

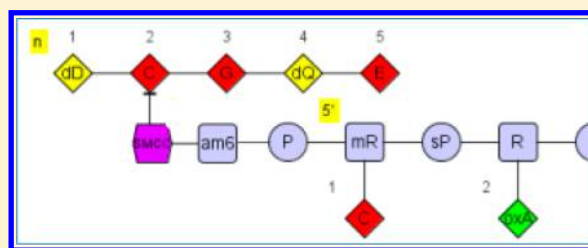
HELM: A Hierarchical Notation Language for Complex Biomolecule Structure Representation

Tianhong Zhang,* Hongli Li, Hualin Xi, Robert V. Stanton, and Sergio H. Rotstein

Pfizer Inc., 35 Cambridge Park Drive, Cambridge, Massachusetts 02140, United States

S Supporting Information

ABSTRACT: When biological macromolecules are used as therapeutic agents, it is often necessary to introduce non-natural chemical modifications to improve their pharmaceutical properties. The final products are complex structures where entities such as proteins, peptides, oligonucleotides, and small molecule drugs may be covalently linked to each other, or may include chemically modified biological moieties. An accurate in silico representation of these complex structures is essential, as it forms the basis for their electronic registration, storage, analysis, and visualization. The size of these molecules (henceforth referred to as “biomolecules”) often makes them too unwieldy and impractical to represent at the atomic level, while the presence of non-natural chemical modifications makes it impossible to represent them by sequence alone. Here we describe the Hierarchical Editing Language for Macromolecules (“HELM”) and demonstrate its utility in the representation of structures such as antisense oligonucleotides, short interference RNAs, peptides, proteins, and antibody drug conjugates.



INTRODUCTION

For small molecules, there exist a number of formats for the in silico representation of chemical structures. These include the following: (1) MDL's molfile format, which is coordinate-based with connection tables;¹ (2) Daylight's SMILES (Simplified Molecular Input Line Entry System) format, a graph-based inline notation;^{2–4} (3) Chemical Markup Language (CML), which is XML-based with connection tables;⁵ and (4) the IUPAC International Chemical Identifier (InChI), which is a nonproprietary identifier for chemical substances.⁶ In addition, there are many commercial and open source software systems based on one or more of these formats that enable the depiction, storage, and visualization of small molecules.

For oligomeric structures such as peptides, a method called CHUCKLES was developed which represents molecules at both the sequence and atomic structure levels.⁷ CHORTLES was later added as an extension of CHUCKLES to handle the representation of oligomeric mixtures.⁸ It is worth noting that the monomer sequence notation in CHUCKLES and CHORTLES is very similar to SMILES, where monomer IDs are concatenated into a string with special characters to indicate branches, cycles, and mixtures. SYBYL Line Notation (SLN) was introduced to represent a wide range of chemical structures, including organic molecules, polymers, and combinatorial libraries.^{9,10} SLN can be viewed as an extension to SMILES, where molecular fragments can be treated as macro atoms and incorporated in the notation directly. Additionally, a protein line notation (PLN) was developed and used to build biocheminformatics databases, where modified peptides and proteins can be treated as chemical structures as well as sequences.¹¹

The hierarchical structure information of complex biomolecules is challenging to represent in a concise notation. For example, a therapeutic agent could consist of a modified peptide conjugated to an antibody via a chemical linker. Although traditional notation formats are capable of representing each individual component (peptide, antibody, and chemical modifier), they generally fall short in representing the combination of the component structure types and how these components are connected to one another. Recently, the notation format SCSR (Self-Contained Sequence Representation) was developed to bridge the gap between bioinformatics and cheminformatics.¹² SCSR uses the enhanced v3000 molfile format, where connection tables are used to describe large molecule structure and pseudo-atom structures. In this paper, we describe HELM, the Hierarchical Editing Language for Macromolecules, which was developed concurrently to SCSR. The two methods take different approaches to the problem with HELM being more “SMILES like” and SCSR more “Molfile like”. As with small molecules, where the decision to use Molfiles vs SMILES depends on the specific application or task at hand, we expect that the HELM and SCSR notations will coexist and the choice of which one to use at any given time will be made on a case-by-case basis. In HELM the structure hierarchy consists of four levels (*Complex Polymer*, *Simple Polymer*, *Monomer*, and *Atom*) with higher level components being defined as a combination of lower level components. This approach allows us to use a small molecule structure format such as SMILES to represent monomers and

Received: April 19, 2012

Published: September 4, 2012

CHUCKLES type notation to represent simple polymers. Complex polymer notation only needs to deal with how simple polymers are connected. Hierarchy information is built into the notation language itself, which allows the structure to be analyzed and depicted at whichever level is deemed appropriate at the time.

MATERIALS AND METHODS

Design Considerations. The design goal for HELM was to enable the representation, registration, storage, analysis, and visualization of an ever-growing set of complex biomolecular structures. The notation language needed to be simple yet versatile enough to represent all types of polymeric structures. More importantly, notations needed to be interoperable and systematic enough to be generated and parsed by computer programs. Specific requirements included the following:

1. Hierarchical structure information encoding: Structure hierarchy should be encoded in the notation and extractable from it. This enables analysis and visualization of the structure at different levels depending on the context.
2. Simple polymer type extensibility: Although proteins, peptides, and oligonucleotides are the most commonly studied biological polymers, the notation language should allow the addition of new types of polymers. The creation of a new polymer type should be considered when new polymer backbone chemistry is introduced.
3. Monomer extensibility: For each simple polymer type, the addition of new monomers should be allowed when new monomer structures are introduced. For example, it should be easy to add unnatural amino acids to the “protein” polymer type.
4. Convertibility to atom-level chemical structure representation: In order to convert the notation to a full atom-by-atom chemical structure representation, it is necessary to have a precise definition of connections between monomers and simple polymers. This capability provides an integration point for small molecule informatics tools.
5. Convertibility to sequences: For amino acid and nucleotide polymers, the notation should allow the conversion to natural amino acid or nucleotide sequences. This capability provides an integration point for sequence-based bioinformatics tools.

Structure Hierarchy. The structure hierarchy consists of the following four levels, in order of increasing granularity: complex polymer, simple polymer, monomer, and atom. Due to HELM’s hierarchical nature, high level components can be described using lower level components.

A complex polymer is comprised of simple polymers which can be optionally connected to each other. As an example, a therapeutic agent might contain one peptide and one oligonucleotide connected via a chemical linker. This complex polymer structure has three simple polymers (peptide, oligonucleotide, chemical linker) and two connections (peptide to chemical linker and oligonucleotide to chemical linker).

