Analysis of small molecule molecular data in R

Kevin Horan, Tyler Backman, Eddie Cao, Thomas Girke

Department of Botany and Plant Sciences
University of California, Riverside

July 26, 2014

Needs of Cheminformatics

Need to be able to search through large libraries of compounds
Requirements:

- A means of representing compound information
- An index to enable fast searching
- Algorithms to perform the search

Representing Compounds

- SDF
  - Stores a list of atoms and a connection table describing the connections between atoms
- SMILES
  - A line based format using parenthesis to represent branches in the compound structure
  - Example: CC(=O)Oc1ccccc1C(=O)O

SDF Exmaple

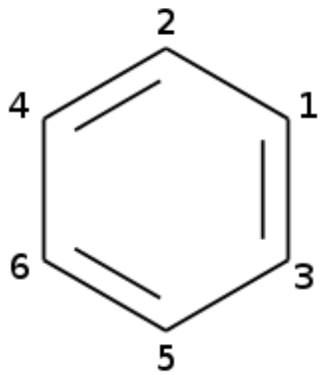
```
benzene
ACD/Labs0812062058

  6  6  0  0  0  0  0  0  0  0  1 V2000
    1.9050   -0.7932    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.9050   -2.1232    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.7531   -0.1282    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.7531   -2.7882    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.3987   -0.7932    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.3987   -2.1232    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  2  1  1  0  0  0  0
  3  1  2  0  0  0  0
  4  2  2  0  0  0  0
  5  3  1  0  0  0  0
  6  4  1  0  0  0  0
  6  5  2  0  0  0  0
M  END

>  <Unique_ID>
XCA3464366

>  <ClogP>
5.825

>  <Molecular Weight>
499.611
$$$$
```



Compound Formats in ChemmineR

- Reading/Writing:

```
>sdfset = read.SDFset("file.sdf")
>write.SDF(sdfset,file="output.sdf")
```

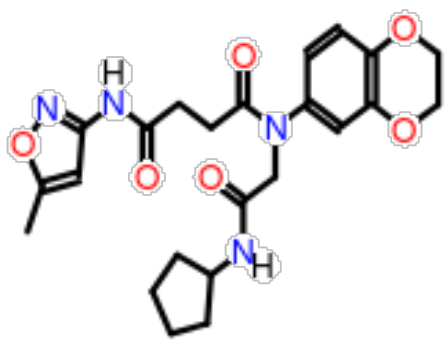
- Converting between formats (uses ChemmineOB):

```
>convertFormatFile("CML","SDF","file.cml","file.sdf")
```

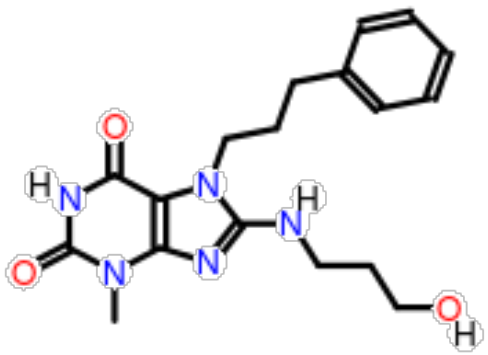
- Plotting compounds:

```
>plot(sdfset[1:2], print=FALSE)
```

