CHUCKLES: A Method for Representing and Searching Peptide and Peptoid Sequences on Both Monomer and Atomic Levels

Michael A. Siani,*,† David Weininger,‡ and Jeffrey M. Blaney†

Chiron Corporation, 4560 Horton Street, Emeryville, California 94608, and Daylight Chemical Information Systems, 419 East Palace, Santa Fe, New Mexico 87501

Received August 4, 1993®

Dual representation of peptide and non-peptide structures in a chemical database as atomic-level molecular graphs and sequence strings permits chemical substructure and similarity searches as well as sequence-based substring and regular expression searches. CHUCKLES interconverts monomer-based sequences with SMILES, which represent atomic-level molecular graphs. Forward-translation maps peptide or other sequences into SMILES. Back-translation extracts monomer sequences from SMILES. This approach permits a generalized representation of monomers allowing user specification of any monomer. CHUCKLES allows mixing of atoms with user-defined monomer names; that is, monomer representation is consistent with SMILES notation. In addition, oligomer branching and cyclization are handled.

INTRODUCTION

Peptides play an important role in drug lead-discovery. They are rapidly synthesized by standard techniques and are increasingly a part of chemical libraries, which are screened against various receptors and enzymes. Peptoids (oligo(*N*-substituted)glycine polymers) have also been described recently as another source of molecular diversity appropriate for chemical libraries.[2] As peptides, peptoids, and other structures proliferate, in both pure form and as mixtures, we must find new ways of storing and accessing their chemical information.[1-5] We will describe an approach for storing and searching mixtures in a subsequent paper.

While it has been sufficient until now to store and access peptide data via the amino acid sequence as simple text strings, we need to standardize at the atomic level. As man-made monomers proliferate, and basic peptide structure diverges from standard L- and D-amino acids, and even from the standard peptide backbone, chemical databases must permit access to these compounds both at the sequence level and at the atomic level.

We developed CHUCKLES, a method for converting peptide sequences to atomic-level SMILES (forward-translation) and converting SMILES back to sequences (back-translation). This method employs a user-defined monomer reference table which pairs a monomer name with atomic chemical SMILES and SMARTS. Extensions to simple monomer names, which may be strung together to create polymers, allow us to specify branching and cyclization. Monomer names are consistent with any SMILES representation, so monomer names can be mixed with SMILES, which permits the specification of arbitrary peptide–organic structures. When treating monomers as atomic units, we refer to them as *residue units*.

SMILES is an alphanumeric string representation of a molecule's atoms and bonds, the same information one would find in an extended connection table. For example, the SMILES for alanine is NC(C)C(=O)O. The isomeric SMILES is a molecular structure represented as a graph with chirality specified when necessary. For example, the isomeric SMILES for alanine, including chirality, is N[C@@H](C)C(=O)O. SMARTS is a superset of the SMILES language that specifies a pattern in a molecular graph.[6]

REPRESENTATION RATIONALE AND SEARCHING METHODS

Maintaining the atomic representation, rather than only a sequence representation, provides several advantages. Initial conversion of sequence information to a SMILES representation allows us to verify that the compound is chemically valid. The SMILES representation may be used to regenerate sequence strings later, perhaps after monomer names are changed. Note that many monomers may have multiple, synonomous names. Users may change their minds from time to time regarding choice of a "standard" name. With structures represented correctly at an atomic level, sequences can be automatically regenerated if the monomer table is changed. Dual representation, as both an atomic structure and a sequence string permits database searching and display by either string search or substructure/similarity search.[7,8] This representation also makes it convenient to store peptide and peptoid structures along with standard organic structures in a single database.

Traditional sequence searching is limited to a finite number of monomers, usually consisting of the 20 standard L-amino acids plus a few variants. Suites of programs from IntelliGenetics[9] and the Genetics Computer Group (GCG)[10] permit rapid, complex queries of natural peptide sequences based upon a sequence alphabet. Generally, searches with these packages are performed on a library of known peptide sequences such as GenBank,[11] SWIS-PROT,[12] and PIR.[13] Although the sequence database may be user-supplied, it is alphabet-based only; no detailed structural information is considered. In addition, one is limited to a standard set of letters, A–Z, where each entry may be represented by a single character only; thus one cannot have more than 26 different monomers in the user-supplied library. With the IntelliGenetics and GCG packages, certain amino acids can be classified according to physicochemical properties (e.g., hydrophobic, hydrophilic, charged, aromatic, and aliphatic side chains) or any user-supplied categories. However, assumptions about constant backbone chemistry and the integrity of a monomer within the context of an oligomer result in limitations which become apparent as the diversity of monomers increases.