A simple polymer is comprised of monomers of the same polymer type. To reduce complexity at this level, we define simple polymers as single linear chains; therefore, simple polymer representation does not need to handle chain branching and cyclization. For specific polymer types such as PEPTIDE, explicit connection rules between monomers are

defined. Furthermore, monomer attachment points and connection rules determine monomer directionality in the polymer chain allowing differentiation between compounds such as clockwise and anticlockwise cyclic peptides. For example, if we define two amino acid monomers, A and G, we could represent a peptide as AGAA, from which we can infer that this simple polymer contains four amino acid monomers connected via amide bonds from N terminus to C terminus.

A monomer is comprised of atoms and bonds and can be represented by a known chemical structure format such as Molfile or SMILES. Each monomer in a simple polymer has to have a unique ID (such as A or G in the peptide example presented above), which is used in the simple polymer notation. Finally, attachment point is an integral part of the monomer definition, which is used to describe the connection with other monomers. In addition to being the building blocks for simple polymers, monomers and their attachment points are used to describe the connections between simple polymers in complex polymers.

Monomer Definition. Monomers have the following set of properties: structure, ID, attachment points, natural analog, polymer type, monomer type, and name:

- **Structure:** The structure is the atom-bond representation of the monomer. This can be generated with existing cheminformatics tools and stored as SMILES strings or Molfiles. A monomer can contain query atoms, but this results in a nonspecific structure.
- **ID:** The ID is required and must be unique within a given polymer type. However, different simple polymer types may use the same monomer ID. For example, a monomer with an ID of A can be used to represent the amino acid alanine in a PEPTIDE polymer, and it can also be used to represent the base adenine in an RNA polymer.
- **Attachment Points:** Attachment points are specified using R groups, each of which needs to have a unique name (e.g., R1, R2, R3). The number of attachment points on a monomer dictates how many bonds can be formed between it and other monomers. It is possible for a monomer to have no attachment points, which means that it can only be used standalone. To fully define an attachment point, it is required to specify the R group in the structure and its capping group. A capping group is a chemical fragment, such as a hydroxyl group, that will be used if no other monomers are connected to the attachment point in the final structure. For example, if we have an R1 group in the structure, and the capping group for R1 is a hydroxyl, we can specify the attachment point as R1–OH. The R groups in attachment points must match those in the structure.
- **Natural Analog:** Each monomer can optionally have a natural analog. For example, selenocysteine is a non-natural amino acid with an ID of seC, which is very similar to the natural amino acid cysteine (C), so the natural analog of seC is C. The natural analog is useful when converting simple polymer notation to amino acid and nucleotide sequences.
- **Polymer Type:** Each monomer must belong to a simple polymer type such as PEPTIDE or RNA.
- **Monomer Type:** Each polymer type can have two types of monomers: *backbone* and *branch*. Backbone mono-

mers are used to form the polymer chain, and branch monomers are used to form short branches from backbone monomers. For example, in RNA polymers, sugar and phosphate linkers are backbone monomers, and the base is a branch monomer.

- **Name:** Whereas the ID of a monomer is generally a very short descriptor (e.g., “C”), the name is a more complete description of the chemical entity (e.g., “cysteine”).

As an example, the full definition of the amino acid monomer alanine in the PEPTIDE polymer is shown in Figure 1. As the

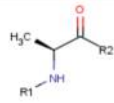
Structure	
SMILES	<chem>C[C@H](N([*])C([*])=O)[r.\$:::_R1:::_R2:\$]</chem>
ID	A
Attachment Points	R1-H R2-OH
Natural Analog	A
Polymer Type	PEPTIDE
Monomer Type	Backbone
Name	L-Alanine

Figure 1. Definition of alanine monomer in PEPTIDE polymer.

structure shows, it has two attachment points, R1 at the nitrogen atom on the amino group and R2 at the carbon atom on the carbonyl group. Furthermore, the capping group for R1 is hydrogen (–H) and the capping group for R2 is hydroxyl (–OH).

Simple Polymer Notation Specification. Since simple polymers are defined as single linear chains of monomers of the same polymer type, it is not necessary to deal with mixtures, long chain branches, and cyclization in simple polymer notation. Consequently, we can use a simplified version of CHUCKLES to represent simple polymer. Simple polymer notation is based on the monomer definition, particularly the monomer type and attachment points. For backbone monomers, we specify R1 and R2 as the chain extending attachment points, and R3 as the branching attachment point. For branch monomers, we specify R1 as the default attachment point to backbone monomers. In simple polymer, chain extending connections are always between R1 of one backbone monomer and R2 of another backbone monomer, and branching connections are always between R3 of a backbone monomer with R1 of a branch monomer. Additional attach-

ment points can exist, but their connections to other monomers need to be described in complex polymer notation.

Simple polymer notation is a line notation which reads from left to right. For polymer chain extension, we simply append the ID of the new monomer to the right of the existing chain. For example, assuming we have three backbone monomers with the IDs A, B, and C, then the polymer notation ABC represents two chain extending connections, from A to B, and from B to C. Furthermore, the backbone is formed by connecting R2 from the first monomer to R1 from the second monomer and so on, thus determining the directionality of the monomers in the polymer chain.

For branching connections between backbone monomers and branch monomers, we first enclose the ID of the branching monomer in parentheses “()”, then append to the right of the backbone monomer. For example, if we have a backbone monomer A, and a branch monomer B, then the simple polymer notation A(B) represents a branching connection from A to B that connects R3 from the backbone monomer to R1 from the branch monomer.

When both a backbone and a branch monomer are connected to the same backbone monomer, the branch monomer takes precedence in the notation. For example, if we have backbone monomers A and C, and branch monomer B, then the simple polymer notation A(B)C represents two connections, one is a chain extending connection from A to C, and the other is a branching connection from A to B.

For non-natural monomers with multiletter monomer IDs, square brackets “[]” are used to enclose the ID. In the above example, if we replaced backbone monomer C with a modified backbone monomer mC, the simple polymer notation would become A(B)[mC].

Furthermore, a period symbol “.” is used to separate simple polymer notation into groups. For example, the notation A(B)C.A(B)C.A(B)C.A(B)C describes not only how the twelve (12) monomers are connected but also how the polymer chain is broken into four groups.

Simple polymer notation always starts with a backbone monomer but can end with either a backbone or branch monomer. Given that monomer directionality is encoded within the notation, AB and BA represent two different polymer structures. In AB, the bond is formed between R2 on monomer A and R1 on monomer B. In BA, the bond is formed between R2 on monomer B and R1 on monomer A.