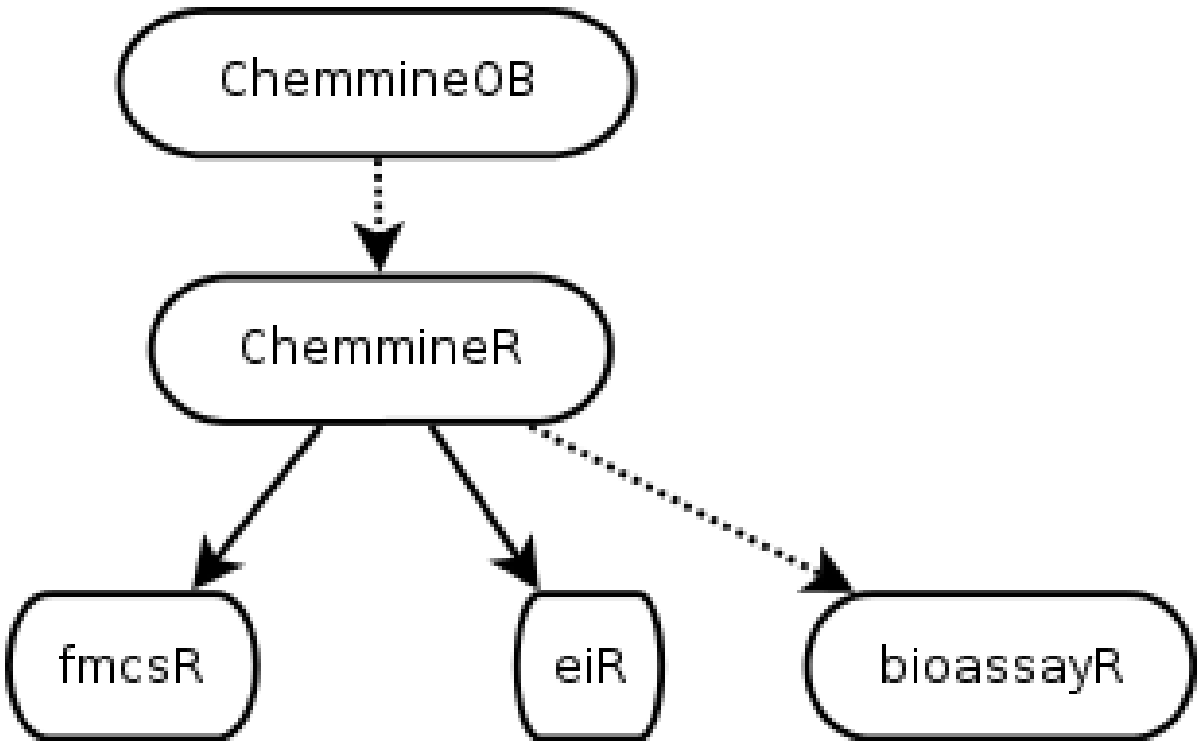
650001



650002



Tools in Bioconductor



Compound Similarity

Fundamental rule: similar compounds have similar properties

- Similarity can be defined in two ways:
  - Structural: looking at the atom connections and bond types
    - FMCS (Flexieble Maximum Common Substructure): A fast and fuzzy similarity score
  - Physical: looking at various properties of the compound, such as molecular weight
    - These properties are encoded into descriptors

Compound Descriptors

- Short descriptions of certain aspects of a compound
- Fingerprints
  - Stored as a bit string
  - Each bit represents the presence or absence of a single feature
  - For example, whether or not a benzene ring is present
  - Example: 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 . . . length: 1024
- Atom Pairs
  - A list of pairs of atoms and the shortest lengths between them
  - Example: 53822408832 53822408833 53822408834 53822408835 53822408836 . . . length: 15

Indexing Compounds

- Performing some up-front work to make search faster later
- Descriptors are small enough to be indexed
- Commonly done with a similarity function
- Examples: Euclidean, Tanimoto

Computing Descriptors

Compute Atom Pairs from an SDFSet object:

```
>apset = sdf2ap(sdfset)
```

Convert that to a fingerprint:

```
>fpset = desc2fp(apset)
> fpset[[1]]
An instance of "FP" of type "unknown-5682"
<<fingerprint>>
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ... length: 1024
```

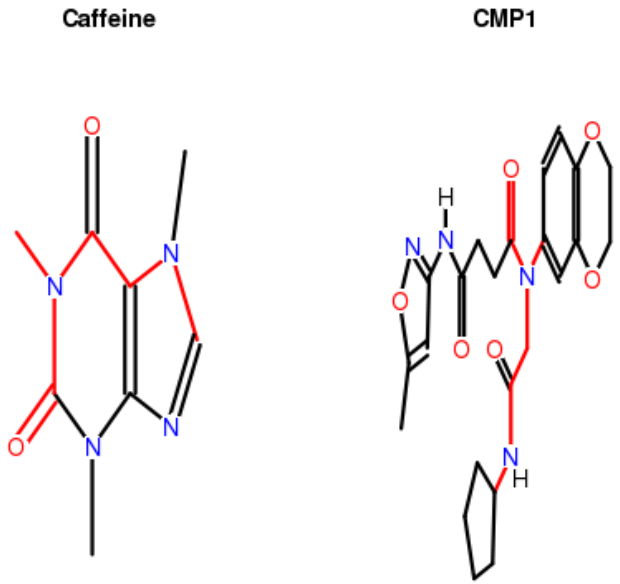
Compute Tanimoto similarity between first compound and all others:

```
>similarities = fpSim(fpset["650065"],fpset,
                      method="Tanimoto")
```

FMCS

A graph-based similarity method that is defined as the largest substructure shared among two compounds

- Very sensitive and accurate search method
- Allows bond/atom mismatches
- Implemented in the fmcsR package



Similarity Searching Methods

- Clustering: Compounds in the same cluster are similar to each other
- EI: A fast nearest neighbor based search method
- bioassayR: Find active compounds using screening data

Clustering

- Binning clustering
  - divide compounds into discrete similarity groups for a given set of cutoff values
- Jarvis-Patrick
  - An $O(n)$ algorithm using nearest neighbor data
- Distance matrix based methods
  - Export a distance matrix that can used with many other types of clustering algorithms supported in R
  - Example: hierarchical clustering with *hclust*
- Implemented in the ChemmineR package