† Chiron Corp.
‡ Daylight Chemical Information Systems.
® Abstract published in *Advance ACS Abstracts*, March 1, 1994.

0095-2338/94/1634-0588$04.50/0

O'Donnell et al.[14] created an approach for investigating atom-type assignments and interconverting between different molecular structure files. Their approach is similar to ours for extracting peptide sequences from an atomic structure. This method identifies the polypeptide nature of a molecule by matching backbone atoms against backbone atom patterns. After the locations of the side chains are identified, the side chains are identified. A known, fixed backbone is assumed for the polypeptide.

Rapid development of new monomer side chains and increasingly varied backbone chemistry require a more comprehensive treatment of sequence searching. Our method of extracting monomer sequences from atomic representations lets us handle any new monomer. We have applied this technique to a peptoid set which initially consists of N-substituted glycines (NSGs) with the standard amino acid side chains on the nitrogen.[2] With a new "submonomer" synthetic process,[5] we are now able to include over a thousand readily available monomers. Because of the rapid proliferation of these monomers, we have also automated the naming of monomers. Moreover, we are also exploring alternative backbone structures (manuscript in preparation). As polymers become more complex and less like standard peptides, the need for both sequence-based and atomic representations is more apparent. The approach described here works for peptides of any size, including proteins.

We are currently using the search strategies in MERLIN (Daylight Chemical Information Systems, Santa Fe, NM). MERLIN displays the database as a spreadsheet, where each row represents a different compound. Columns contain the following: SMILES, isomeric SMILES, sequence, and other data (e.g., receptor binding affinity, *in vivo* activity, etc.). Columns may be searched by substring, regular expression, substructure, and similarity. Searches on different columns can be combined with Boolean operators. In addition, oligomeric sequences are much easier to interpret in sequence form in the spreadsheet than the atomic form; either representation (or both) can be displayed. Substructure and similarity searches are performed by entering a SMILES or SMARTS representation or by drawing a molecule. MERLIN performs the substructure search on SMILES or isomeric SMILES, leaving entries which satisfy the search in the spreadsheet. Sequence searches are performed by entering a subsequence (substring) or regular expression.[15] For example, a subsequence may be specified as "AlaThrLysArgAla". A sample regular expression, with wild cards in positions 3 and 4, is "AlaThr[A-Z][a-z][a-z]*[A-Z][a-z][a-z]*Ala". The regular expression specifies exact matches in residue positions 1, 2, and 5 but allows a match to any single residue at positions 3 and 4. The "[A-Z][a-z][a-z]*" construct corresponds to a single capital letter followed by zero or more lower-case letters, the naming convention for a single residue. Searching at the substring and substructure level on sequences and molecular graphs gives us greater flexibility than previous search methods. Also, all compounds may be represented and stored in this format.

MONOMER AND SEQUENCE NOTATION

We define a notation for monomers and various linking methods (straight chain, branched, and cyclized) for representing sequences. Our monomer and sequence notation correlates directly with atomic SMILES, which permits us to always generate a valid atomic representation from our sequence representation.

All standard amino acids and nonstandard monomers are listed in a table where the monomer code is associated with a SMILES and SMARTS. For example, alanine is represented by the name Ala, the SMILES NC(C)C(=O), and the SMARTS N[C@@H]([CH3])C(=O). Monomer names always start with an upper-case letter and are followed by lower-case letters only. Note that in the SMILES and SMARTS the first atom, nitrogen, and the last, carbonyl carbon, do not have their complete valence filled. This allows us to link them together by concatenation. The "@" character lets us specify chirality about an atom center. The "[C@@H]" in Ala indicates L-alanine. Alternatively, "[C@H]" indicates the D-amino acid.