1. Simple Peptide Polymer (PEPTIDE). Peptides and proteins are amino acid sequences with a backbone of amide bonds. In the context of polymer notation, the polymer type PEPTIDE applies to all polymers with amide bonds between monomers, which are natural or synthetic amino acid residues. The

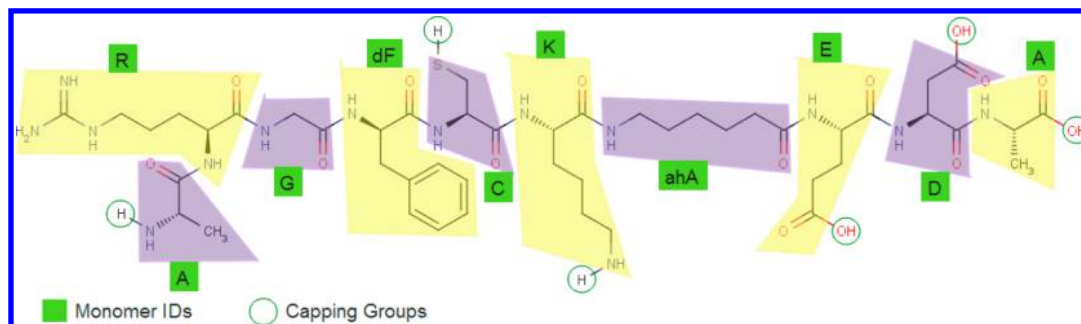


Figure 2. Chemical structure from simple PEPTIDE notation: A.R.G.[dF].C.K.[ahA].E.D.A.

PEPTIDE polymer type only has backbone monomers and no branch monomers. Natural amino acids have single letter IDs, while synthetic amino acids have multiletter IDs. In the case of a synthetic amino acid without a natural analog, the letter X is used as its natural analog. A list of commonly used amino acid monomers is given in the Supporting Information.

On the basis of simple polymer notation specification and the PEPTIDE monomers listed in the Supporting Information, an example PEPTIDE polymer could be represented as:

A.R.G.[dF].C.K.[ahA].E.D.A

The following can be derived from the notation:

- There are ten (10) amino acids in the sequence, as there are nine (9) period symbols.
- The fourth and seventh monomers are non-natural amino acids, as they have multiletter IDs, dF and ahA, which are enclosed within square brackets.
- The single letter amino acid sequence using natural analog replacements is ARGFCKXEDA.

As an illustration, Figure 2 shows the chemical structure of the PEPTIDE polymer represented with the above notation, where monomers and capping groups are highlighted. It can be seen that the amino attachment point (R1) on the first monomer (A) is not used in the chain extension and is capped with H atom. It can be inferred that the first backbone amide bond is formed between the carbonyl attachment point (R2) on the first monomer A and the amino attachment point (R1) on the second monomer (R). The same pattern is repeated for the rest of the monomers. The carbonyl attachment (R2) on the last monomer (A) is not used and is capped with OH group.

It should be noted that PEPTIDE notation represents amino acid sequences from N-terminus to C-terminus, since R1 is on the nitrogen atom and R2 is on the carbon atom in the monomer definition.

2. Simple Nucleotide Polymer (RNA). Antisense oligonucleotides and siRNAs are common biological polymers, which have a sugar–phosphate backbone and base side chains. The polymer type RNA applies to all nucleotide polymers including DNAs and RNAs. Since we choose RNA as the polymer type, DNA is treated as modified RNA. There are three different approaches to defining RNA monomers. In the first approach, a nucleotide is the monomer unit, and the RNA polymer is composed of nucleotide monomers. In the second approach, a nucleotide is broken into nucleoside and phosphate linker, and the RNA polymer is composed of nucleoside and phosphate linker monomers. In the third approach, a nucleoside is further broken into sugar and base, and the RNA polymer is composed of sugar, phosphate linker, and base monomers. Since chemical modifications can be made on sugar, linker, and base, the first two approaches can lead to an unnecessarily large number of monomers due to the combinatorial possibilities among them. We therefore opted for the third approach in defining RNA monomers, where we have sugars and phosphate linkers as backbone monomers and bases as branch monomers. As with the PEPTIDE polymer type, natural monomers have single letter IDs while modified monomers have multiletter IDs. A list of commonly used RNA monomers is given in the Supporting Information.

On the basis of simple polymer notation specification and the RNA monomers listed in the Supporting Information, an example RNA polymer could be represented as:

R(A)P.[mR](U)[sP].R(G)P.R([5meC])P.[dR](T)P.
[dR](T)P.[dR](T)P.[dR](T)P

The following can be derived from the notation:

- There are eight (8) nucleotides in the sequence, since there are seven (7) period symbols.
- There are seven (7) modified monomers, since there are seven (7) square brackets. The second nucleotide has a modified sugar mR and a modified linker sP. The fourth nucleotide has a modified base 5meC, and the fifth, sixth, seventh, and eighth nucleotides have a modified sugar dR.
- The single letter nucleotide sequence using natural replacements is AUGCTTTT.

It should be noted that RNA polymer notations represent nucleotide sequences from the 5' end to 3' end, since R1 is on the 5' position and R2 is on the 3' position in the sugar monomer.

3. Additional Simple Polymer Types. Additional polymer types with different backbone chemistries can be added to the notation language. For example, to represent polysaccharides, we just need to add a saccharide polymer type (SAC), and define a set of monosaccharide monomers. Each monomer could have a R1 attachment point on a hydroxyl oxygen atom and a R2 attachment point on an anomeric carbon atom. Connecting the oxygen atom and carbon atom would form a glycosidic bond and extend the polysaccharide chain.

4. Nonspecific Polymer Type (CHEM). Specific simple polymers have well-defined backbone chemistries, and monomer connection rules are specified. For generic chemical modifiers such as cross-linkers, their connections to other monomers are not predefined. As a result, we created a generic polymer type named CHEM for chemical modifier monomers. These monomers are described with type unknown and can have any number of attachment points. Since connection rules do not exist between CHEM monomers, each simple CHEM polymer will have only one monomer. Therefore, connections between CHEM monomers are connections between simple polymers and will be handled in the complex polymer notation. A list of CHEM monomers is given in the Supporting Information which will be used in the Polymeric Structure Examples section.

