to form a ''drugspace'' [2, 3]. But still, considering that only the subspace of carbon-based molecules with molecular mass less then 500 daltons has approximately 10^{60} molecules [4], it is obvious that complete enumeration is not an option.

A combinatorial way to model the chemical universe is the use of ''fragment spaces'' (for an recent review see [5]). In this work, we define fragment spaces as a tuple consisting of a set of fragments and a set of rules. Fragments basically are small molecular substructures having one or several special link atom(s) (attachment points). The set of rules defines which link atoms are compatible and can be used to merge two fragments into a new one. Due to the usage of fragments instead of atoms, fragment spaces do not cover the whole chemical space, but significant parts of it [6, 7].

A fragment space in this sense was generated for the first time by Lewell et al. applying a retrosynthetic combinatorial analysis procedure (RECAP) [8]. The RECAP rules were derived from common and easily accessible chemical reactions with the purpose that the resulting fragment combinations can easily be synthesized as well. Besides the aspect of chemical synthesis, the rules also take into account common motifs of biological active molecules. The conservation of chemical features is supposed to raise the probability that the resulting molecules also show biological activity (for an overview refer to [5]).

For the above reasons, fragment spaces are very attractive for typical tasks in pharmaceutical research like library design, de novo design or lead optimization. Nevertheless, their potential enormous size requires the usage of appropriate tools. For example, the World Drug Index (WDI) [9] results in roughly 30,000 fragments if shredded with the RECAP rules [7, 8]. The enumeration of this space for all molecules that consist of up to five fragments would result in 10^{18} molecules [6]. This amount is far too large to handle with classical tools, for instance in combination with high throughput screening. Therefore, tools for exploring fragment spaces or similar concepts need to enable the user to navigate within the space in order to find and investigate interesting subspaces of reasonable size.

Several tools have already been developed to address this challenge: Topas [7] performs a ligand-based search in fragment spaces by employing an evolutionary algorithm. The similarity to a template structure in terms of a 2D fingerprint in combination with a 2D topological pharmacophore is used as a fitness function. FTrees-FS [6] performs a fast similarity search by extending the feature tree approach [10] to fragment spaces. In structure-based drug design, a large variety of tools based on fragment combination exist [5, 11]. For example, FlexNovo [12] performs structure-based searching in fragment spaces as defined here by utilizing a *k*-greedy algorithm in combination with the FlexX [13] docking engine and scoring function.

Fragment spaces are similar but conceptionally different from virtual combinatorial libraries. While in the latter, all molecules are constructed from fragments employing the same topology (most commonly a core and two to three r-groups), molecules from fragment spaces are neither limited in the number of fragments nor in the topology in which they are connected. Therefore, fragment spaces can be seen as a generalization of combinatorial libraries. Several tools exist to manipulate and search in virtual combinatorial libraries. For an overview we refer to [14, 15].

To our knowledge, no tool exists for the complete enumeration of molecules from fragment spaces so far. Our new approach allows the enumeration of all molecules in a certain region of a given input space. In combination with min–max values for certain physicochemical properties, the algorithm generates all molecules fulfilling the requested constraints. Table 1 shows the currently supported physicochemical properties that can serve as boundaries for the enumeration. A key feature of our tool is that it considers the requested properties directly during the enumeration to guide the selection of fragments. By means of special data structures, only those fragments are fetched from the fragment space that are able to satisfy the requested constraints in combination with the already built-up fragment.

The enumeration algorithm can be used for different applications during the drug discovery process, for example, for the generation of focused compound collections for

Table 1 Physicochemical properties that can be used for the enumeration

Strictly additive, numerical properties	
Atoms	Molecular weight
Non-hydrogen atoms	Non-hydrogen bonds
Rings (as sum of sssr)	Rotatable bonds
H-bond acceptors	H-bond donors
Non-strictly additive, numerical properties	
Calculated log *P* value	Tetrahedral stereo centers
Non-additive, non-numerical properties	
Inclusion smarts	Exclusion smarts

virtual screening or for the diversity or scaffold analysis of chemical spaces. Another approach could be, for example, the generation of targeted libraries as we did in the context of this paper. Therefore, we shredded annotated collections of inhibitors and enumerated all molecules from the resulting fragment spaces lying within similar property ranges. We characterized the individual target-specific chemical spaces and the corresponding results.

Methods

Fragment spaces

A chemical fragment space is a tuple consisting of a set of molecular fragments and a set of rules. The fragments have one or more open valences, called link atoms, each having a certain type (see also Table 2). Besides the link type, these atoms have no further information attached and can be seen as dummy atoms. Furthermore, the fragments in the initial set are defined as ''atomic'' fragments in order to distinguish them from fragments that are generated during the enumeration. The set of rules defines which link types are compatible and can be used to combine fragments to form new (non-atomic) fragments. Besides the compatibility, the rules also define terminal fragments for each link type. The terminal fragments have one link atom and can only be used to resolve open (unused) links types in fragments. The atomic fragments and the rules span the fragment space. It should be mentioned that the size of fragments spaces is not bounded, i.e. it can be infinite in general.

Constraint molecule enumeration algorithm

The input for the main algorithm is a fragment space and constraints in terms of min–max values for physicochemical properties. The algorithm then enumerates all molecules in the given space which obey the requested constraints.

The main data structure for the enumeration is a tree [16] in which every node stores an atomic fragment. Besides the fragment, every node also stores information about via which link atoms its ancestor and its children are connected (see also Fig. 1). Thus, the whole tree and the corresponding molecule can be reconstructed given an arbitrary node of the tree.

Even if the enumeration of trees has been deeply investigated, see for example [16], the results are not directly applicable to our enumeration task since valid trees are dictated by the fragment space used.

For the algorithm, three problems had to be solved, namely how to access fragments with certain properties in an efficient way, how to neglect redundant fragment-trees as soon as possible and how to confirm the uniqueness of a generated molecule effectively. We start with the main algorithm and illustrate solutions for the three sub-problems afterwards. In the following, we assume that the requested properties for the molecules are numerical and strictly additive like, for example, molecular weight and the number of rings. The handling of non-numerical and non-strictly additive properties will be discussed at the end of this section.

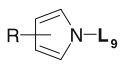
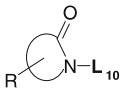
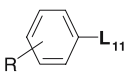
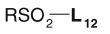
The main steps of the algorithm are illustrated in Fig. 2. The input for every recursion step is a fragment-tree. The given fragment-tree is tested with the redundancy filter. If the fragment-tree passes this test, the algorithm forks into a molecule generating part and a fragment-tree extension part. The molecule generating part checks in a first step if all requested properties are satisfied. After this test the canonical SMILES [17] notation for the corresponding molecule is generated. The SMILES is used to verify the uniqueness of the molecule by means of string comparisons with all other molecules generated so far. If the molecule passes this test, it is written to a file and will be considered in subsequent uniqueness tests.