SMARTS, a superset of the SMILES language, permits one to specify a variable number or a limit on the number of covalent bonds to non-hydrogen atoms from a particular atom. In a SMILES, the unspecified valences are assumed to be filled with hydrogens. For example, the SMILES for a cysteine residue is N[C@@H](CS)C(=O). The sulfur may be bonded to any other single atom; if the atom is not specified, it is assumed to be hydrogen. In the SMARTS, N[C@@H]-([CH2][S;D2])C(=O), we specify the exact number of hydrogens on some atoms. In addition, the "[S;D2]" indicates the sulfur is bonded to two (from D2) non-hydrogen atoms, in this case the methylene β-carbon and some other unspecified atom. [A new representation, not available at the time of implementation of CHUCKLES, allows us to be more specific about the number of bonds. One may represent "[S;D2]" more accurately as "[SX2H0]". "D" represents explicit connectivity, so "[S;D2]" can match "S" in both cysteine-(linked) and cysteine(unlinked) if the hydrogens are explicitly represented. "[SX2H0]" will only match two total connections and no hydrogens.] We can use this to indicate a cysteine which is attached to something via the sulfur (e.g., disulfide).

Specifying branched or cyclized polymers requires an extension of this monomer naming method since one may only extend (by concatenation) the sequence string to the left or right at the two *ends*. Branching and cyclization are indicated by new monomer names which end with "x". These monomers contain one branch point, a site where an atom may be bonded to a non-hydrogen atom. Cysx is a branched cysteine fragment: N[C@@H]([CH2][S;D2])C(=O). The sulfur has two covalent bonds to non-hydrogen atoms. We represent a disulfide bond as AlaCysx1AlaCysx1Thr. The Cysx's with the same index number are bonded via their branch points, yielding a disulfide bridge. See Figure 1b. If we wish to specify the C-terminal free acid on the Thr, the sequence is AlaCysx1AlaCysx1ThrO.

In addition to branch points for "side chains", we use "y" and "z" for N-terminal and C-terminal connections. For example, one C-to-N terminally cyclized peptide, Alay1-Ser-Arg-Ser-Thrz1, depicts a peptide bond from the carbonyl carbon of Threonine to the nitrogen of Alanine. See Figure 2.

Our choices of x, y, and z are arbitrary and are selected to be compatible with SMILES (to avoid conflict with characters that can appear in SMILES). We chose x to indicate cross-linking or side-chain-connected linkages. Note that "n", "c", and "s" would conflict with the usage of these characters in aromatic rings in SMILES; in standard SMILES notation, a lower-case atom name indicates the atom is a member of an aromatic ring.

For branches that are not cyclized, we make use of the ".", the SMILES break character which indicates a disconnected structure. For example, a side-chain protecting group like Boc may be attached to a lysine side chain by GlyLysx1Ala.Bocx1. See Figure 3. The dot indicates that

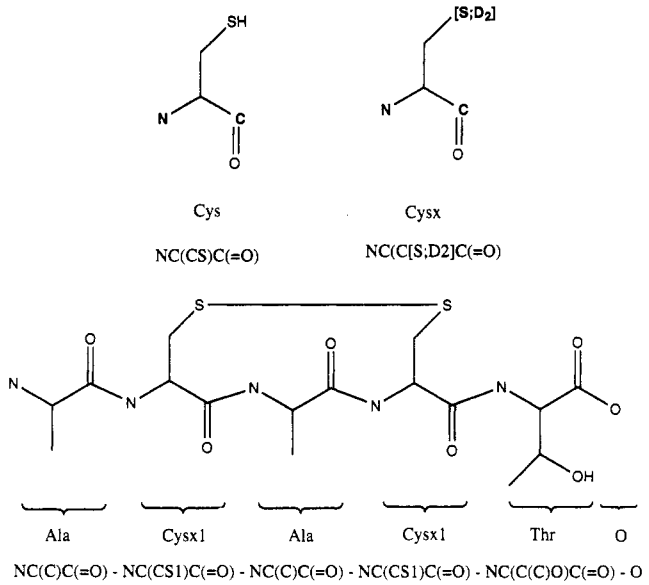

Figure 1. (a, top) SMARTS representation for regular and branched monomers from macro lookup table. (b, bottom) Example of sequence with disulfide: AlaCysx1AlaCysx1ThrO. Illustration of relationship between molecular graph, monomer-based sequence and atomic SMILES.

