HELM Search Project

Andrea Chlebíková

September 19, 2014

Acknowledgements

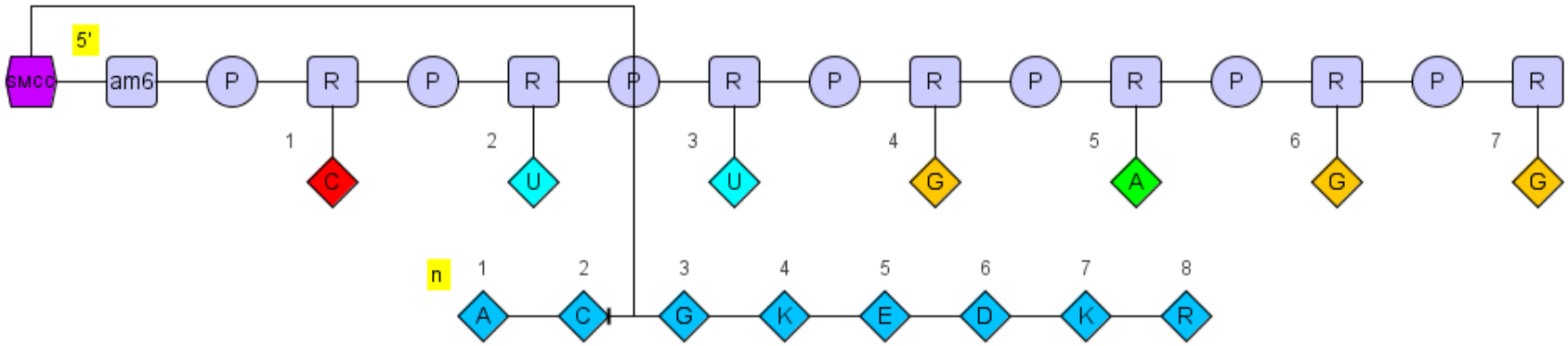
- Dr Andreas Bender and group, Centre for Molecular Informatics, Department of Chemistry, University of Cambridge
- Dr Tianhong Zhang, Pfizer
- Dr Roland Knispel, ChemAxon

Overview

HELM - Hierarchical Editing Language for Macromolecules

Example

RNA1{[am6]P.R(C)P.R(U)P.R(U)P.R(G)P.R(A)P.R(G)P.R(G)}|
PEPTIDE1{A.C.G.K.E.D.K.R}|CHEM1{SMCC}$
PEPTIDE1,CHEM1,2:R3-1:R2|RNA1,CHEM1,1:R1-1:R1$$$



Aims of the Summer Search Project

- Develop open source search facility (beyond existing identity check used to identify uniqueness, based on canonicalising the notation)
- Extend existing Java-based Notation Toolkit (which already has methods for generating and manipulating the notation)
- Focus on exact submatch search - similarity searching important in future
- Methods for finding structures which match search query or combination of search queries

Overview of Approach

- Makes sense to search by various criteria - peptide/nucleotide sequence, presence of particular monomers, modifications, etc.
- Need ability to search based on combination of these criteria
- Existing small molecule substructure search methods in practice too slow on whole structures
- Providing tools for natural equivalent sequence searching and substructure searching on a variety of levels, along with methods for combining the results allows for variety of search criteria

Subsequence Search

...AUUCUGGCAAAG...

- Need to extract natural equivalent sequence data from HELM notation
- Perform regular expressions match on lists of strings obtained this way (allows for flexibility regarding matching gaps/ambiguity characters)
- Standard sequence notation well-defined (IUB/IUPAC)
- Small snag: backbone cyclic polymers - easily overcome as explained later

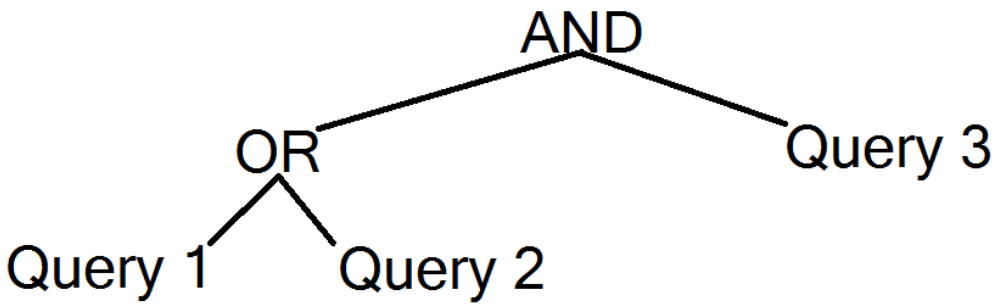
Substructure Search

- Existing SMILES search tools implemented in Java in open-source CDK (Chemistry Development Toolkit)
- Need to convert HELM notation to SMILES notation - either for whole structure or parts of it
- For parts, need to cap break locations appropriately
- Standard SMARTS notation for substructure query