In the fragment-tree extension part, ranges for fragments are calculated to satisfy the requested min–max values in combination with the fragment-tree. After this, all fragments lying in the calculated ranges are fetched and, in the last step, every fetched fragment is combined in every possible combination with the fragment-tree. The newly generated (extended) fragment-tree is then used as input for the algorithm in a recursive fashion.

To avoid unnecessary generations of identical molecules, a fragment redundancy test on topology level is performed for every potential molecule. This step is necessary since the same molecule can be built up from the same fragments in different ways. Consider, for example, the three fragments *A, B* and *C*. With these three fragments, the molecule *A*-*B*-*C* can be built up in the four following orders: (*A*, *B*, *C*), (*C*, *B*, *A*), (*B*, *C*, *A*), (*B*, *A*, *C*). The main idea for the topology test is to give every fragment an identification number (ID). With this information, a powerful redundancy test can be performed. The first part of the test is to check whether any of the fragment-tree nodes has a smaller ID than the root. In this case, the tree is redundant since the node with the smaller ID was or will be the root of an equivalent tree. The second part of the test is to check for every node, if the ID's of the children are attached in ascending order (see Fig. 3). Again, if this is not the case, the tree is redundant and was already generated or will be generated. For comparison, a valid fragment tree is shown in Fig. 4.

The redundancy test avoids the regeneration of identical molecules to a large degree. However, some situations still remain in which identical molecules are constructed. For

Table 2 Fragment space obtained after applying the RECAP rules

Fragment	Terminal group	Link compatibility matrix											
		L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11}	L_{12}
	L_1-O^-	□	■	■	□	□	□	□	□	□	□	□	□
	L_2-H	■	□	□	□	□	■	□	□	□	□	□	■
	L_3-H	■	□	□	■	□	□	□	□	□	□	□	□
	L_4-H	□	□	■	□	■	□	□	■	■	■	□	□
	L_5-H	□	□	□	■	□	□	□	□	□	□	□	□
	L_6-CH_3	□	■	□	□	□	□	□	□	□	□	□	□
	L_7=CH_2	□	□	□	□	□	□	■	□	□	□	□	□
	L_8-CH_3	□	□	□	■	□	□	□	□	□	□	□	□
	L_9-H	□	□	□	■	□	□	□	□	□	□	□	□
	L_{10}-H	□	□	□	■	□	□	□	□	□	□	□	□
	L_{11}-H	□	□	□	□	□	□	□	□	□	□	■	□
	L_{12}-CH_3	□	■	□	□	□	□	□	□	□	□	□	□

On the left hand side, the fragment prototypes and their respective terminal groups are shown. Each fragment has a (dummy) link atom that represents the corresponding chemical environment ($L_1 - L_{12}$). The diversity of the fragments is in the R-groups that can contain further link atoms. A gray box in the central matrix indicates the (symmetrical) compatibility of two link atoms. The link atoms themselves will be removed upon fragment connection. Terminal groups are used for the substitution of unused link atoms

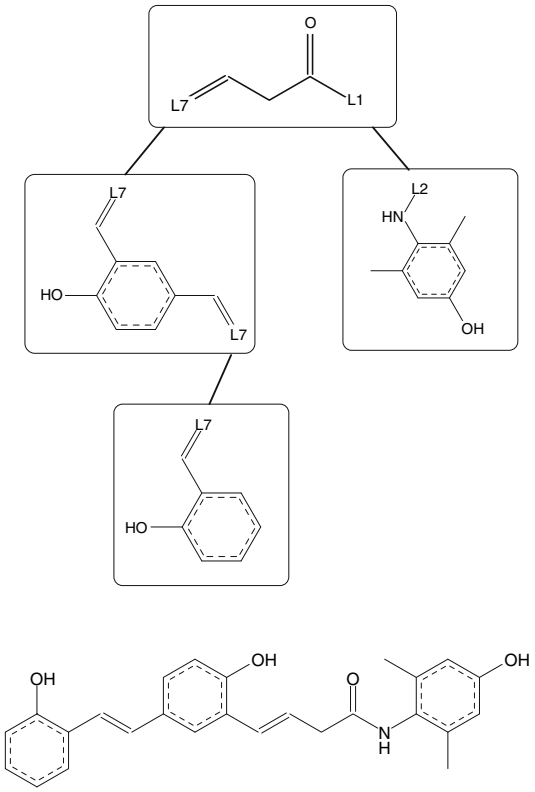


Fig. 1 Fragment tree and the corresponding molecule. Every node of the fragment-tree stores an atomic fragment and information about via which link the ancestor and the children are connected. If the connection information is not considered explicitly the tree above would be inconclusive since the connections of the root's left child can be swapped

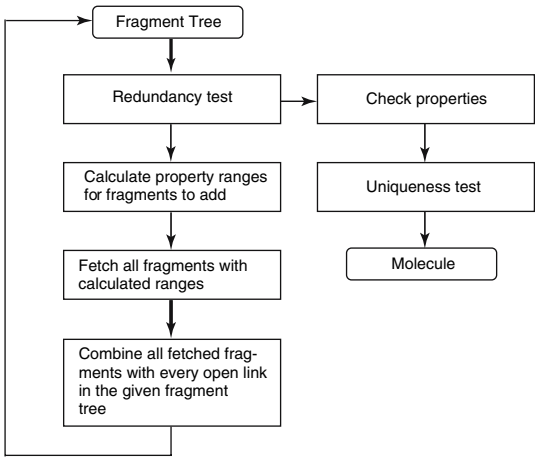


Fig. 2 Main steps in the constraint molecule enumeration algorithm

example, the redundancy test defines trees as not redundant if all nodes have the same ID, i.e. are identical. In this case, all valid trees with the nodes are enumerated and pass the filter although only few are really unique. To have an efficient way to confirm the uniqueness of newly generated

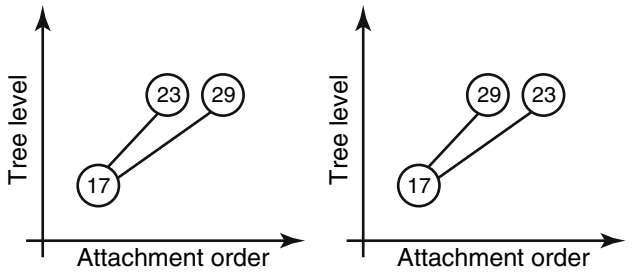


Fig. 3 Fragment redundancy test. Two fragment trees representing an equivalent fragment/molecule. The fragment tree on the left hand side would be accepted whereas the tree on the right hand side would be rejected because the ID of the first child is greater then the ID of the second child