5. Simple Polymer Notation Summary. There are two types of simple polymers, specific and nonspecific. Specific simple polymers, such as PEPTIDE and RNA, have well-defined backbone chemistries. Monomer connections within a specific polymer are specified. New polymer types with different backbone chemistries can be introduced. The nonspecific polymer (CHEM) is a generic category, and connections between monomers are not predefined. In addition to monomer IDs and monomer connection rules, simple polymer notation introduces the following symbols:

- Square brackets to enclose monomers with multiletter IDs, which generally represent synthetic or non-natural analogues of other monomers
- Parentheses to enclose branch monomers
- Period symbol to separate monomers into logical groups

Complex Polymer Notation Specification. While the simple polymer notation described in the above section is capable of representing single chain polymers, it is inadequate to represent complex polymer structures where multiple

polymer chains exist, and are optionally connected to each other. Complex polymers can take many different forms, including:

- Conjugated polymers, in which a simple polymer chain is covalently linked to a chemical linker (e.g., protein–drug conjugate)
- Cross-linked polymers, in which multiple simple polymer chains are covalently linked to a multifunctional chemical linker (e.g., antisense oligonucleotide dimers)
- Polymer mixtures, in which multiple simple polymer chains are mixed together (e.g., siRNA)

The complex polymer notation is based on graph representation, where simple polymer chains are treated as graph nodes and connections between simple polymer chains are treated as graph edges. In addition to the description of simple polymer chains and their connections, complex polymer notation can include the description of hydrogen bonds and polymer attributes. The complex polymer notation has the following format:

```
ListOfSimplePolymers$
ListOfConnections$
ListOfHydrogenBonds$
ListOfAttributes$
PlaceHolderForExtension
```

The dollar sign symbol “\$” is used to delimit each section, and it is not required to have any white space after it. There are five sections in the complex polymer notation, and only the first section (ListOfSimplePolymers) is required. As implied by its name, the last section is a place holder for extension, and it not used currently.

1. Simple Polymer List. The first section of the complex polymer notation is the list of simple polymers. To generate the list of simple polymers in the structure, we first identify all simple polymer chains (including chemical modifiers), and assign a unique polymer ID to each chain in the format of PolymerType#, where PolymerType is the type of the polymer chain, such as PEPTIDE, RNA, and CHEM, and # is an arbitrary number to make the polymer ID unique. Generally, it starts with 1 and increments by 1 for each occurrence of the same type of polymer chains. For example, RNA1 is the polymer ID for the first RNA polymer, and CHEM2 is the polymer ID for the second chemical modifier. Polymer IDs must be unique within a complex polymer notation. Each polymer ID will have a corresponding simple polymer notation, which describes its structure. Each simple polymer is represented as

```
PolymerID{SimplePolymerNotation}
```

where curly brackets “{}” are used to enclose the simple polymer notation. Once all simple polymer chains are represented, they are concatenated together using vertical pipe “|” as a delimiter. The delimiter is required only in the middle, and sorting order in the list is not important from a structural point of view.

As an example, the following represents the list of simple polymers for a complex polymer which contains one PEPTIDE polymer, one RNA polymer, and one CHEM polymer:

```
PEPTIDE1{A.R.G.[dF].C.K.[ahA].E.D.A}|
RNA1{R(A)P.[mR](U)[sP].R(G)P.R([5meC])P.
[dR](T)P.[dR](T)P.[dR](T)P}|
```

```
CHEM1{SS3}
```

2. Connection List. The second section of the complex polymer notation is the list of connections between simple polymers. To generate the list of connections in a complex polymer, we first need to identify all monomer pairs involved in the connections. For each monomer pair, we then need to identify the parent simple polymer ID as defined in the list of simple polymers, the monomer position in its parent simple polymer, and the attachment point used in the connection. With this information, each connection can be described as:

```
SourcePolymerID,TargetPolymerID,
```

```
SourceMonomerPosition:SourceAttachment-
```

```
TargetMonomerPosition:TargetAttachment
```

There are three components in each connection description, separated by the comma symbol “,”. The first is the source polymer ID, which identifies the source polymer. The second is the target polymer ID, which identifies the target polymer. The third is the detailed connection description, which identifies the source monomer position and attachment point as well as target monomer position and attachment point. The dash symbol “-” is used to separate source monomer from target monomer, and the colon symbol “:” to separate monomer position from attachment point. Source and target can be switched in the connection notation.

In the example given in the Simple Polymer List section, assuming that there is a bond between the R2 attachment point on the P monomer at the 3’ end in RNA1 and the R1 attachment point on chemical modifier SS3 in CHEM1, we can generate the following connection information. The source polymer ID is RNA1, and the target polymer ID is CHEM1. The source monomer position is 21, since 3’ P is the 21st monomer in the RNA polymer notation. The target monomer position is 1, since there is only one chemical modifier monomer in CHEM polymer type. The full connection can be represented as:

```
RNA1,CHEM1,21:R2-1:R1
```

Similar to the list of simple polymers, once all connections are represented, they are concatenated together using a vertical pipe as a delimiter. The delimiter is required only in the middle of the list, and sorting order is not important.

3. Hydrogen Bond List. Sometimes it is desirable to represent hydrogen bonds in a structure. For example, short interfering RNAs (siRNA) are double stranded RNA molecules where hydrogen bonds exist between base pairs. The third section of the complex polymer notation is used for this purpose. Hydrogen bond list representation is similar to connection list representation discussed above, except that no attachment points are used. Instead, a four letter code “Pair” is used to replace the attachment point in the connection string. An example of the hydrogen bond notation for a siRNA is given in the Polymeric Structure Examples section.

4. Attribute List. It is at times useful to include some nonstructure attributes in the complex polymer notation, and the fourth section is used for this purpose. To generate the attribute list, we first identify the attribute name and value, and then associate it with a PolymerID defined in the simple polymer list. Each attribute is represented in the following format:

```
PolymerID{AttributeName:AttributeValue}
```

