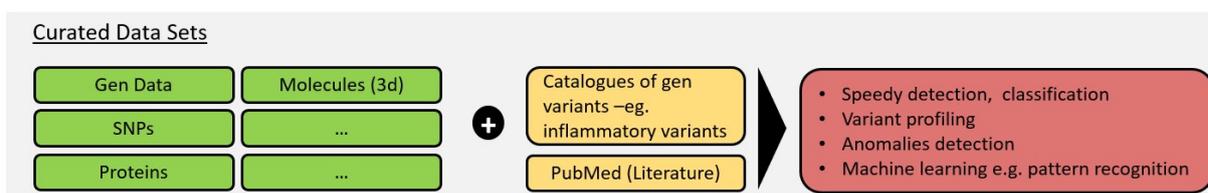
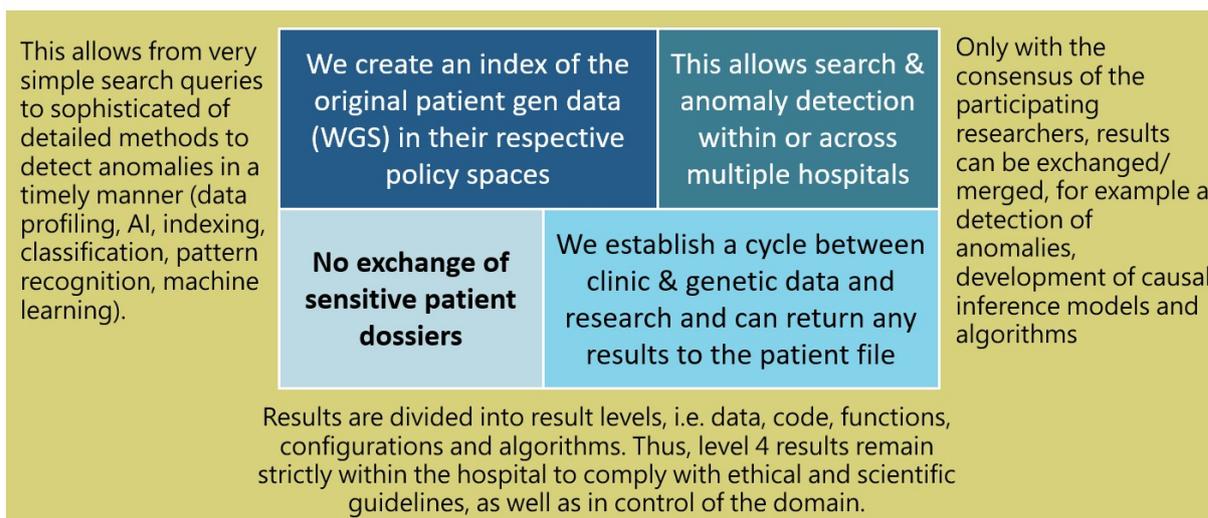


New area of bio-sequence computation

Biomolecular Sequences: Symbolic DNA, Proteins and much more

Iterata Health Platform supports several methods and powerful analytic tools for **bio-sequence computation**. Iterata enables geneticists and clinicians to identify and detect genetic anomalies without exchanging the sensitive patient gen data. Based on gene and protein data, novel abilities to do flexible, general computation with bio sequences in a way with many of the chemical computation capabilities and visualization



Life Sciences & Medicine: Data & Computation

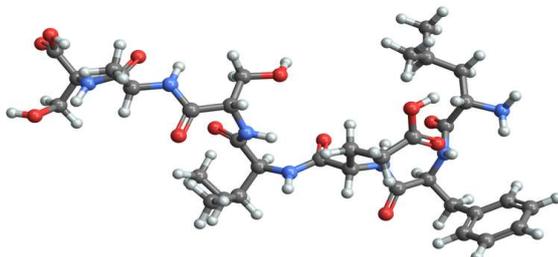
... immediate access to extensive life science data, as well as powerful tools for bioinformatics and biostatistics

Biomolecular Sequences using Wolfram Language & System

BioSequence is a string-based representation for biomolecules with chained primary structure. This class of biomolecules includes DNA, RNA, peptides and other sequences, which play important biological roles in maintaining genetic information and undertaking the work of the cell. This representation is supported by functions for recognition, comparison, transliteration and further operations. Degenerate letter handling is integrated throughout these operations. Interaction with the entity system allows for analyzing gene and protein sequences as well as customizing the underlying definitions of sequences and their behavior. BioSequence integrates with existing String functionality to enable novel biomolecular sequence processing.