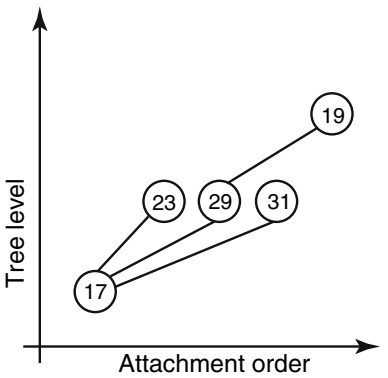


Fig. 4 Valid fragment tree. All ID's of all fragment-tree nodes are greater than the ID of the root. In every level the ID's are in ascending order with regard to the attachment order

molecules, the corresponding canonical SMILES are stored in a self-balancing binary search tree [18]. This type of tree as well as the *k*D-Trees introduced below are not to be confused with the enumeration tree. They all share the same data structure idea but are implemented for different purposes. In self-balancing binary search trees it is guaranteed that for every node the difference in height between the left and the right child tree is at most one, resulting in logarithmic access times.

For an efficient access to fragments with certain (numerical) properties, *k*D-Trees [19] are utilized. Every property which is constrained by the query will be calculated for every fragment and reflects one dimension of the resulting *k*-dimensional space. *k*D-Trees decompose the input space by alternately dividing one of the *k* dimensions into two subspaces by means of a hyperplane. This is performed recursively until only a few or one data point lies within every generated subspace. This *k*D-Tree is then used to fetch only atomic fragments which will not violate the requested max values after attaching them to the current fragment-tree. Therefore, the current fragment-tree needs to be analyzed: The max boundary for each of the *k*

dimension is calculated by summing up the corresponding values for each fragment in a fragment-tree and then subtracting the requested max value. The min boundary is the minimum value of all atomic fragments in the space regarding the *k*th dimension.

If a mix of strictly additive and non-strictly additive properties is requested, the procedure stays the same, except that, after the molecule passes the topology test, the non-additive properties have to be calculated and checked if they lie in the requested ranges. If only non-strictly additive properties are given, the complete space has to be enumerated and for every generated molecule the properties have to calculated and checked. In this case, only the redundancy check on topology level improves the speed of the enumeration.

Note that the number of molecules which can be enumerated depends on the structure of the fragment space. In most cases, the number grows exponentially with the number of atomic fragments used per molecule [6]. Therefore, an exhaustive enumeration is not feasible in most cases. However, as will be shown in the result section below, the enumeration can be performed in customized spaces with reasonable constraint definitions.

Results and discussion

For the validation of our new method we used the following scenario: First, we provide the results of a general analysis of the runtime behavior of the tool. We then describe the calculations for different targets in detail and analyze the results. Finally, we summarize the characteristics of the calculations and provide some case examples.

Runtime

As the main part of the algorithm is a simple recursive algorithm with no time consuming parts, different function calls like the *k*D-Tree query and mainly the build-up of the molecules are the dominating parts in terms of runtime.

The number of enumerated compounds and, consequently, the runtime heavily depends on the fragment space. Fragment spaces containing fragments with many link types enormously amplify the enumeration steps as can be seen in Table 3.

The `wdi_sub_47` space contains 35 fragments with 1 link atom and 12 fragments with 2 link atoms which are drawn randomly from the WDI-Space [6]. For the space `wdi_sub_48`, we extended the `wdi_sub_47` space with a fragment having 6 link atoms shown in Fig. 5. By adding this one fragment, the number of unique compounds containing up to 5 fragments grows by a factor of 1.3 from 73147 to 95713. However, the number of enumerated molecules without using the filters would grow from 1,048,401 to 17,399,458, a factor of nearly 17. This shows that only adding one fragment to a space can result in an enormous boost of enumeration steps. This also illustrates the need for a redundancy filter, since the number of enumerated molecules grows much faster than the number of unique molecules.

Table 3 Impact of redundancy filter on topology level

Space	Enum. Cpds	Unique Cpds	Time (s)
`wdi_sub_47`	1,048,401	73,147	9,412
`wdi_sub_47`	96,549	73,147	686
`wdi_sub_48`	17,399,458	95,713	114,697
`wdi_sub_48`	1,066,648	95,713	7,311

The impact is shown on two spaces `wdi_sub_47` and `wdi_sub_48`. In the first row of every space a complete enumeration with up to five fragments without topology filter and in the second row with topology filter is performed

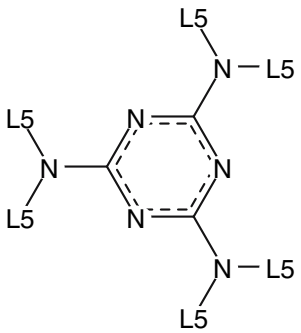


Fig. 5 Additional fragment in `wdi_sub_48`

The impact of different fragment spaces and redundancy filter is shown in Table 3.

In the `wdi_sub_47` space, the redundancy filter can skip nearly 91% of the molecules to be enumerated. The time to enumerate the whole space decrease from 9412 to 686 s, i.e. a factor of nearly 14.

In the `wdi_sub_48` space, the filter avoid nearly 94% of the enumerated molecules. The time needed to enumerate the whole spaces decreases by a factor of nearly 16. This results shows impressively the benefit of the redundancy filter.

Application scenarios

For our application purposes we selected sets of inhibitors from the ZINC database annotated with respect to different targets (in the following we also used the same short forms as in the original publication) [20]. Based on these compound sets, we generated tailored fragment spaces by shredding and enumerated the complete chemical space

underlying different target classes (see below). From the enumeration experiments, we basically obtain all the molecules that can be generated and, thereby, the size of the corresponding space. But this information alone is clearly not sufficient to characterize the nature of a target-specific chemical space. Therefore, we additionally carried out similarity calculations using the Feature Tree program [10] in order to assess also the similarity of the generated molecules to the inhibitors we used as the input structures.

Generation of the fragment spaces

In a first step, we applied a retrosynthetic fragmentation procedure to each set of inhibitors for each target class. The approach we used is a modified version of the RECAP procedure [8] that has already been used before for the generation of fragments [6, 7]. The inhibitor sets differ in size and, therefore, the amount of fragments obtained after the shredding varies also significantly between the different classes. We did not perform any further filtering steps in order to retain all chemical features of the original inhibitors and not to introduce any bias in the fragment set.