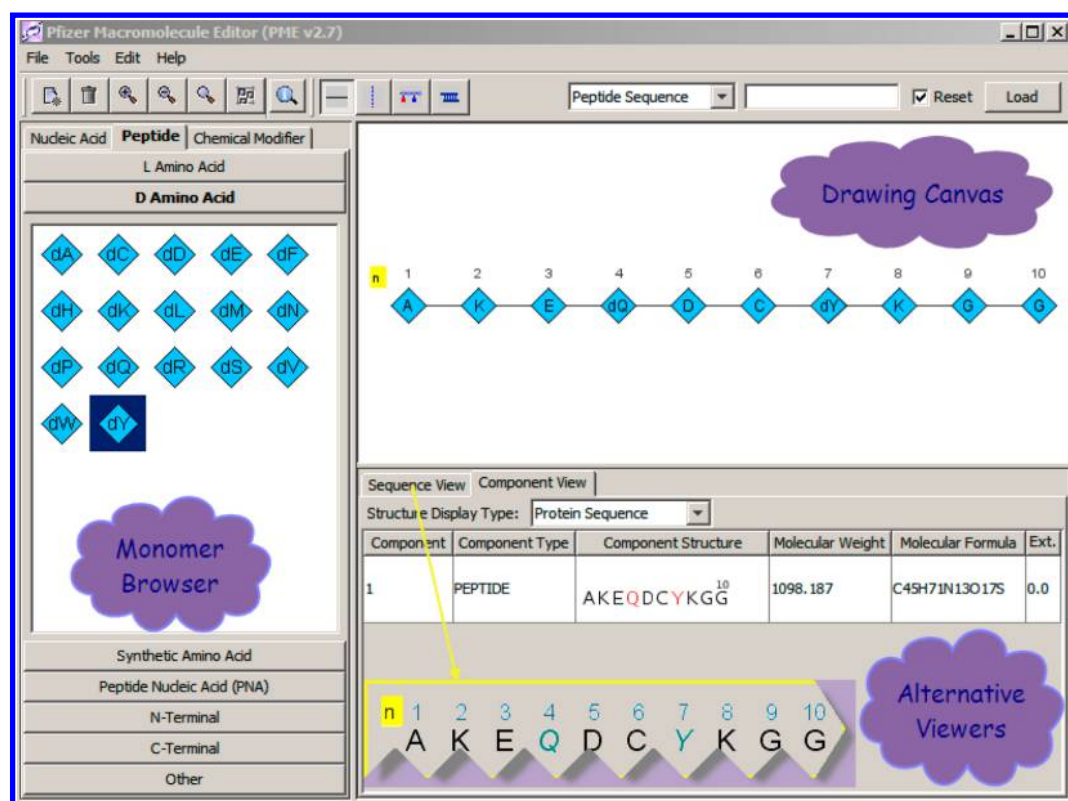


Figure 3. PME window.

Similar to the list of simple polymers and connections, once all polymer attributes are represented, they are concatenated together using a vertical pipe as a delimiter. The delimiter is required only in the middle of the list, and sorting order is not important.

For example, there are two RNA strands in an siRNA, and we can use the following attribute list to indicate RNA1 is the sense strand, and RNA2 is the antisense strand.

RNA1{StrandType:ss}|RNA2{StrandType:as}

5. Cycles and Branches. Since simple polymers are single linear polymer chains, cycles and long branches in complex polymer are realized through connections between simple polymers. For example, a linear peptide polymer could have two separate cysteine monomers (C), which could be connected to form a cyclic peptide via a disulfide bridge. In the complex polymer notation, we will have one simple PEPTIDE polymer, and one intrapolymer chain connection. As another example, a linear peptide could have an aspartic acid monomer (D), where the side chain carboxylic group can connect to the N-terminus of another peptide to form a long branch. This would be represented by two simple PEPTIDE polymers, and one interpolymer chain connection.

6. Complex Polymer Notation Summary. A graph based notation is used to represent complex polymers where simple polymers are treated as graph nodes and connections between them are treated as graph edges. Furthermore, syntaxes are provided to describe hydrogen bonds and polymer attributes. It should be noted that hydrogen bonds and polymer attributes are not essential to the complex polymer structures, and they are ignored in structure manipulations. Symbols used in the complex polymer notation are the following:

- Dollar sign "\$" to separate the major sections of the complex polymer notation
- Curly brackets "{}" to enclose simple polymer notation in the simple polymer list and to enclose polymer attribute in polymer attribute list
- Vertical pipe "|" to separate simple polymers, connections, hydrogen bonds, and polymer attributes
- Comma "," to separate the three components in a connection or hydrogen bond string
- Dash "-" to connect the source and target in a connection or hydrogen bond
- Colon ":" to separate monomer position from attachment point or hydrogen bond code, and attribute name from attribute value

All these symbols, plus the ones used in simple polymer notation, brackets, parentheses, and periods, are considered reserved characters and should therefore not be used in monomer IDs.

Pfizer Macromolecule Editor (PME). As with any language, it is necessary to master both the vocabulary and the grammar in order to become proficient in it. In HELM, the vocabulary is the monomer collection, and the grammar is the set of connection rules and symbols. While it is clear from the examples presented above that it is possible to manually generate notations to represent complex polymer structures, it is much more desirable to have a graphical user interface (GUI) that can simplify the monomer lookup and hide the complexity of the connection rules and syntax. To that end, we have developed the Pfizer Macromolecule Editor (PME), a graphical tool that can be used for managing monomers, sketching and visualizing polymer structures, and calculating molecular properties. PME is meant to serve for biomolecules the same

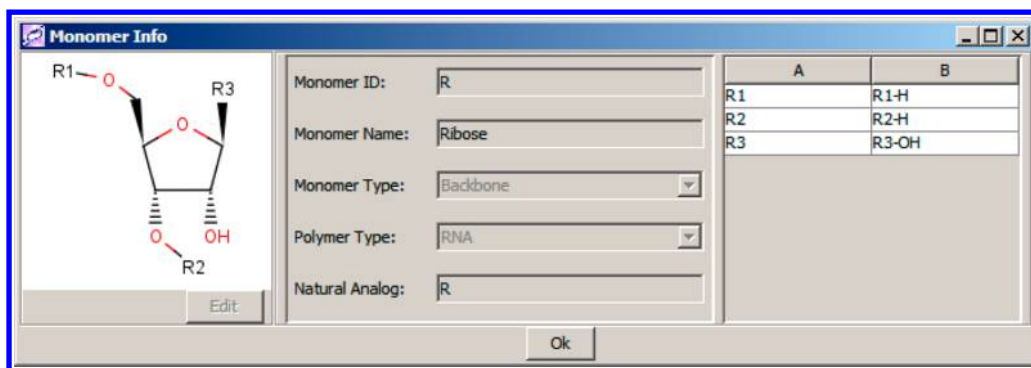


Figure 4. Monomer information for ribose (R) in RNA polymer.

