# Updates – Jul 26

* Got a document of PubChem CID IDs and the respective SMILES code
  + Issue is that file is 8 GB long
* Preston has a dataset of some of the CID IDs, the SMILES code, molecular name, and InChI value
* Two ideas:
  + Run a query of the SMILES code into NCI/CADD database, get the molecular name
    - Issue is that SMILES is not normalized, so a PubChem SMILES might be different from NCI/CADD SMILES
      * Test with CID 2
  + Find the PubChem list of all its molecules

Useful Links:

<https://chemistry.stackexchange.com/questions/122109/how-to-get-the-smiles-of-all-compounds-on-pubchem>

<https://docs.google.com/spreadsheets/d/1guwx5QVjg9seOLe6l2LuCC7uFeBhK6J4ccCPM-VmQeg/edit#gid=1527918723>

<https://www.researchgate.net/post/How_to_download_the_SMILES_of_all_compounds_in_a_batch_from_PubChem>

# Notes – Jul 18

* Get data of compounds and SMILES code
* Change the code to use SMILES instead of InChI
  + Or talk to Sridhar about it
* Work on Sridhar’s representation
  + Aromatic bonds, order of symbols, rings, etc.

# Updates – Jul 18

* Built a program which takes in a molecule name and outputs an adjacency matrix or NetworkX graph
* Accesses the National Cancer Institute database for Standard InChI
  + SMILES is not canonized
  + We will build molecular file from InChI and then use RDKit to generate SMILES
* Converts InChI to .mol file, explicitly adding hydrogen atoms
* Builds adjacency matrix from .mol file by iterating through each atom
* Uses NetworkXz to build a graph from the adjacency matrix

# Notes – Jul 17

* Take molecular name as input
* Output the adjacency matrix of the molecule for Caple to use
  + Explicitly state hydrogens
  + Capture bond information

# Updates – Jul 17

* Use [] to denote rings instead of |
* Use {} to denote non-organic molecules
* “Sridhar’s” representation is easier to note structure since it explicitly states rings and bonds
  + Consider benzene vs cyclohexane
    - Both are almost the same in SMILES, just one is all uppercase
    - “Sridhar’s” differentiates them better and shows bond structure
  + Also note spiro[5.5]undecane
    - “Sridhar’s” shows the chemical structure better
* Syntelly is an AI tool for organic chemistry
  + Outputs SMILES for molecular compounds and diagrams
    - Can also output compound name for SMILES code
  + Also can take a PDF and can output the SMILES code for the molecular diagrams contained in the PDF

# Updates – Jul 14

## Databases

* Most databases just search for pre-stored SMILES
* Some generate SMILES

## ChemInfo

* ChemInfo is a tool which takes a molecule diagram and outputs the SMILES code for it
  + Specific tool is named ChemLib
* GitHub: [www.github.com/cheminfo](https://www.github.com/cheminfo)
  + <https://github.com/cheminfo/openchemlib-js/tree/817ec9789a604660641966936c3806dd00fc0d46/src/com/actelion/research/gwt>
* SMILES Generator folder: <https://github.com/cheminfo/openchemlib-js/tree/817ec9789a604660641966936c3806dd00fc0d46/src/com/actelion/research/gwt/chemlib/com/actelion/research/chem>
* Takes in a molecule and outputs a SMILES (or SMARTS) code by looking through each atom step by step, then looking at the neighbor atoms (for bonds, isomerism, and implicit hydrogen)

## Not Using Stereochemistry:

* Input: ExtendedMolecule

Comments from GitHub

\* While the Molecule class covers all primary molecule information as atom and bond properties, the atom connectivity and coordinates, its derived class ExtendedMolecule handles secondary, i.e. calculated molecule information. Most important are the directly connected atoms and bonds of every atom, information about rings and whether they are aromatic, and some atom propertiesthat depend on their near neighbors. This calculated information is cached in helper arrays and stays valid as long as the molecule's primary information is not changed. High level methods, e.g. getPath(), that require valid helper arrays take care of updating the cache themselves. Low level methods, e.g. isAromaticAtom(), which typically are called very often, do not check the validity of or update the helper arrays themselves for performance reasons. If you use low level methods, then you need to make sure that the required helper array information is valid by calling ensureHelperArrays(). Typically you will never instantiate an ExtendedMolecule, but rather use a StereoMolecule.

## Using Stereochemistry

* Input: StereoMolecule
  + Class extended from ExtendedMolecule

# Notes from Jul 8 Call

**Part 1:**

* SMILES (how its created, what are inputs)
* Word doc on how databases make SMILES from compound name
  + How database is populated
* Given chemical name in format and run it in program to generate SMILES
  + OR: does database have name and SMILES (so its dictionary instead of algorithm)
* Figure out how each one works

**Part 2:**

* Some drugs have properties
* Get thousands of the following: (drug name, properties, SMILES)
  + Make a huge database
* Goal is to eventually use machine learning

**Part 3:**

* Use RanDepict to get random chemical structures
* From there, hand-write the parenthesized representation for 5-6 of those
* Eventually, you need to create an algorithm
  + Input could be SMILES or chemical name or chemical structure

## 6/13 Update

* Built dgl heterogeneous graph based on drkg.tsv
* Built a function which creates a subgraph of two types of nodes
  + *Input*: nodeType1, nodeType2, triplets, directed (Boolean)
    - If directed == True, makes sure each edge leads from nodeType1 to nodeType2
    - If not, then order doesn’t matter
  + This can be used to build a bipartite graph
  + Also, can be used to build a homogeneous graph if edge type doesn’t matter
    - Can also draw the homogeneous graph
    - Keeps the ID/label of the original node
* Get a dictionary of the neighbors and the corresponding edge type of a node
  + *Input*: nodeID, triplets
  + Can be used to find the degree of a particular node

Ideas:

* Make graph homogeneous by assigning each node an ID, and then a dictionary contains each ID and its corresponding type (gene, disease, etc.)

## June 16th – June 19th

* Told to work on SMILES
* Worked on understanding the rules for SMILES
* SMILES formats DON’T have to be the same for the same compound if different conventions are used (see: canonical)