It has to be emphasized at this point already, that some of the inhibitors were not cleaved at all due to the specific shredding rules used and also due to the large variety in size of the input structures. As a consequence, small structures or molecules with less decoration are not fragmented due to lacking cleavage bonds. Additionally, we avoided the generation of very small terminal fragments, for example, short alkyl chains, single hydrogen or halogen atoms, nitro or hydroxo groups etc. As a consequence, for most of the spaces generated, not all inhibitors are used for shredding and, therefore, these may not be reproduced during the enumeration. We will come back to this point during the discussion of the individual results.

In the second step, we defined a fragment space for every fragment set we obtained for each target class. This particular type of fragment space has already been used in other programs and, therefore, the basic properties have already been described [6]. On average, 75% of the fragments contain one link, 22% have two links, 2.5% have three links and 0.5% have four links. Fragments with more than four links were not generated. In our case, the same set of rules is used for cleavage and recombination. The results of the shredding procedure and the properties of the corresponding fragment spaces are listed in Table 4.

Further analyses of the spaces reveal that they have quite different properties. Typically, the larger spaces are connected and infinite: *Connected* means that every pair of fragments can theoretically occur in the same molecule. The theoretical size of the space is *infinite* in the case where an infinite number of molecules can be generated, according to the compatibility of the links. Additionally, the resulting molecules can be constructed by using multi-link fragments and can also contain multiple copies of the same fragment. This information is also listed in Table 4. Note that the size of a space is typically only of theoretical relevance. Since the molecular weight of molecules is typically limited, the fragment space becomes also finite in

Table 4 Shredding results and fragment space properties

Target class	No. of inhibitors	No. of fragments	Fragment space size	No. of components
ACE	49	45	∞	2
AChE	107	100	∞	2
ADA	39	25	Finite	12
ALR2	26	17	∞	4
AmpC	21	26	Finite	12
AR	79	15	∞	4
CDK2	72	62	∞	4
COMT	11	7	Finite	12
COX-1	25	25	∞	5
COX-2	426	271	∞	1
DHFR	410	147	∞	3
EGFr	475	125	∞	2
$ER_{agonists}$	67	29	∞	4
$ER_{antagonists}$	39	51	∞	3
FGFr1	120	106	∞	2
FXa	146	189	∞	1
GART	40	23	∞	3
GPB	52	24	∞	2
GR	78	57	∞	5
HIVPR	62	70	∞	2
HIVRT	43	40	∞	2
HMGA	35	31	∞	1
HSP90	37	26	∞	4
InhA	86	92	∞	1
MR	15	3	Finite	12
NA	49	26	∞	5
P38	454	194	∞	2
PARP	35	20	∞	5
PDE5	88	93	∞	2
PDGFrb	170	141	∞	2
PNP	50	27	Finite	5
PPARg	85	90	∞	4
PR	27	26	Finite	9
RXRa	20	4	∞	5
SAHH	33	27	∞	5
SRC	159	122	∞	2
Thrombin	72	87	∞	2
TK	22	23	∞	5
Trypsin	49	58	∞	2
VEGFr2	88	118	∞	1
All classes	3,961	1,942	∞	1

size. Therefore, Table 6 contains a more realistic estimate of each fragment space. This issue will be addressed in more detail during the description of the enumeration procedure and in the results section.

For the smaller fragment spaces we obtained, the above is not necessarily the case. They may decompose into individual components that can be finite or infinite in size. In any of the two cases, the resulting molecules cannot contain every arbitrary pair of fragments any more. This is per se not a limitation since the amount of theoretically possible molecules may still be very large. But if all of the components are also finite in size, or the number of theoretically accessible molecules is too low (see below), this is in fact not a *real* space any more and, as a consequence, we did not consider them further in our analyses. Therefore, the corresponding classes are omitted in the following tables.

Enumeration procedure

The input for the enumeration consists of the fragment spaces we generated (see above) together with the definition of additional constraints, which are min–max values for a number of physicochemical properties. For our purposes, we decided to use molecular weight, number of rotatable bonds, number of rings, the number of hydrogen bond donors and acceptors and the calculated log *P* values. We derived the corresponding ranges from the same set of inhibitors that we also used for generating the fragments by taking the minimum and maximum values of all selected properties for the whole set.

In this context, the number of hydrogen bond donors and acceptors is calculated as described by Lipinski et al. [21] The log *P* values are calculated using the atom substructure descriptors described by Wildman and Crippen [22]. The ''number of rings'' is determined as the number of ring closures (no. of bonds-no. of atoms +1).

In accordance with the fragment generation procedure before, we did not modify the derived values but rather directly used the ranges that we obtained from the individual inhibitor sets. This led to relatively broad margins in some cases. For a real application scenario, where there may be a lot more information available, this should of course be used and also the fragments should probably be selected and filtered by more sophisticated means. However, we deliberately decided not to use any further (expert) knowledge about the individual targets in order to focus on the capabilities of the enumeration algorithm. Table 5 contains a list of the corresponding property ranges that we used for the calculations.

Most of the fragment spaces we generated are, in principle, infinite in size. But for our practical examples, this is of little relevance since the size of the resulting molecules is limited. Additionally, we avoided the generation of very small terminal fragments during the fragmentation (see above). Hence we can derive another constraint for the enumeration from the fragment properties. Therefore, we limited the maximum number of fragments allowed per molecule to five, which is also in accordance with other approaches [6]. By doing so, the size of the corresponding fragment spaces becomes finite and an upper bound for the number of theoretically accessible molecules can be derived according to the compatibility of the different link types. The corresponding values are listed in Table 6. Note that the enumeration algorithm presented is not limited to a maximum number of fragments.

The output of the enumeration is a list of molecules that are stored in a canonical SMILES representation [17]. For further analyses, we converted these structures to Tripos mol2 format [23] using CORINA [24, 25]. Since we were not interested in performing analyses that would depend on a good conformation at this point, we also kept rarely occurring bad 3D models (according to CORINA). The molecules generated result from a single run of the program with no further parameter tuning etc. Therefore, the results and also the statistics of the calculations can probably be improved in most cases by using more knowledge about the individual targets.

Results of the enumeration

The results of the calculations we performed are quite different for the individual targets. First of all, when looking at Table 6, it becomes apparent that not all the calculations could be completed successfully, as indicated by ''DNF'' (did not finish) in the third column. This is a consequence of the limited memory on the Linux nodes we used (4 GB) and also due to the size of the fragment spaces. In all but one case, the theoretical number of molecules exceeds 10^7. This value equals the number of theoretically accessible molecules that can be generated when connecting up to five fragments according to the recombination rules.