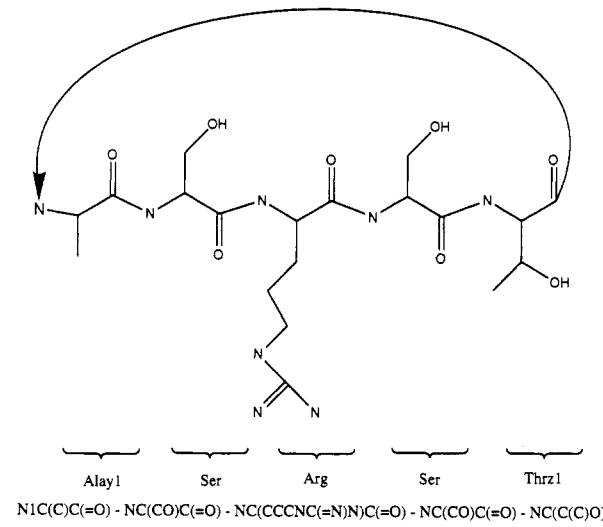

Figure 2. C-to-N terminal cyclized peptide: Alay1SerArgSerThrz1. Illustration of relationship between molecular graph, monomer-based sequence, and atomic SMILES.

the Ala and the Bocx are not attached in the linear SMILES, but the index 1 on both the Lysx and Bocx indicates that they are bonded through their branching points. While one could use parentheses to depict branching notation such that GlyLysx(Bocx)Ala is equivalent to GlyLysx1Ala.Bocx1, the latter representation is desirable when back-translating.

A more complex scenario involves creating a branched peptide by attaching another segment to a side chain. The AlaLysx1SerAla.AlaPheValz1 peptide illustrates a branching of the lysine side chain; see Figure 4. Note we have an effective reversal of the branch segment sequence such that the C-terminus of the AlaPheVal is attached to the lysine.

Because monomer naming conventions are consistent with standard SMILES notation, we can specify arbitrary peptide–organic structures. See Figure 5, for example.

FORWARD-TRANSLATION

Forward-translation converts a user-input peptide sequence into a valid SMILES. See the forward-translation flowchart, Figure 6.

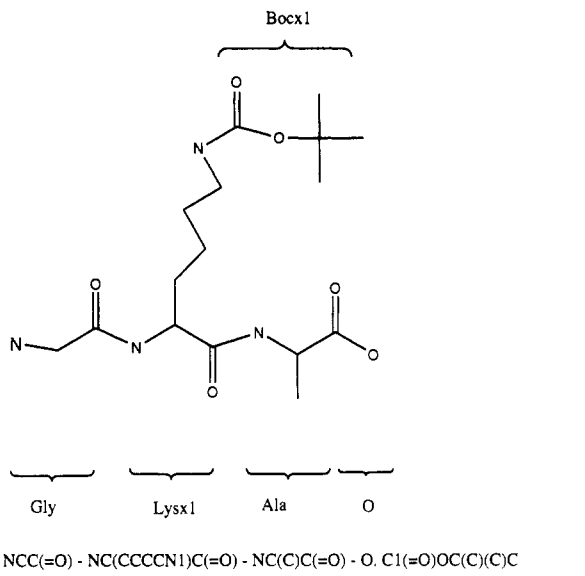

Figure 3. Boc-protected lysine in peptide GlyLysx1AlaO.Bocx1. Illustration of relationship between molecular graph, monomer-based sequence, and atomic SMILES.

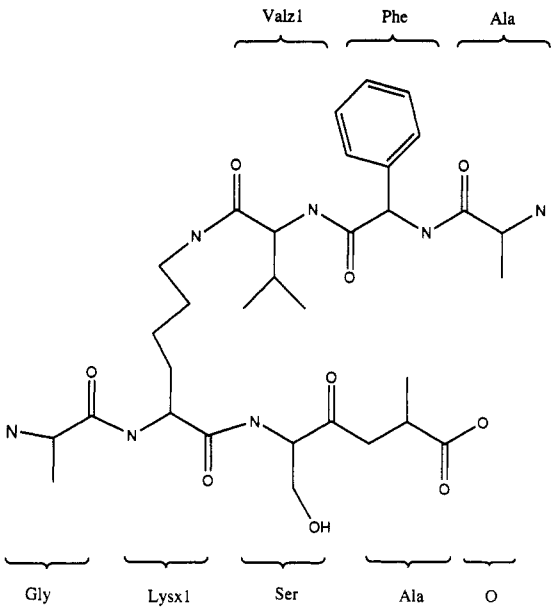

Figure 4. Example of branched peptide with reversed C-to-N segment attached to lysine side chain: AlaLysx1SerAlaO.AlaPheValz1.