Monomer List

Symbol	Natural Analog	Name	Structure
dA	A	D-Alanine	<chem>C[C@H](N[*])C([*])=O r,\$;...;</chem>
meA	A	N-Methyl-Alanine	<chem>C[C@H](N(C)[*])C([*])=O r,\$;...;</chem>
mA	A	alpha amino iso-butyric acid	<chem>CC(C)(N[*])C([*])=O \$;...;_R1;...;</chem>
A	A	Alanine	<chem>C[C@H](N[*])C([*])=O r,\$;...;_R...</chem>
nA	A	Naphthylalanine	<chem>[*]N[C@H](Cc1ccc2ccccc2c1)C...</chem>
dC	C	D-Cysteine	<chem>[*]N[C@H](CS[*])C([*])=O r,\$;...</chem>
seC	C	SelenoCysteine	<chem>[SeH]C[C@H](N[*])C([*])=O r,...;</chem>
meC	C	N-Methyl-Cysteine	<chem>CN([*])C[C@H](CS[*])C([*])=O...</chem>
pyssC	C	Pyridine disulfide Cysteine	<chem>[*]N[C@H](CSSc1ccccc1)C([*])...</chem>
C	C	Cysteine	<chem>[*]N[C@H](CS[*])C([*])=O r,...;</chem>
meD	D	N-Methyl-Aspartic acid	<chem>CN([*])C[C@H](CC([*])=O)C([*])...</chem>
dD	D	D-Aspartic acid	<chem>[*]N[C@H](CC([*])=O)C([*])=O...</chem>

Monomer Detail

Chemical structure: C[C@H](N(C)R1)C(=O)R2

Monomer ID: meA
 Monomer Name: N-Methyl-Alanine
 Monomer Type: Backbone
 Polymer Type: PEPTIDE
 Natural Analog: A

Buttons: Request, Upload, Register, Structure Sketcher, Close

Figure 5. Monomer Manager.

purpose as tools like ChemDraw,¹³ Marvin Sketch,¹⁴ and Accelrys Draw¹⁵ do for small molecules.

1. PME Overview. As shown in Figure 3, the main window of PME consists of three panels, Monomer Browser on the left, Drawing Canvas at the top right, and Alternative Viewers at the bottom right.

In the Monomer Browser, predefined monomers are grouped into different tabs based on their polymer type and subcategory. The structure is shown when the mouse cursor is placed over each monomer. Double-clicking on a monomer results in its detailed information being displayed, as shown in Figure 4 for the ribose monomer (R) in RNA polymer.

In the Drawing Canvas, monomers from the Monomer Browser can be added, deleted, replaced, and connected to create the desired structure. The Alternative Viewer panel contains a Sequence View tab and a Component View tab. In the Sequence View, the single-letter natural amino acid sequence for peptides, or nucleotide sequence for nucleic acids, is shown. The sequence is derived from the natural analog of each monomer. In the Component View, calculated properties such as molecular weight, molecular formula, and extinction coefficient are listed for each covalently linked structure.

2. Monomer Management. Monomers are the building blocks of polymeric structures. Before a new monomer can be used in a polymeric structure, it has to be defined and registered into the monomer database. During this process, both structure and ID uniqueness are checked within a given polymer type. To control the ID naming convention and avoid uncontrolled growth of the monomer database, we implemented a two stage process where general users can request a new monomer to be added, but the request must be reviewed and approved by a data curator before being added to the active monomer database. Figure 5 shows the Monomer Manager tool, where registered monomers can be browsed and new monomers can be requested, reviewed, and registered.

3. Structure Drawing. While it is possible to build a polymeric structure by connecting monomers one by one, it is generally preferable to use a structure template at the beginning. A structure template can be loaded into the Drawing Canvas by specifying a nucleotide sequence, a peptide sequence, or a HELM notation. It is also possible to load other data formats, as long as they can be converted to HELM notation. Once a starting structure is loaded, it can be further modified. New monomers can be added by dragging them from the Monomer Browser and dropping them into the Drawing Canvas. Connections can be created between monomers by placing the mouse cursor on top of the first monomer, pressing down the left mouse button, and dragging the mouse cursor to the second monomer. For monomers with more than one attachment point available, the user is prompted to select the desired attachment point. To replace one monomer with a different one, the new monomer is dragged from the Monomer Browser and dropped on top of the monomer to be replaced in the Drawing Canvas. To remove monomers and connections, the user selects them by clicking on a monomer or pressing the left mouse button and dragging the mouse cursor to generate a selection area. Once the desired monomers and connections are selected, pressing the delete key erases them. The drawing tool knows the monomer database, understands the connection rules, displays how monomers are connected, and produces the HELM notation automatically.

4. Structure Visualization. The default structure view is the monomer graph view where each monomer is shown as a node and each connection is shown as an edge. In this view, each monomer and its connections are displayed. However, it is possible to generate a chemical structure view, where a monomer is expanded into its atom bond representation. It is also possible to generate a simple polymer component view, where a simple polymer is collapsed into a single polymer node. Furthermore, a hybrid view can be generated, where the chemical structure view, monomer graph view, and simple polymer component view coexist. All these visualization options are made possible by the hierarchical nature of the HELM notation.

RESULTS AND DISCUSSION

Polymeric Structure Examples. To demonstrate the utility of HELM notation language in representing biomolecule structures, we include a few example polymeric structures with their monomer graph views and HELM notations.

1. Linear Peptide. Figure 6 shows the monomer graph view and HELM notation for a linear peptide. The whole structure contains only one simple PEPTIDE polymer, which contains ten amino acid monomers, two of which are non-natural.

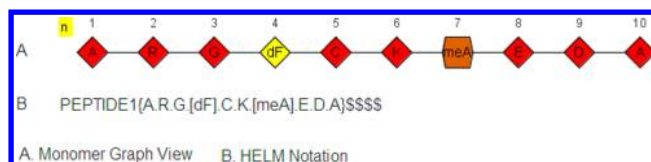


Figure 6. Linear peptide.

2. Linear Oligonucleotide. Figure 7 shows the monomer graph view and HELM notation for a linear oligonucleotide.

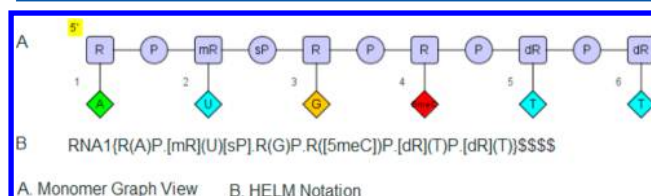


Figure 7. Linear oligonucleotide.

The whole structure contains only one simple RNA polymer, which contains six nucleotides. There are three modified sugar monomers, one modified linker, and one modified base.

3. Cyclic Peptide. Figure 8 shows the monomer graph view and HELM notation for a cyclic peptide. This structure

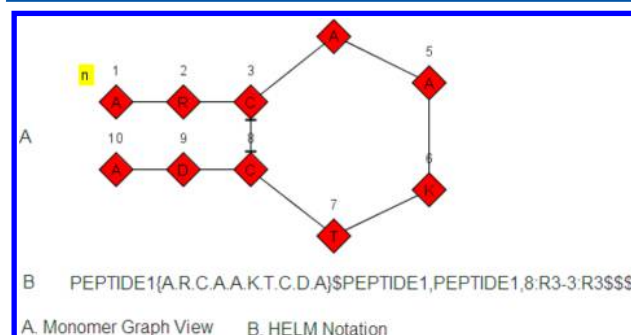


Figure 8. Cyclic peptide.

contains one simple PEPTIDE polymer, which contains ten amino acid monomers. There is an intrachain connection between the third monomer C and the eighth monomer C on the R3 attachment point, which results in a cyclic peptide.