The properties of a space are an indication but not sufficient to judge whether or not a space can be enumerated under constraints. For example, the spaces of HMGA and InhA could both be enumerated successfully, although they are connected and infinite. On the other hand, the calculations for HIVPR, P38 and PPARg could not be finished although these decompose into two or more components (but still are infinite in size). Furthermore, the DHFR space containing 147 fragments could be enumerated whereas this was not possible for the ER antagonist space containing 51 fragments only.

Table 5 Physicochemical property ranges for the inhibitor sets

Target class	Molecular weight	Rot. bonds	Rings	h-bond acc.	h-bond don.	c log *P*
ACE	176.2–516.3	3–12	0–3	4–9	0–4	–0.2–4.2
AChE	166.2–482.6	0–21	0–5	2–8	0–6	1.1–6.0
ALR2	225.2–449.2	0–8	1–4	2–7	0–4	–1.2–6.2
AR	256.4–556.5	1–9	1–5	1–8	0–3	2.0–10.0
CDK2	235.2–483.6	0–14	2–8	4–10	1–7	–1.1–6.2
COX-1	137.1–371.4	1–11	1–3	2–6	0–4	1.4–5.6
COX-2	243.3–571.1	0–12	2–6	0–8	0–7	1.7–9.2
DHFR	206.3–521.4	0–11	2–5	4–13	4–8	–0.4–5.7
EGFr	221.3–597.5	0–12	2–5	2–10	0–8	–2.6–7.3
$ER_{agonists}$	212.2–356.4	1–11	2–5	1– 6	0–4	2.0–7.2
$ER_{antagonists}$	347.5–508.7	5–12	3–6	2–5	1–3	5.1–8.8
FGFr1	281.3–597.5	0–14	3–5	4–10	1–5	0.7–7.5
FXa	293.4–574.4	1–13	2–7	4–12	1–12	–1.9–6.7
GART	429.4–479.5	7–11	2–3	11–14	4–7	–2.7–3.6
GPB	180.2–424.9	4–11	1–4	4–11	0–8	–6.2–4.3
GR	271.4–460.6	3–11	2–5	2–6	0–3	1.1–8.8
HIVPR	362.5–631.8	5–16	2–7	3–10	0–7	2.0–8.2
HIVRT	244.3–584.3	1–13	1–5	2–9	0–4	1.3–8.6
HMGA	256.3–557.6	3–20	2–4	3–9	0–3	2.2–6.7
HSP90	296.3–433.9	3–13	3–4	5–8	2–7	1.7–5.0
InhA	197.2–646.3	0–12	1–7	2–9	0–4	0.8–6.9
NA	193.2–408.5	1–14	1–3	5–11	1–9	–7.2–2.4
P38	239.2–509.6	0–10	2–5	2–8	0–5	0.3–7.8
PARP	136.2–366.5	0–4	1–5	3–5	1–4	0.3–6.2
PDE5	271.3–620.7	1–14	2–7	5–12	0–4	1.4–7.4
PDGFrb	210.2–597.5	0–12	3–5	1–10	0–5	0.2–7.5
PPARg	264.3–640.7	2–15	2–6	3–10	0–3	1.0–10.4
SAHH	233.2–321.4	3–7	2–3	6–9	4–6	–3.8 to –0.9
SRC	226.2–597.5	0–13	3–5	4–10	0–5	1.6–7.5
Thrombin	250.3–647.8	2–17	2–6	3–12	1–9	–0.4–5.0
TK	184.2–405.2	3–7	1–3	5–9	2–5	–2.2–2.0
Trypsin	124.2–620.9	3–17	1–6	2–11	5–15	–0.3–4.7
VEGFr2	238.3–692.7	0–14	2–7	2–14	0–6	1.6–8.0
All classes	122.1–692.7	0–21	0–8	0–14	0–15	–7.2–10.4

In general, the amounts of molecules that are created during the enumeration intuitionally correlate with the size of the spaces to some extent. But interestingly, the fraction of molecules that are generated and actually fulfill all constraints compared to the theoretical number of possible compounds varies quite significantly. These numbers range from values in the low sub-percent region in the case of ER agonists, ACE and NA up to two-digit percent values for the AR, GPB, InhA and thrombin examples.

This variation is also true for the runtimes of the calculations. These correlate with the amount of molecules generated and range from seconds, in case of small fragment spaces, to several days, for quite large spaces. In this context, the ''performance'' is probably a more meaningful measure, i.e., the number of unique molecules within the requested property ranges generated per second. Again, this intuitionally correlates with the total amount of molecules generated, but the values range from 2 to more than 50 molecules per second. This effect is mostly due to the fact that the time needed by the algorithm to actually generate the compounds in the end is a dominating factor. Additionally, the size of the binary search tree for the SMILES depends on the number of molecules generated.

When looking at the statistics in Table 6, it becomes clear that the results can roughly be separated into three categories. The first category consists of the classes where

Table 6 Results and statistical information of the enumeration calculations

Target class	No. of enumerated molecules	Theor. no. of molecules	Fraction (%)	Runtime (s)	Performance (mols/s)
ALR2	21	1.06e+04	0.2	1	21
AR	37	9.40e+01	39.4	1	37
NA	56	9.71e+03	0.6	2	28
PARP	102	1.06e+04	1.0	3	34
SAHH	128	4.54e+03	2.8	2	64
GART	188	3.70e+03	5.1	4	47
COX-1	298	9.60e+03	3.1	9	33
TK	416	2.93e+04	1.4	18	23
GR	752	1.74e+04	4.3	23	33
GPB	940	5.96e+03	15.8	27	35
$ER_{agonists}$	1,387	1.61e+06	0.1	31	45
HIVRT	2,666	4.68e+04	5.7	238	11
ACE	5,604	2.38e+06	0.2	159	35
HSP90	5,839	2.28e+06	0.3	560	10
CDK2	7,610	1.60e+05	4.8	259	29
Trypsin	23,716	3.07e+05	7.7	4,816	5
HMGA	27,264	1.14e+06	2.4	5,102	5
DHFR	36,691	9.82e+05	3.7	3,017	12
FGFr1	84,608	2.66e+06	3.2	25,079	3
SRC	105,529	3.47e+06	3.0	35,553	3
AChE	136,025	8.98e+06	1.5	26,551	5
PDGFrb	205,119	7.46e+06	2.7	33,175	6
EGFr	396,085	1.29e+07	3.1	30,505	13
InhA	544,705	3.45e+06	15.8	227,866	2
Thrombin	779,213	5.23e+06	14.9	212,086	4
COX-2	DNF	8.99e+11	–	–	–
$ER_{antagonists}$	DNF	3.44e+07	–	–	–
FXa	DNF	1.61e+09	–	–	–
HIVPR	DNF	1.07e+09	–	–	–
P38	DNF	5.78e+10	–	–	–
PDE5	DNF	9.15e+07	–	–	–
PPARg	DNF	1.86e+08	–	–	–
VEGFr2	DNF	9.30e+07	–	–	–
All classes	DNF	5.38e+14	–	–	–

The table is sorted according to the number of molecules generated by the enumeration. The third column contains the upper bound for the number of theoretically accessible molecules that can be generated when connecting up to five fragments. Additionally, some key number of the calculations are given. For details see text

the enumeration led to several dozens to a few thousand molecules and could be carried out within seconds to minutes, for example for ACE, ALR2, CDK2, ER agonists, HIVRT and NA. The second category is made up by the classes where the enumeration yielded up to a few hundred thousand molecules and took up to a few hours, as observed, for example, for DHFR, HMGA and trypsin. Into the last category fall the classes where we either obtained several hundred thousands of molecules or none at all and the calculations took up to several days or did not finish, respectively.