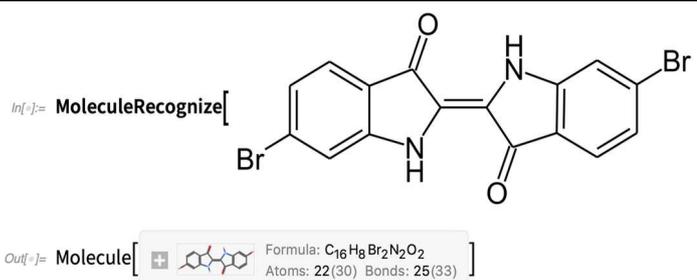
Molecular Structure & Computation

The Molecule is a symbolic representation of a chemical species and is a fully computable first-class member of the Wolfram Language. More than 20 new functions allow users to create, analyze and modify chemical species. A wide range of computed properties such as stereochemistry, symmetry elements, molecular graphs and molecular mechanics energies facilitate problem solving across all chemical disciplines. Built-in support for organic and inorganic nomenclature allows fast creation of molecule objects from chemical names. Molecule folds seamlessly into existing functionality, e.g. Graph, to enable novel computations.

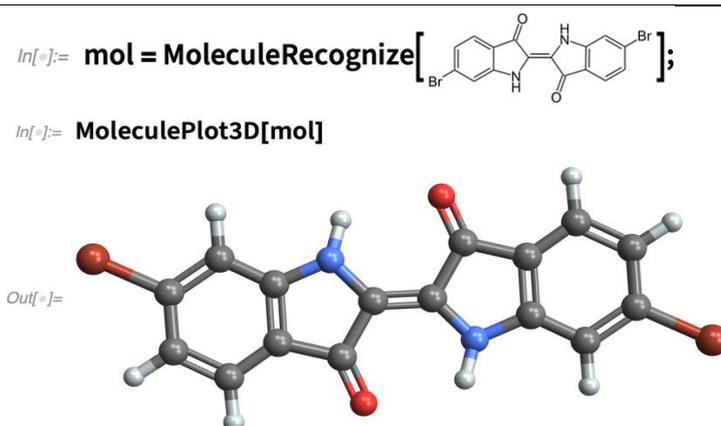
Representations of DNA sequences	
How to represent a DNA sequence	<p>In[]:= <code>BioSequence["DNA", "CTTTTCGAGATCTCGGCGTCA"]</code></p> <p>Out[]:= <code>BioSequence</code> [ Type: DNA Sequence Content: CTTTTC—TCA (21 letters)]</p>
Translates the sequence to a peptide	<p>In[]:= <code>BioSequenceTranslate[%]</code></p> <p>Out[]:= <code>BioSequence</code> [ Type: Peptide Sequence Content: LFEISAS (7 letters)]</p>
Find out what the corresponding molecule is	<p>In[]:= <code>Molecule[%]</code></p> <p>Out[]:= <code>Molecule</code> [ Formula: C₃₅H₅₅N₇O₁₂ Atoms: 109 Bonds: 109]</p>
Visualize it in 3D	<p>In[]:= <code>MoleculePlot3D[%]</code></p> <p>Out[]:= </p>
Deal with other branches of life too	<p>In[]:= <code>ascidian mitochondrial GENETIC TRANSLATION TABLE ["StartCodons"]</code></p> <p>Out[]:= {ATA, ATG, GTG, TTG}</p>
Work with bio sequences	<p>In[]:= <code>StringReverse[BioSequence["DNA", "CTTTTCGAGATCTCGGCGTCA"]]</code></p> <p>Out[]:= <code>BioSequence</code> [ Type: DNA Sequence Content: ACTGCG—TTC (21 letters)]</p>
finds a complementary base-pair sequence:	<p>In[]:= <code>BioSequenceComplement[BioSequence["DNA", "CTTTTCGAGATCTCGGCGTCA"]]</code></p> <p>Out[]:= <code>BioSequence</code> [ Type: DNA Sequence Content: GAAAAG—AGT (21 letters)]</p>
String patterns also understand things like this for bio sequences	<p>In[]:= <code>StringMatchQ[BioSequence["DNA", "CTTT"], "STTT"]</code></p> <p>Out[]:= True</p>
New functions like <code>BioSequenceInstances</code> for resolving degenerate characters	<p>In[]:= <code>BioSequenceInstances[BioSequence["DNA", "STTT"]]</code></p> <p>Out[]:= { <code>BioSequence</code> [ Type: DNA Sequence Content: CTTT (4 letters)], <code>BioSequence</code> [ Type: DNA Sequence Content: GTTT (4 letters)] }</p>
Completely integrated <code>BioSequence</code> with built-in genome and protein data e.g. gene that we can ask for in natural language	<p>In[]:= <code>BioSequence[hba1 gene]</code></p> <p>Out[]:= <code>BioSequence</code> [ Type: DNA Sequence Content: ACTCTT—GGC (842 letters)]</p>
Sequence alignment between e.g. two genes	<p>In[]:= <code>SequenceAlignment[BioSequence[hemoglobin, alpha 1 GENE ...], BioSequence[hemoglobin, alpha 2 GENE ...]]</code></p> <p>Out[]:= { { CATAAACCTGGCGGCTCGGGCCGGC, ACTCTTCTGGTCCCCACAGACTCAGAGAGAACCACCATGGTGCTCTCTGCCACAAGACCAA }</p>
FASTA files and start exploring it	<p>In[]:= <code>BioSequence["DNA", First[Import["Genome/Consensus/c1.fa.consensus.fa"]]]</code></p> <p>Out[]:= <code>BioSequence</code> [ Type: DNA Sequence Content: NNNNNN—NNN (247249719 letters)]</p>