4. Branched Peptide. Figure 9 shows the monomer graph view and HELM notation for a branched peptide. This

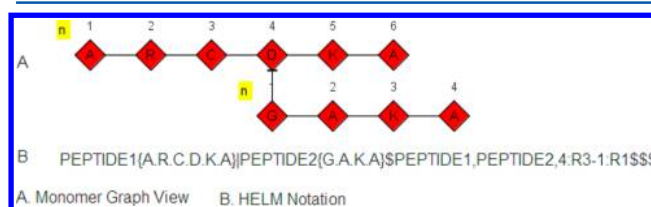


Figure 9. Branched peptide.

structure contains two simple PEPTIDE polymers, one has six amino acid monomers and the other has four amino acid monomers. There is an interchain connection between the fourth monomer D on PEPTIDE1 and the first monomer G on PEPTIDE2. The connection is formed between R3 attachment point on monomer D and R1 attachment point on monomer G. PEPTIDE2 is a branch of PEPTIDE1.

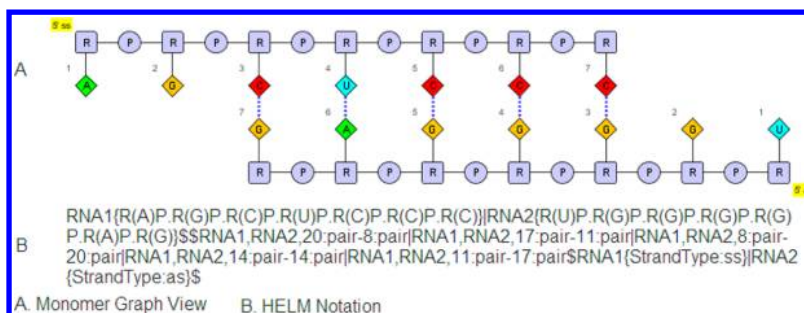


Figure 10. Double stranded RNA.

5. Double Stranded RNA. Figure 10 shows the monomer graph view and HELM notation for a double stranded RNA. This structure contains two simple RNA polymer chains, each has 20 monomers (7 nucleotides). There is no covalent bond between them, but there are five hydrogen bonds. For example, one hydrogen bond is between the eighth monomer on RNA1, which is a base monomer with an ID of C, and the 20th monomer on RNA2, which is a base monomer with an ID of G. RNA1 is annotated as sense strand (StrandType:ss), and RNA2 is annotated as antisense strand (StrandType:ss).

6. *Oligonucleotide Conjugate*. Figure 11 shows the monomer graph view and HELM notation for an oligonucleo-

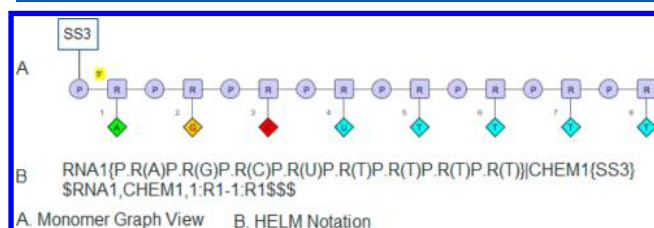


Figure 11. Oligonucleotide conjugate.

tide conjugate. This structure contains one simple RNA polymer, which contains 8 nucleotides with a 5' phosphate monomer (P). It also contains a chemical modifier with the ID of SS3. There is a covalent bond between the R1 attachment point on the first monomer (P) on the RNA polymer and the R1 attachment point on the chemical modifier (SS3).

7. *Oligonucleotide Dimer*. Figure 12 shows the monomer graph view and HELM notation for an oligonucleotide dimerized via a chemical linker. This structure contains two identical simple RNA polymers, each of which has 8 nucleotides with a 3' phosphate. It also contains a chemical modifier with the ID of sDBL_R2 and R3 attachment points on

sDBL are connected to the R2 attachment point of the 3' phosphate (P) monomer.

8. **Oligonucleotide Peptide Conjugate.** Figure 13 shows the monomer graph view and HELM notation for an oligonucleotide and peptide conjugated via a chemical linker. This structure contains one simple RNA polymer, one simple PEPTIDE polymer, and one chemical modifier with the ID of SMCC. There is one connection between the chemical modifier and RNA polymer, and one connection between the chemical modifier and PEPTIDE polymer.

Conversion to Existing Notation Formats. Since HELM notation explicitly represents biomolecule structures and their connections, it is possible to convert it to other notation formats. The HELM notation for the linear peptide polymer shown in Figure 6 and the equivalent SMILES and InChI representations are shown in Figure 14. These representations can, in turn, be converted to additional formats such as Molfile V2000, Molfile V3000, and CML. Additional conversion results are provided in the Supporting Information.

HELM Extension. Since the notation language is monomer-based, each monomer needs to be defined before it can be used in a polymer structure. While this works well for specific simple polymers, it is quite limiting for nonspecific chemical modifiers. As a result, we have extended HELM to support ad hoc chemical modifiers. In this case, the extended SMILES of these chemical modifiers, instead of their IDs, are used in simple CHEM polymer notation. While R groups in the structure are still used to represent potential attachment points, the default capping group for them is hydrogen. Another extension currently being considered is the representation of nonspecific structures (e.g., oligomers where the exact sequence is not known). Additional extensions will likely be necessary in the future to meet specific requirements for complex biomolecule structures as they arise.

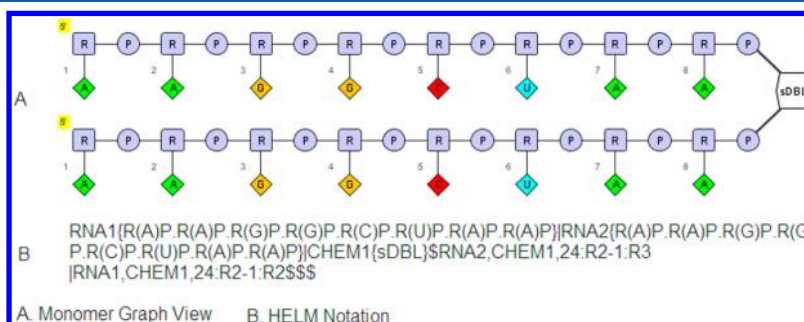


Figure 12. Oligonucleotide dimer.