The potency of the approach presented in this paper becomes most apparent for the first two categories, which comprise a significant part of the validation experiments we performed. In each case it was possible to enumerate the complete underlying chemical subspace under physicochemical constraints within a couple of minutes to a few hours of computation time. Additionally, only a tiny fraction of the theoretically accessible number of molecules has actually to be considered, resulting in several dozens to a few thousand molecules for the corresponding classes. With such a small compound set, more

sophisticated follow-up calculations and analyses can easily be carried out.

It has to be emphasized that all experiments were carried out in a straight-forward fashion, i.e., without using additional knowledge for the individual targets and with no parameter tuning, for example, by using a lower limit for the maximum number of fragments allowed per molecule. Therefore, there is clearly room for improvement, especially for the calculations that took very long, yielded large amounts of molecules or did not finish at all. With a tailored collection of fragments and narrower property ranges it is probably possible to enumerate even those spaces where the calculations could not be completed with our rather simple approach.

Nevertheless, the results generated already demonstrate that it is, in principle, possible to gain detailed insight of the underlying chemical space for several target classes by using a straight-forward approach. Since we have dealt only with the statistics so far and did not consider the actual results of the calculation, we will have a closer look at the molecules generated in the following.

Feature tree similarity analyses

In order to assess the quality of the generated molecules in a topological sense, we carried out similarity calculations using the Feature Tree descriptor [10]. The feature tree similarity measure considers the relative arrangements of functional groups; common substructures play only a minor role. This makes this similarity measure especially suitable for the analysis of different molecules originating from the same fragment space.

Here, we focused on finding the closest inhibitor analogues for the results. For each molecule in each solution set, we determined the highest feature tree similarity value to any of the inhibitors of the corresponding target class. From this analysis, we obtained distributions of similarity values for the complete solution sets.

It has to be emphasized at this point, that for the opposite analysis, i.e. finding the closest analogue in the solution set for each inhibitor, generally much higher average similarity values are observed than the ones presented below (about 0.10 units in terms of feature tree similarity). As mentioned before, typically not all the inhibitors were reproduced during the enumeration, which is due to the fact that some input structures were not cleaved and, therefore, some structural motifs might be missing in the fragment sets generated. This is a consequence of the shredding rules we used (see above). Consequently, the average similarity value for this analysis is not equal to one. However, this information is deliberately not presented in this paper since the results are not as meaningful as the ones shown in the following paragraphs.

For comparison, we also generated a reference (random) feature tree similarity distribution. We used the dataset generated by Stahl and Rarey that was originally compiled for the analysis of scoring functions for virtual screening [26]. This dataset consists of a collection of 7528 random drug-like molecules from the WDI and a complementary set of 452 inhibitors for different targets. All structures belonging to the corresponding inhibitor classes were excluded from random collection. We could only use 398 molecules from the inhibitor set due to proprietary issues.

Figure 6 contains the results of the calculations. The histograms are represented as smoothed line graphs. We also plotted the reference distribution (gray) for every image to facilitate inspection.

Following the analysis of the enumeration results, the feature tree similarity distributions obtained can also be put into three different categories. The first category consists of the solution sets where the amount of molecules generated ranges from several dozens to a few hundreds and, therefore, roughly equals the number of corresponding inhibitors that were used for the generation of the fragment spaces beforehand. Typically, these molecules are topologically highly similar to the actives, for example in the case of AR, GART, NA and, less pronounced, PARP. But there are also exceptions: For example ALR2, where the average similarity value is 0.80, as well as SAHH and COX1 that both show a broader distribution.

The second category is made up by the classes where the enumeration yielded a couple of hundred up to several thousand molecules. In general, the distributions obtained are still shifted to higher similarity values compared to the reference distribution. But, typically, the effect is less pronounced the larger the amount of generated molecules is, although there is clearly no linear dependence. Into the third category fall the classes where up to several hundred thousands of molecules were obtained. The corresponding distributions are more or less identical to the reference distribution we generated.

The fact that the feature tree similarity distributions resemble the reference distribution all the more, the higher the amount of generated molecules is, does not mean that the outcome for a particular target is of lower quality than for other targets. It rather reflects the properties of the underlying space and indicates that some solution sets are more diverse than others. According to the analysis of the properties of the individual fragment spaces (see above), the effects we observed for the similarity analyses are a result of the combination of all properties. And although we used an identical retrosynthetic fragmentation setup for all inhibitor sets, the differences we observe must also depend on the fragments that we used for generating the molecules. Consequently, some of the target classes contain highly specific structural motifs that are retained in the