What Is That Molecule? Advances in Chemical Computation

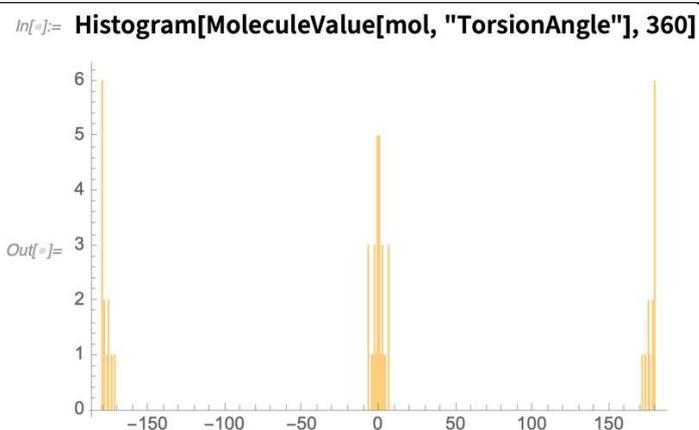
Given image of a molecular structure diagram. How can you get the molecule it represents in a computable form? use `MoleculeRecognize`



It's the analog of `TextRecognize`, but for molecules (symbolic representation of the molecule). Then generate a 3D structure



Compute the distribution of torsion angles of the structure

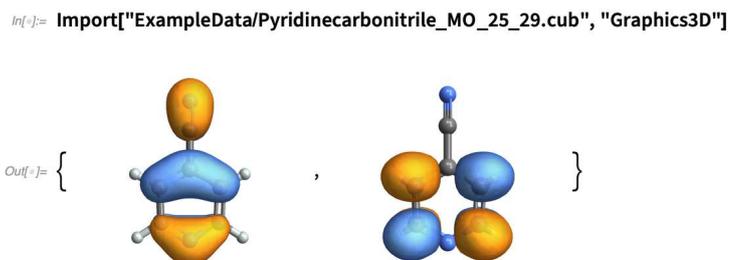


Connect to the world of external identifiers

`In[]:= MoleculeValue[mol, "PubChemCompoundID"]`

`Out[]:= { PubChem compound 177057 }`

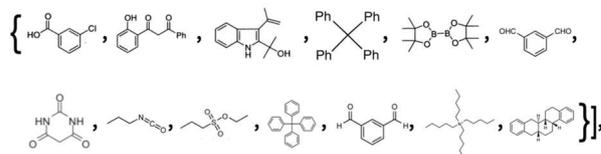
Ability to import data about molecular orbitals



Programmatically use with MoleculeRecognize

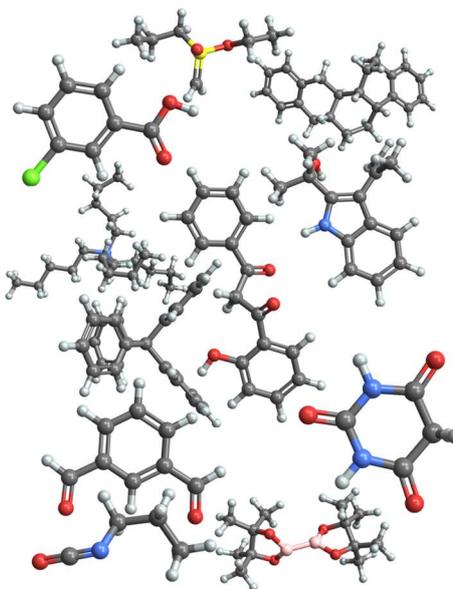
Take all the images of chemicals from a paper, "molecule OCR" them—then do things like check whether the molecules are equivalent, or make a word cloud of their 3D structures

`In[]:= WordCloud[MoleculePlot3D/@DeleteDuplicates[MoleculeRecognize[`



`MoleculeEquivalentQ]]`

`Out[]:=`



Please do not hesitate to contact us

Sincerely yours, Iterata Team

Phone +41 62 842 88 27 | info@iterata.ch