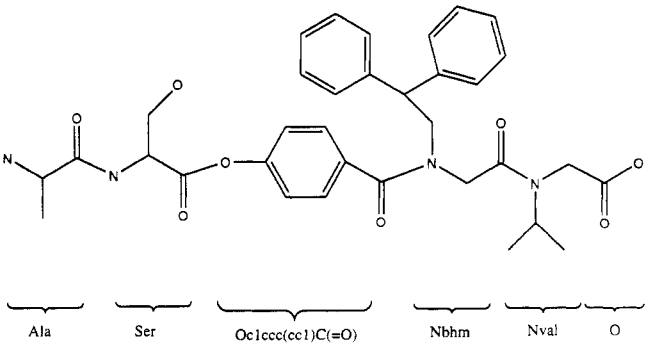

Figure 5. Example mixed peptide/peptoid monomer–organic structure sequence: AlaSerOc1ccc(cc1)C(=O)NbhmNvalO. Ala and Ser refer to the standard amino acids L-alanine and L-serine. Nbhm is a benzhydryl–methyl N-substituted glycine (peptoid). Nval is the peptoid version of valine.

Our approach uses SMILES to link adjacent monomers in an oligomer chain by simply concatenating monomer SMILES together. Each monomer is associated with a SMILES in a

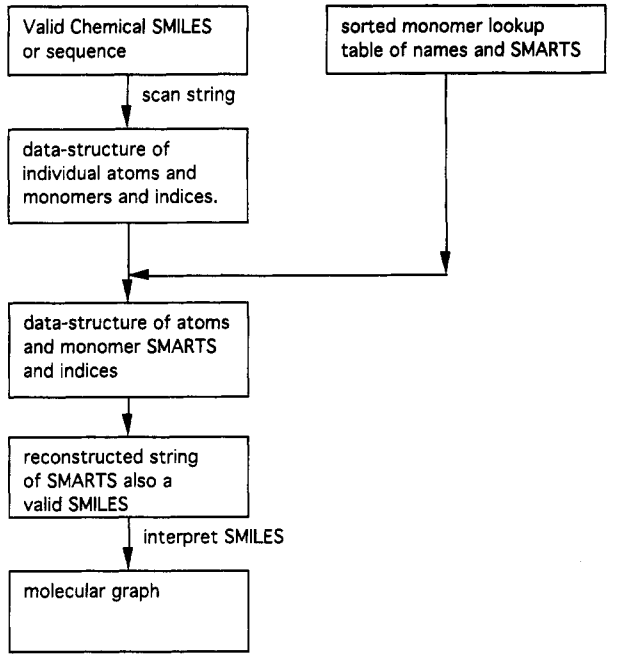

Figure 6. Forward-translation flowchart: Conversion of monomer-based sequence to atomic SMILES.