Combining the Two Approaches

(Query 1 OR Query 2) AND Query 3

- Boolean connectors AND/OR to define logic, alongside negation
- Logic is stored in tree form
- Input possible manually via methods to define the tree
- . . . or by using a simple parser which interprets infix input
- In future user input of tree structure graphically may be a good idea
- Search optimisation by reordering partial queries - search is carried out left-to-right, bottom-to-top



Overview of Toolkit Search Functionality

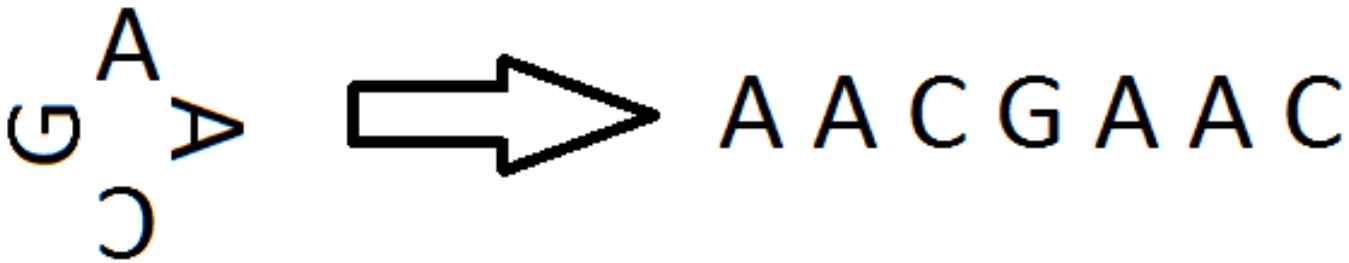
- Text file containing HELM notations of structures to be searched
- Loaded and kept in memory as list of strings
- Successive restriction of indices of notations which satisfy the search criteria based on searches and their combinations
- Generation of output

Lists

- List of HELM Strings denoting complex polymers
- List of Sets of peptide Sequences present
- List of Sets of nucleotide Sequences present
- SmilesList of SMILES notations of complex polymers
- SmilesSetList of SMILES notations of parts of whole structures

Sequence

- Object that holds peptide or nucleotide sequence as string
- Also holds boolean information whether the sequence is present as part of a backbone-cyclic structure
- This allows search to be carried out on a doubled sequence, e.g.



- Need to check length of match if this is done

SmilesList and SmilesSetList

- SmilesList is an object that holds a list of SMILES strings denoting the whole structures
- SmilesSetList is an object that holds a list of sets of SMILES strings denoting parts of the whole structure
- Both of these objects also hold warning flags which are raised if some SMILES notations are not generated

StructureQuery and SequenceQuery

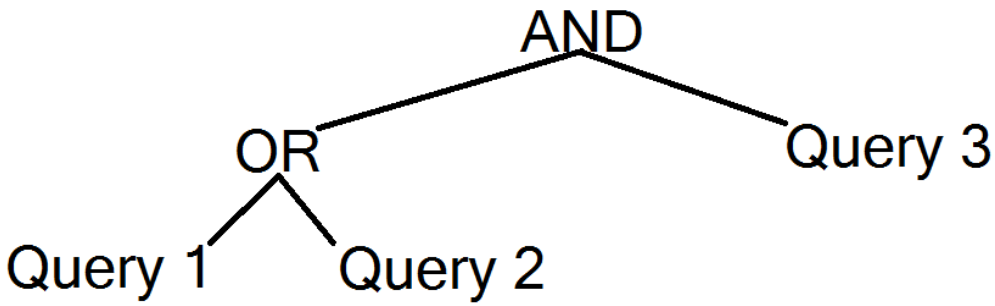
- Objects that hold query information
- queryString is the SMARTS notation/sequence of interest
- searchType denotes the type of search - currently only SUBMATCH search is supported
- cutoff which will apply to similarity search
- a boolean (negation) stores information about whether the content of this query is to be negated
- StructureQuery holds smilesLevel in addition to this
- SequenceQuery holds sequenceType in addition to this

Matches

- Object that holds indices of structures that match a query or combination of queries
- Also holds list of HELM notations based on which the search was carried out
- Also holds flags indicating possible errors in the results (e.g. not all SMILES strings necessary were actually generated)
- Methods for generating output data in different forms (indices/HELM strings)

Grouping

- Object that holds logic of query combinations
- Tree form with data at each node, in the form of logical Connectors, Queries, or Matches
- Recursively defined - each node except the root has a grouping as its parent
- List of groupings as children



Some important methods

- Constructors for Queries
- Constructors for Groupings
- CombinedSearch.performSearch(grouping, notationList)
- Various methods called by the above, including to optimise groupings, perform query-level searches using CombinedSearch.performSequenceSearch and CombinedSearch.performStructureSearch, as well as combine search results using CombinedSearch.combineSearches