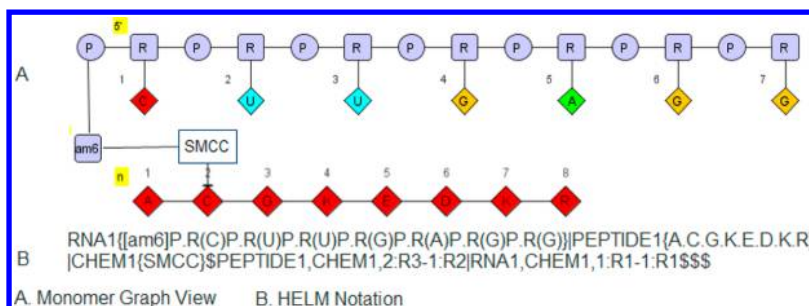


Figure 13. Oligonucleotide peptide conjugate.

HELM	PEPTIDE1{A.R.G.[dF].C.K.[meA].E.D.A)}\$\$\$\$
SMILES	<chem>NCCCC[C@H](NC(=O)[C@H](CS)NC(=O)[C@@H](Cc1ccccc1)NC(=O)CNC(=O)[C@H](CCNC(=N)N)NC(=O)[C@H](C)N)C(=O)N(C)[C@H](C)C(=O)N[C@H](CCC(=O)O)C(=O)N[C@H](CC(=O)O)C(=O)N[C@H](C)C(=O)O</chem>
InChI	1S/C45H72N14O15S/c1-23(47)36(65)54-27(14-10-18-50-45(48)49)38(67)51-21-33(60)53-30(19-26-11-6-5-7-12-26)41(70)58-32(22-75)42(71)56-29(13-8-9-17-46)43(72)59(4)25(3)37(66)55-28(15-16-34(61)62)39(68)57-31(20-35(63)64)40(69)52-24(2)44(73)74/h5-7,11-12,23-25,27-32,75H,8-10,13-22,46-47H2,1-4H3,(H,51,67)(H,52,69)(H,53,60)(H,54,65)(H,55,66)(H,56,71)(H,57,68)(H,58,70)(H,61,62)(H,63,64)(H,73,74)(H4,48,49,50)/t23-,24-,25-,27-,28-,29-,30+,31-,32-/m0/s1

Figure 14. Conversion of HELM notation to SMILES and InChI.

CONCLUSION

We have described HELM, the Hierarchical Editing Language for Macromolecules, which is aimed at enabling the representation of complex biomolecule structures. The notation language is simple but versatile as demonstrated by its usage in peptides, antisense oligonucleotides, siRNAs, proteins, antibodies, conjugates, and combinations thereof. We have also introduced the Pfizer Macromolecule Editor (PME), a graphical tool that implements the HELM notation language, and demonstrated its utility for biomolecule structure drawing, visualization, and molecular property calculations.

With the creation of HELM notation language and the development of the Pfizer Macromolecule Editor (PME), biomolecules can be represented in a standard format and their structures can be visualized in a context-sensitive manner. In a subsequent paper, we will describe how HELM notation is being used to perform biomolecule registration at Pfizer, including a detailed implementation of structural uniqueness-checking, validation, and verification.

ASSOCIATED CONTENT

Supporting Information

Table 1: PEPTIDE monomers. Table 2: RNA monomers. Table 3: CHEM monomers. Table 4: HELM notation to SMILES and InChI conversion results. This material is available free of charge via the Internet at <http://pubs.acs.org>.

AUTHOR INFORMATION

Corresponding Author

*Tel.: 1-617-551-3310. E-mail: Tianhong.Zhang@pfizer.com.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the input and feedback provided by many colleagues at Pfizer, particularly Dr. David Klatte and Dr. Peter Henstock from Research Business Technologies, Dr. Jason Hughes from Biological Profiling, and Dr. Theresa Johnson from the Oligonucleotide Therapeutics Unit. We further wish to thank the former Oligonucleotide Therapeutics Unit and the Research Business Technologies organizations for budgetary and organizational support.

REFERENCES

- (1) Dalby, A.; Nourse, J. G.; Hounshell, W. D.; Gushurst, A. K.; Grier, D. L.; Leland, B. A.; Laufe, J. Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 244–255.
- (2) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.
- (3) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for Generation of Unique SMILES notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97–101.
- (4) Weininger, D. Smiles. 3. Depict. Graphical Depiction of Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 237–243.
- (5) Murray-Rust, P.; Rzepa, H. S.; Wright, M. Development of chemical markup language (CML) as a system for handling complex chemical content. *New J. Chem.* **2001**, *25*, 618–634.
- (6) Heller, S. R.; McNaught, A. D. The IUPAC International Chemical Identifier (InChI). *Chem. Int. [online]* **2009**, *31*, 1.
- (7) Siani, M. A.; Weininger, D.; Blaney, J. M. CHUCKLES: A Method for Representing and Searching Peptide and Peptoid Sequences on Both Monomer and Atomic Levels. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 588–593.
- (8) Siani, M. A.; Weininger, D.; James, C. A.; Blaney, J. M. CHORTLES: A Method for Representing Oligomeric and Template-Based Mixtures. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 1026–1033.

- (9) Ash, S.; Cline, M. A.; Homer, R. W.; Hurst, T.; Smith, G. B. SYBYL Line Notation (SLN): A Versatile Language for Chemical Structure Representation. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 71–79.
- (10) Homer, R. W.; Sqanson, J.; Jilek, R. J.; Husrt, T.; Clark, R. D. SYBYL Line Notation (SLN): A Single Notation To Represent Chemical Structures, Queries, Reactions, and Virtual Libraries. *J. Chem. Inf. Model.* **2008**, *48*, 2294–2307.
- (11) Jensen, J. H.; Hoeg-Jensen, T.; Padkjær, S. B. Building a BioChemformatics Database. *J. Chem. Inf. Model.* **2008**, *48*, 2404–2413.
- (12) Chen, W. L.; Leland, B. A.; Durant, J. L.; Grier, D. L.; Christie, B. D.; Nourse, J. G.; Taylor, K. T. Self-Contained Sequence Representation (SCSR): Bridging the Gap between Bioinformatics and Cheminformatics. *J. Chem. Inf. Model.* **2011**, *51* (9), 2186–2208.
- (13) *ChemDraw*, version 12.0; PerkinElmer: Waltham, MA, 2010.
- (14) *MarvinSketch*, version 5.10.3; ChemAxon: Budapest, 2012.
- (15) *Accelrys Draw*, version 4.0; Accelrys: San Diego, CA, 2010.