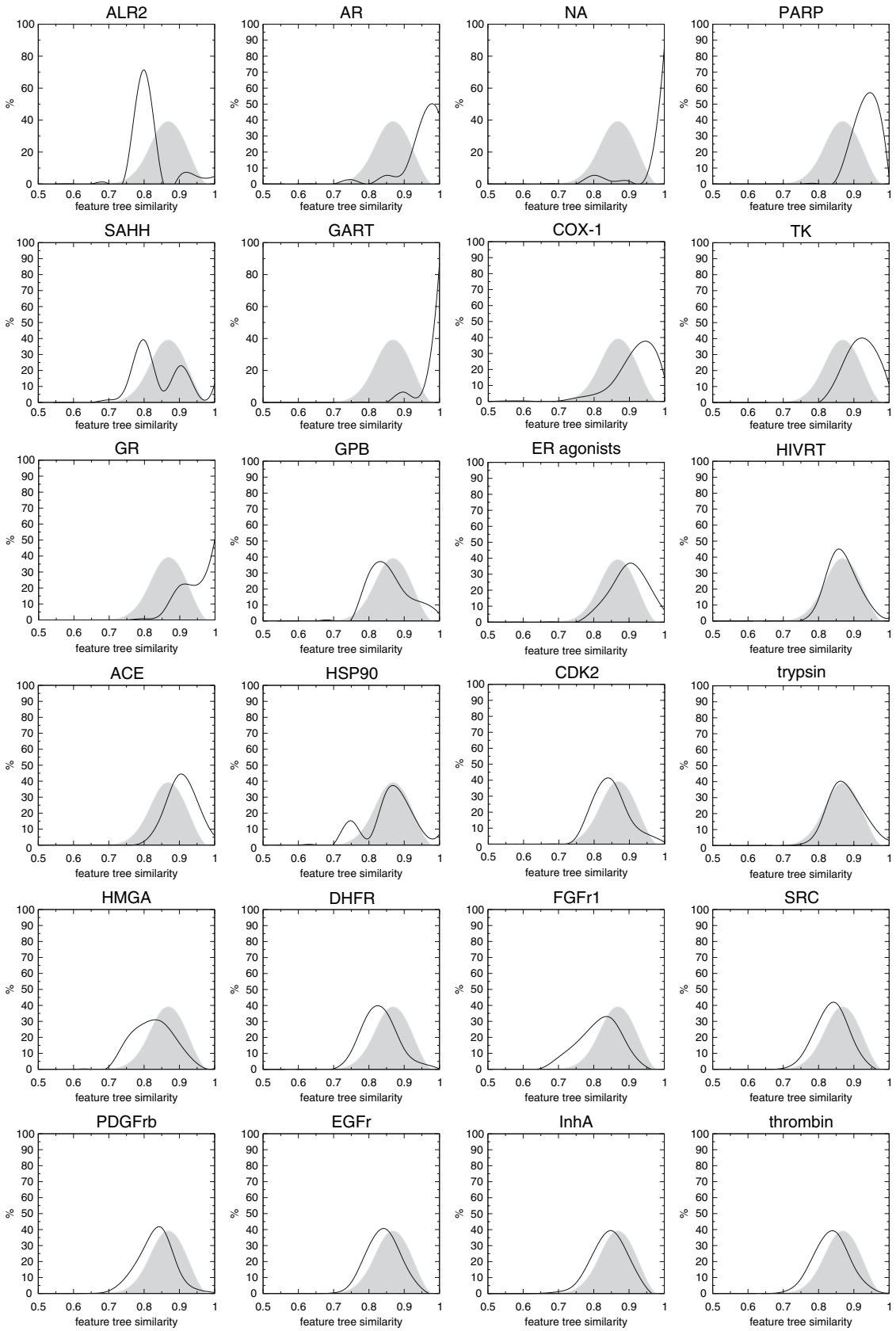


Fig. 6 Feature tree similarity distributions for the different target classes. Histograms are sorted according to Table 6, normalized and represented as smoothed line graphs. The reference distribution is included in each individual image and shown in gray

fragments whereas fragment sets for other collections of inhibitors are probably more populated with general chemical features.

In general, the enumeration generated sets of molecules that all have reasonable to high similarity to the original inhibitors that we used as the input. This is something we expected since a complete enumeration should intuitionally generate a set of highly similar molecules. For example, Fig.7 contains the images of different pairs of inhibitors and molecules that we selected from the results of the calculations described above. The individual figures were created using the method from Fricker et al. [27] In all cases, the structural similarity of the compound pairs becomes apparent and reasonable functional variations can be observed.

Summary and conclusions

We presented a new algorithm for the enumeration of fragment spaces under multiple physicochemical constraints, which are directly used during the calculation to guide the selection of fragments and molecules. Furthermore, powerful filters avoid the generation of redundant fragments and enormously improve the efficiency of the tool. For an extensive validation, we applied the program to 40 different target classes. We used annotated sets of inhibitors and subjected them to retrosynthetic fragmentation. We defined corresponding fragment spaces employing the same set of rules as used for fragmentation. The fragment spaces together with physicochemical property ranges derived from the inhibitor sets formed the input of the calculations.