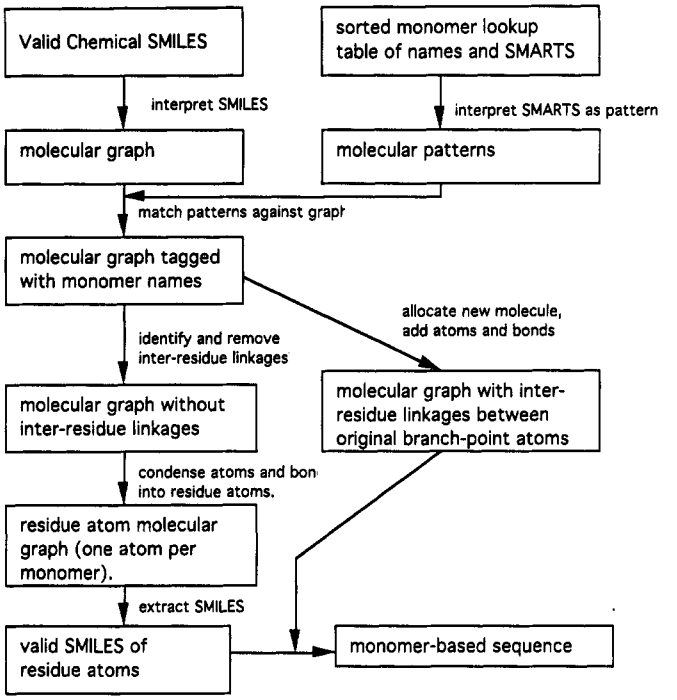

Figure 7. Back-translation flowchart: Conversion of atomic SMILES to monomer-based sequence. Identification of monomer substructure matches and subsequent generation of sequences were implemented using the Daylight Chemical Information SMILES and SMARTS object-oriented toolkits. These subroutine libraries permit creation, traversal, and modification of molecular graphs directly from C and FORTRAN programs. The libraries allow modification of the molecular graph at the atomic and bond levels, acting through handles to the molecule data type.

lookup table. Alanine is defined as NC(C)C(=O). Thus, the dipeptide AlaAla is NC(C)C(=O)NC(C)C(=O). The peptide bond between the carbonyl carbon of the first Ala and the nitrogen of the second is then implicit in the concatenated SMILES. From the SMILES a valid molecular graph may be generated.

In branched and cyclized oligomers, the sequence monomer indices are carried over to the branch points in the SMILES representation. Consider a disulfide bridge between two cysteines in a sequence. In the sequence AlaCysx1Ala-Cysx1Ala, the Cysx's with the index number 1 are bonded via their branch points, yielding a disulfide bridge. Since the SMILES for alanine is NC(C)C(=O) and for cysteine is NC(CS1)C(=O), this translates to

NC(C)C(=O)NC(**CS1**)C(=O)NC(C)C(=O)-
NC(**CS1**)C(=O)NC(C)C(=O)

We have concatenated the SMILES associated with each monomer and mapped the monomer indices from the sequence into the SMILES.

For branching oligomers, such as the side-chain protecting group Boc attached to a lysine side chain, GlyLysx1Ala.Bocx1, the indices are carried through during translation from sequence to concatenated SMILES. The SMILES associated with the monomers may be concatenated in the sequence order. This yields

NCC(=O)NC(CCCN**1**)C(=O)NC(C)C(=O).
C1(=O)OC(C)(C)C

BACK-TRANSLATION

Back-translation is the method for converting a SMILES, which represents a valid molecular graph, to a monomer-based sequence. See the back-translation flowchart, Figure 7.

The first step in back-translation is to identify substructures which correspond to substructure queries (patterns) generated from the monomer SMARTS table. The monomer patterns are sorted by the number of atoms, such that the monomer with the greatest number of atoms is first and the smallest

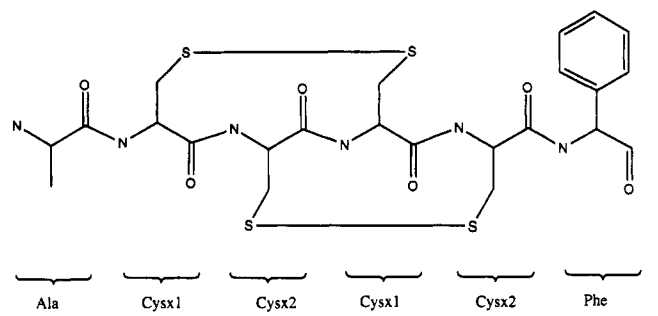

Figure 8. Molecular graph representation of user-input peptide, containing two disulfides. Monomer matches indicated by names are shown below relevant atoms and bonds in graph.

monomer is last. Monomer patterns are matched against the molecular graph by descending size. As a set of atoms and bonds matches, those atoms in the molecular graph are tagged as MATCHED by attaching the monomer name. See Figure 8. This approach guarantees that the largest possible monomer will match. For example, a Phe pattern will match before benzyl. We impose this ordering to avoid premature fragmenting of the molecule by small monomers and protecting or capping groups. For monomers which are chemically equivalent (such as sarcosine and the alanine peptoid equivalent, Nala), the ordering is arbitrary, and the matched name depends upon the ordering in the original monomer table.