Binning

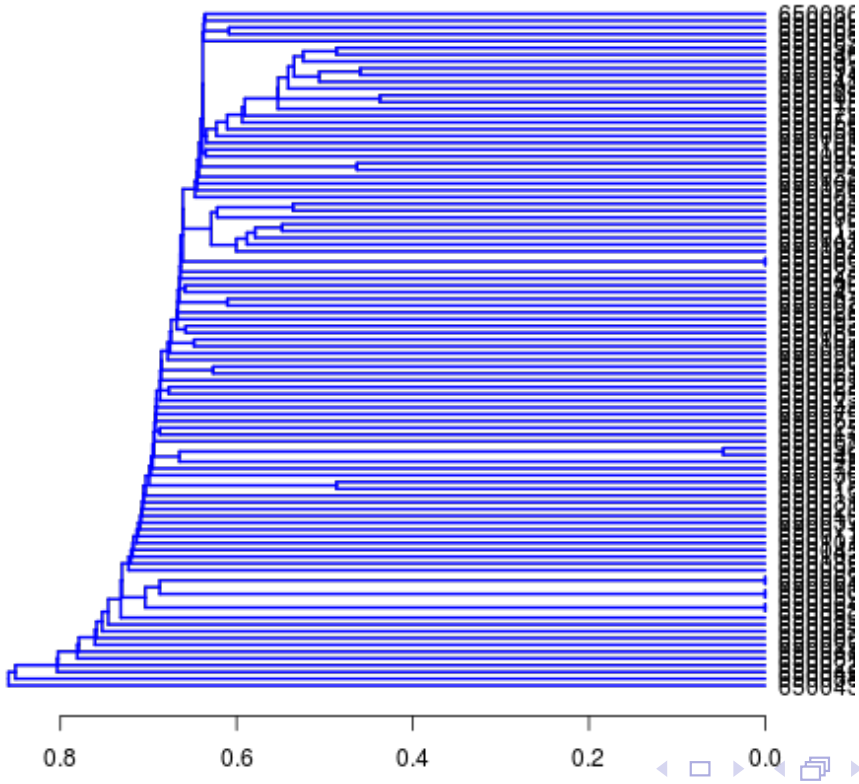
```
> clusters <- cmp.cluster(db=apset,
                          cutoff = c(0.7, 0.8, 0.9), quiet = TRUE)
> clusters[1:12,]
ids CLSZ_0.7 CLID_0.7 CLSZ_0.8 CLID_0.8 CLSZ_0.9 CLID_0.9
650049   2   48         2       48         2       48
650050   2   48         2       48         2       48
650059   2   54         2       54         2       54
650060   2   54         2       54         2       54
650061   2   56         2       56         2       56
650062   2   56         2       56         2       56
650063   2   58         2       58         2       58
```

Jarvis-Patrick

```
>cl = jarvisPatrick(nearestNeighbors(apset,cutoff=0.6),
                    k=2,mode="b")
> byCluster(cl)
$`48`
[1] "650049" "650050"
$`53`
[1] "650059" "650060"
$`54`
[1] "650061" "650062"
$`55`
[1] "650063" "650064"
$`56`
[1] "650065" "650066"
```

Hierarchical

```
>cmp.cluster(db=apset, cutoff=0,
    save.distances="distmat.rda", quiet=TRUE)
>load("distmat.rda")
>hc <- hclust(as.dist(distmat), method="single")
```



EI

- Uses precomputed compound descriptors, Atom Pair by default
- Select a small set of n exemplary compounds
- For each compound, computes the similarity to each of the n exemplars
- Embeds this n dimensional vector into k dimensional space using MDS
- Creates a nearest neighbor index using Locality Sensitive Hashing (LSH) which allows finding the nearest neighbor in near constant time
- User can then submit a query compound and find a set of similar compounds
- Query time does not scale with database size
- Implemented in the eiR package

EI Example

```
>library(eiR)
>data(sdfsample)

#Create the database
>eiInit(sdfsample[1:99])
>runId <- eiMakeDb(60,40)

#peform a query
>eiQuery(runId,sdfsample[45],K=10,asSimilarity=TRUE)

##    query target similarity target_ids
## 1 650046 650046     1.0000        245
## 2 650046 650011     0.4651        211
## 3 650046 650092     0.3923        286
## 4 650046 650004     0.1853        204
## 5 650046 650021     0.1383        220
```

Conclusion

- ChemmineOB
  - Provides access to the rich and fast set of functionality provided by Open Babel
- ChemmineR
  - A general cheminofrmatics framework
  - provides:
    - compound storage as SDF or SMILES objects
    - search algorithms
    - clustering algorithms
    - compound plotting
- eiR
  - Fast chemical search for libraries with millions of compounds
- fmcsR
  - High accuracy sub-structure based search with flexible pattern matching options
- bioassayR