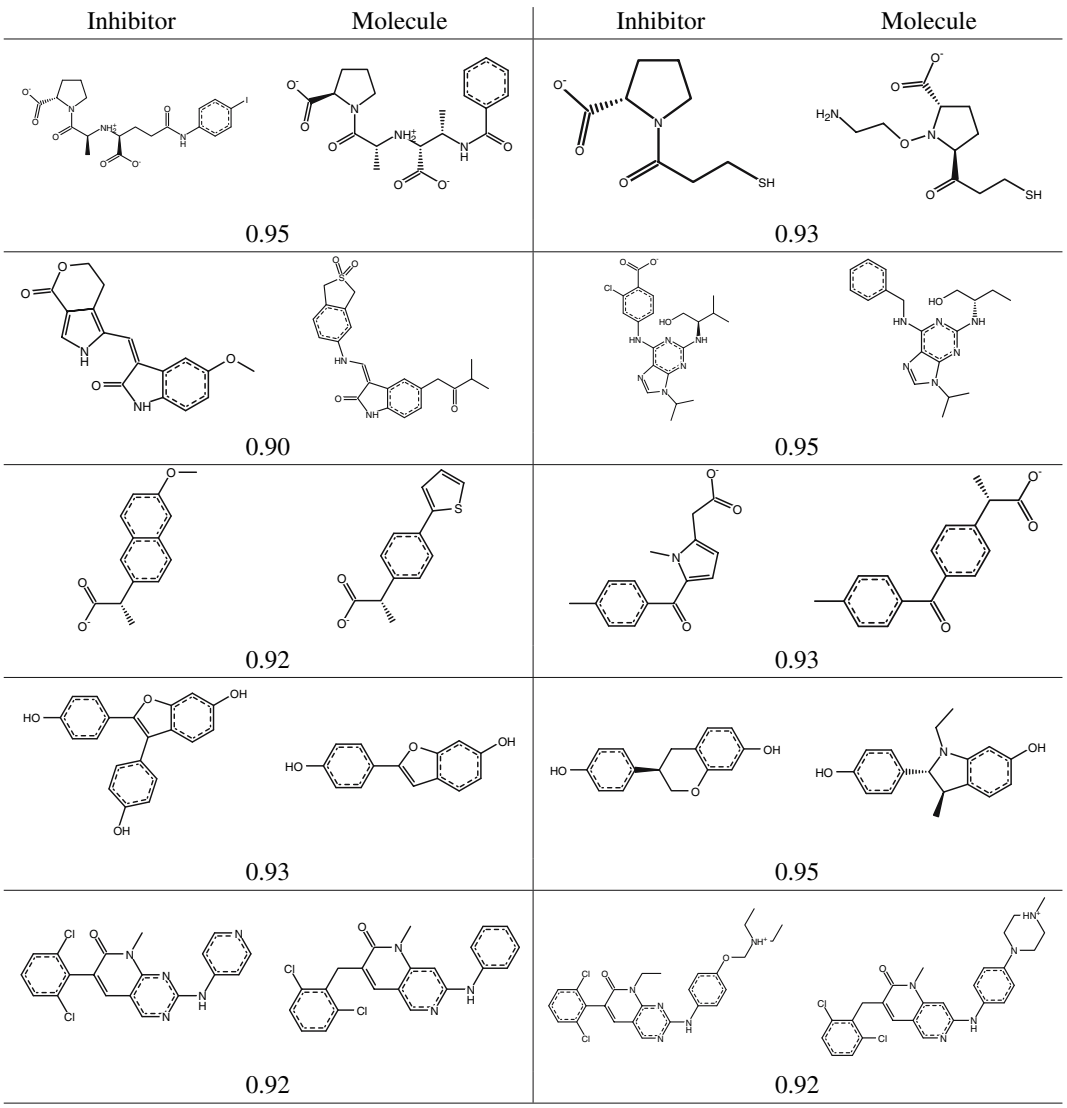


Fig. 7 Selected pairs of inhibitors and molecules obtained from the enumeration for the target classes ACE, CDK2, COX-1, ER agonists and SRC (from top to bottom). Feature tree similarity values are given below each compound pair

At first we characterized the fragment spaces and got surprising results. The individual spaces differ quite significantly regarding their properties. Most of them are infinite in size and the larger ones are also connected. The smaller spaces decompose into several components meaning that certain combinations of fragments cannot occur in the same molecule. From the results we obtained from the enumeration, we demonstrated that it is possible to enumerate the complete chemical subspace for several target classes under multiple physicochemical constraints. In most of the cases, this could be done within minutes to a few hours on a standard workstation computer (Table 6).

We additionally carried out feature tree similarity analyses to quantify the similarity of the resulting molecules to the inhibitors used for generating the fragments. The results show that molecules with reasonable to high similarity were generated for all classes. Further analysis of the individual distributions revealed that the outcome of the enumeration does neither depend on individual properties of the fragment space nor on a linear combination of those. The results rather reflect the overall characteristics of each space. Therefore, the underlying fragments must also have a significant effect. For some target classes, there are probably very specific structural features whereas other classes are more populated with general chemical structures. These results demand further investigation that cannot be part of this publication but will be subject of future research.

Some of the spaces were not considered at all due to the low number of theoretically accessible molecules or because the resulting fragment space was actually not a ''real'' space, being decomposed in many components, containing only a few fragments or being very small in size. For some spaces, the number of molecules is too large for enumeration under constraints. Again, this is due to the overall characteristics of the fragment spaces and no linear dependence on simple properties could be identified that determines whether or not a space can be enumerated on a standard workstation computer.

In our experiments, a clear indication for spaces that could not be enumerated is the number of theoretically accessible molecules consisting of up to five fragments. This value did not exceed 10^8 in all successful calculations. This is also due to the relatively broad property ranges we used in this first (straight-forward) validation. For a real application scenario, where there typically is a lot more (expert) knowledge available, the number of constraints will be higher and the individual ranges also more precisely defined. As a consequence, even the larger spaces will be amendable to enumeration and probably even the combined fragment space, containing up to 10^{14} potentially drug-like molecules. The potency of the approach presented in this paper will be further increased the more knowledge there is available. Therefore, the enumerator opens new views on chemical spaces, focused library design and targeted compound collections, which can be helpful in the early phases of the drug discovery process.

Acknowledgments The authors thank Andrea Zaliani (Lilly Research Laboratories, Hamburg, Germany) for careful reading of the manuscript and Tora Pommerencke, Christoph Müller and Florian Krull, who developed some useful ideas how to improve the performance of the enumeration during their time as project students. Additionally, we thank Patrick Maaß for help in generating the fragments out of the inhibitors sets. We further thank all partners in the NovoBench project for fruitful feedback and the german federal ministry of education and research for funding (grant 313324A).

References

1. Walters WP, Stahl MT, Murcko MA (1994) Drug Discov Today 4:160
2. Oprea TI, Gottfries J (2001) J Comb Chem 3:157
3. Lipinski C, Hopkins A (2004) Nature 432:855
4. Dobson CM (2004) Nature 432:824
5. Rarey M, Degen J, Reulecke I (2007) Docking and scoring for structure-based drug design. Bioinformatics – from genomes to therapies, vol 2. Wiley-VCH Verlag GmbH & Co. KGaA, Weinmheim, p 541
6. Rarey M, Stahl M (2001) J Comput-Aided Mol Design 15:497
7. Schneider G, Lee M-L, Stahl M, Schneider P (2000) J Comput-Aided Mol Design 14:487
8. Lewell XQ, Judd DB, Watson SP, Hann MM (1998) J Chem Inf Comput Sci 38:511
9. WDI: World Drug Index, Derwent Information
10. Rarey M, Dixon JS (1998) J Comput-Aided Mol Design 12:471
11. Schneider G, Fechner U (2005) Nat Rev 4:649
12. Degen J, Rarey M (2006) ChemMedChem 1:854
13. Rarey M, Kramer B, Lengauer T, Klebe G (1996) J Mol Biol 261:470
14. Weber L (2005) QSAR Comb Sci 24:809
15. Ghose AK, Viswanadhan VN (2001) Combinatorial library design and evaluation. Marcel Dekker Inc
16. Knuth DE (1968) The Art of computer programming – fundamental algorithms. Addison-Wesley
17. Weininger D, Weininger A, (1986) J Chem Inf Comput Sci 29:97
18. Knott GD (1971) ACM 1:175
19. Bentley JL (1975) Commun ACM 18:509
20. Huang N, Shoichet BK, Irwin JJ (2006) J Med Chem 49:6789
21. Lipinski CA, Lombardo F, Dominy BW, Feeney PJ (1997) Adv Drug Deliv Rev 23:3
22. Wildman SA (1999) J Chem Inf Comput Sci 39:868
23. SYBYL mol2 file format. Tripos Inc
24. Sadowski J, Rudolph C, Gasteiger J (1990) Tetrahedron Comput Methodol 3:537
25. Sadowski J, Schwab CH, Gasteiger J (2006) Corina, version 3.10, Molecular Networks GmbH Computerchemie. Erlangen
26. Stahl M, Rarey M (2001) J Med Chem 44:1035
27. Fricker P, Gastreich M, Rarey M (2004) J Chem Inf Comput Sci 44:1065