After all potential monomer matches have been found, the atom-based molecular graph is transformed into a residue-based graph. For each monomer, all the atoms are condensed into a single residue unit, with links (bonds) to "adjacent" monomers and atoms preserved. Each residue is given a unique identifier which includes the monomer type (e.g., Ala). For all residue units present in the graph, we assign successive

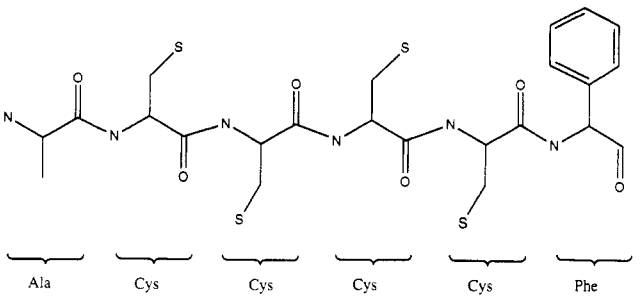

Figure 9. Molecular graph with interresidue linkages broken. Disulfide bonds have been removed so that each residue has at most two neighbors.

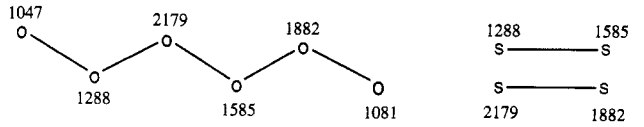

Figure 10. (a, left) Molecular graph of condensed residue atoms, with no interresidue bonds except along the backbone. Each unique atom identifier corresponds directly to a residue in Figure 8. (b, right) Molecular graph of interresidue linkages. In this case, the disulfides are represented as bonds between pairs of sulfers from the corresponding cysteines in Figure 8.

unit numbers starting at 1000 which uniquely identify each residue type. Bonds from branch point atoms (indicated in the monomer SMARTS pattern) to other residues or atoms outside the residue are then broken; see Figure 9. For each one of these branch point connections, called interresidue links, we add the pair of atoms and bond to a new molecule. This additional molecular graph will hold the interresidue links for later reincorporation; see Figure 10a,b.

Ther terminating residue units, which generally contain unsatisfied bonds (implicit in the monomer SMARTS) are flagged, so that subsequent generation of a SMILES will start or end with those residues. The N-terminus can be detected by finding a residue unit bonded to only one other residue unit. This N-terminal residue unit is weighted so that it appears first when translated into a residue-based sequence.

The residue unit-based molecular graph, which may still contain elemental atoms (which did not match any monomer patterns), is then converted to a canonical SMILES. This yields a SMILES where each atom is a coded, unmatched atom, or a residue unit with no interresidue links. Interresidue links through branch points were excised to permit generation of a SMILES with a linear sequence based upon a backbone. For example, the coded SMILES generated from the residue unit molecular graph in Figure 10a is [1047][1288][2179]-[1585][1882][1081].

This concept of backbone is inherent in the monomer table SMARTS, where monomers may be concatenated along the backbone only. This approach circumvents preconceived notions of a specific backbone chemistry. The backbone definition is thus augmented beyond the standard amino acid backbone of –NCC(=O)– to nonbranching atoms lying between the first and last atoms in the input residue reference table.

Bonds in the interresidue molecular graph may then be mapped back into the SMILES via unique atom labels. For every bond in the interresidue molecular graph, a unique identifier is appended to the SMILES for each atom label, starting with %99. After the SMILES corresponds to all of the bonding of a peptide, the coding is then mapped back to monomers found in the lookup table. Incorporating interresidue linkage information into the residue-based SMILES (above) yields the following coded SMILES: [1047]-[1288]%99[2179]%98[1585]%99[1882]%98[1081]. Conversion to standard monomer names yields: AlaCysx%99-Cysx%98Cysx%99Cysx%98Phe.

LIMITATIONS

Limitations arise mainly from redundancy in the user-specified monomer lookup table. Because a monomer may have more than one name, back-translation may be inconsistent. As monomer names are added to the lookup table, back-translation of the same compound may yield different sequences. Also, sequence extraction from SMILES can go too far. That is, proliferation of very small monomers can result in meaningless sequences, particularly when applying back-translation to non-oligomer compounds. The implementation described here does not handle monomers with more than three external connections (two main chain and one side chain); however, monomer definitions can be extended to handle more.

IMPLEMENTATION

CHUCKLES was written in "C" using the Daylight Programming Toolkits (Daylight Chemical Information Systems, Santa Fe, NM) on a Silicon Graphics System (IRIX 4.0.5) and currently contains approximately 3000 lines of code. This implementation is fast enough to permit interactive data entry into a database by sequence, graphical depiction, or atomic SMILES. It is general enough to handle complex cyclic and branched molecules including proteins. For example, CHUCKLES is able to extract an atomic representation from the following sequence for insulin:

GlyIleValGluGlnCysx1Cysx2AlaSerValCysx1Ser-
LeuTyrGlnLeuGluAsnTyrCysx3Asn.PheValAsnGln-
HisLeuCysx2GlySerHisLeuValGluAlaLeuTyrLeu-
ValCysx3GlyGluArgGlyPhePheTyrThrProLysAla

Note that insulin consists of two separate chains, joined by a disulfide bridge. Forward translation of insulin into the atomic representation takes 0.8 CPU s on a Silicon Graphics 320, R3000 MIPS chip. Back-translation from the atomic representation to the above sequence takes 8.2 CPU s.

CONCLUSIONS

Previously available chemical database systems provided the ability to store molecules as atomic structures or as simple, linear strings encoding oligomer (e.g., peptide or nucleotide) sequences, but could not provide simultaneous access to both types. A parallel representation of peptides, both as text strings and atomic structures, permits string-based sequence searching and substructure or similarity searching. Thus, peptide, peptoid, peptide–organic hybrids, nucleotides, and other organic molecules can all be stored and searched conveniently in a single system. CHUCKLES provides a straightforward, table-driven approach to interconvert between a linear sequence code and an atomic structure.

REFERENCES AND NOTES

(1) 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.
(2) Simon, R. J.; Kania, R. S.; Zuckermann, R. N.; Huebner, V. D.; Jewell, D. A.; Banville, S. C.; Ng, S.; Wang, L.; Rosenberg, S.; Marlowe, C. K.; Spellmeyer, D.; Tan, R.; Frankel, A. D.; Santi, D. V.; Cohen, F. E.; Bartlett, P. A. Peptoids: A modular approach to drug discovery. *Proc. Natl. Acad. Sci. U.S.A.* **1992**, *89*, 9367–9371.

(3) Geysen, H.; Meloen, R.; Barteling, S. Use of Peptide Synthesis to Probe Viral Antigens for Epitopes to a Resolution of a Single Amino Acid. *Proc. Natl. Acad. Sci. U.S.A.* **1984**, *81*, 3998–4002.
(4) Houghten, R. A.; Pinilla, C.; Blondelle, S. E.; Appel, J. R.; Dooley, C. T.; Cuervo, J. H. Generation and Use of Synthetic Peptide Combinatorial Libraries for Basic Research and Drug Discovery. *Nature* **1991**, *354*, 84–86.
(5) Zuckermann, R. N.; Kerr, J. M.; Kent, S. B. H.; Moos, W. H. Efficient Method for the Preparation of Peptoids [Oligo(*N*-substituted glycines)] by Submonomer Solid Phase Synthesis. *J. Am. Chem. Soc.* **1992**, *114*, 10646–10647.
(6) *Daylight Software Manual*: Theory; Daylight Chemical Information Systems: Santa Fe, NM, 1993.
(7) Willet, P. *Similarity and Clustering in Chemical Information Systems*; John Wiley & Sons, Inc.: New York, 1987.
(8) Johnson, M. A.; Maggiora, G. M. *Concepts and Applications of Molecular Similarity*; John Wiley & Sons, Inc.: New York, 1990.
(9) IntelliGenetics Suite 5.4; IntelliGenetics: Mountain View, CA, 1992.
(10) Wisconsin Sequence Analysis Package; Genetics Computer Group (GCG): Madison, WI, 1993.
(11) GenBank Genetic Sequence Data Bank; National Institutes of Health: Mountain View, CA, 1993.
(12) SWISS-PROT Protein Sequence Data Bank; University of Geneva: Geneva, Switzerland, 1993.
(13) PIR Protein Identification Resource; National Biomedical Research Foundation: Washington, D.C., 1993.
(14) O'Donnell, T. J.; Rao, S. N.; Koehler, K.; Martin, Y. C.; Eccles, B. A General Approach for Atom-Type Assignment and the Interconversion of Molecular Structure Files. *J. Comput. Chem.* **1991**, *12*, 209–214.
(15) Muster, J.; Birns, P.; Lurnix *Unix Power Utilities for Power Users*; Management Information Source, Inc.: Portland, OR, 1989.