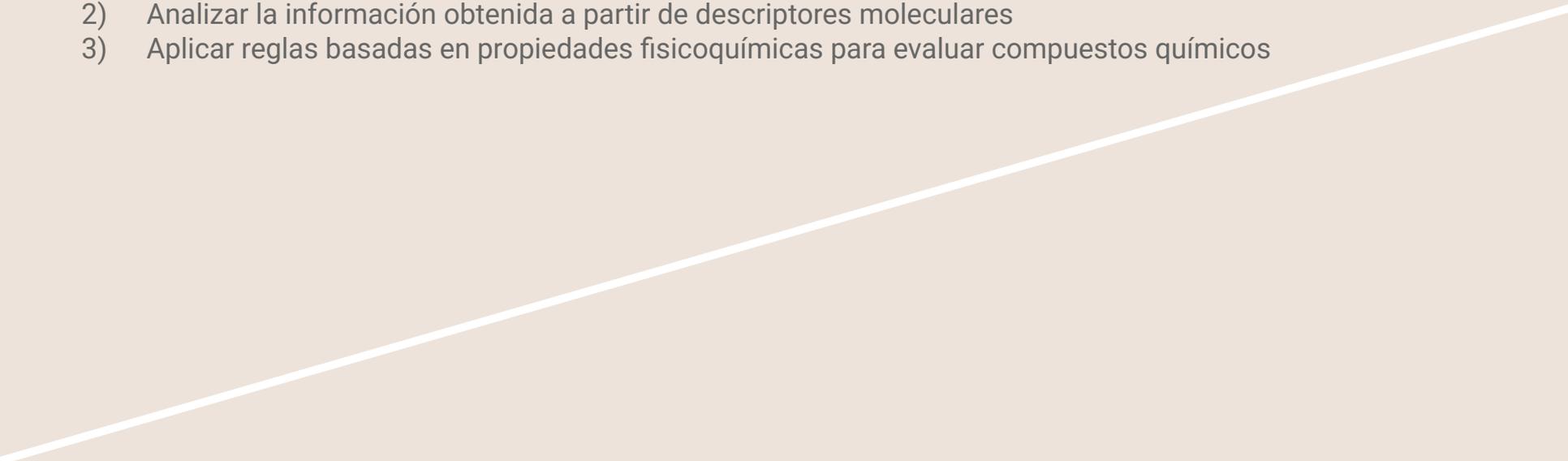


TP3

Caracterización fisicoquímica



Objetivos

- 1) Calcular propiedades fisicoquímicas de moléculas
 - 2) Analizar la información obtenida a partir de descriptores moleculares
 - 3) Aplicar reglas basadas en propiedades fisicoquímicas para evaluar compuestos químicos
- 

Organización de la clase

9:00 a 9:30	Introducción al TP
9:30 a 10:30	Trabajo en la guía de ejercicios (Parte 1: Cálculo de propiedades fisicoquímicas de una molécula)
10:30 a 11:00	Recreo
11:00 a 12:00	Trabajo en la guía de ejercicios (Parte 2: Análisis de conjuntos de datos)
12:00 a 13:00	Lectura de paper y puesta en común

Parte 1

Cálculo de propiedades fisicoquímicas de una molécula

Descriptores

```
# Calcular el peso molecular exacto de la molécula
molecular_weight = Descriptors.ExactMolWt(molecula)

# Calcular el logP (coeficiente de partición octanol-agua) de la molécula
logp = Descriptors.MolLogP(molecula)

# Calcular el número de donadores de enlaces de hidrógeno en la molécula
h_bond_donor = Descriptors.NumHDonors(molecula)

# Calcular el número de aceptores de enlaces de hidrógeno en la molécula
h_bond_acceptors = Descriptors.NumHAcceptors(molecula)

# Calcular el número de enlaces rotativos en la molécula
rotatable_bonds = Descriptors.NumRotatableBonds(molecula)

# Obtener el número total de átomos en la molécula
number_of_atoms = Chem.rdchem.Mol.GetNumAtoms(molecula)

# Calcular la refractividad molar de la molécula
molar_refractivity = Chem.Crippen.MolMR(molecula)

# Obtener el área superficial topológica mapeada de la molécula
topological_surface_area_mapping = Chem.QED.properties(molecula).PSA

# Obtener la carga formal de la molécula
formal_charge = Chem.rdmolops.GetFormalCharge(molecula)
```

Descriptores

```
def drug_likness_descriptors(df):  
    # Desactivar las advertencias de asignación encadenada en pandas  
    pd.options.mode.chained_assignment = None  
  
    # Crear listas vacías para almacenar los descriptores  
    NumHDonors_list = []  
    NumHAcceptors_list = []  
    MW_list = []  
    LogP_list = []  
    rotatable_bonds_list = []  
  
    # Calcular los descriptores para cada molécula en la columna 'ROMol'  
    for element in df['ROMol']:  
        try:  
            # Calcular el número de donadores de enlaces de hidrógeno  
            NumHDonors = Descriptors.NumHDonors(element)  
            NumHDonors_list.append(NumHDonors)  
        except:  
            NumHDonors_list.append('N/A') # Si ocurre una excepción, agregar 'N/A' a la lista  
            pass
```

```
# Agregar las listas de descriptores al DataFrame  
df['HBD'] = NumHDonors_list  
df['HBA'] = NumHAcceptors_list  
df['MW'] = MW_list  
df['logP'] = LogP_list  
df['nRotB'] = rotatable_bonds_list  
  
return df
```

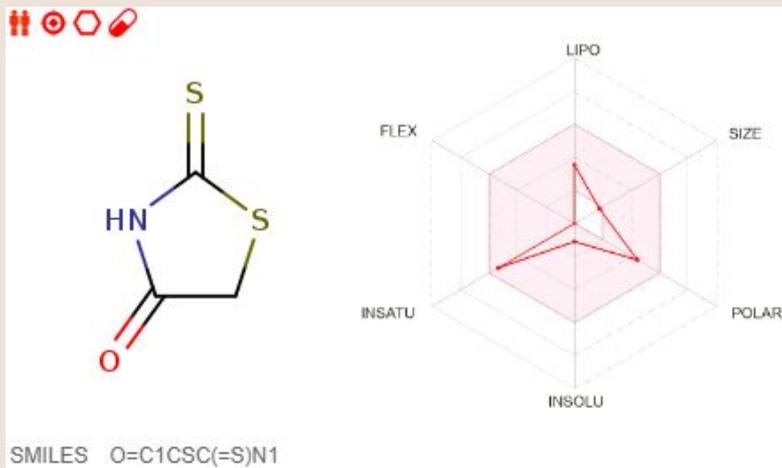
Descriptores

List of Available Descriptors

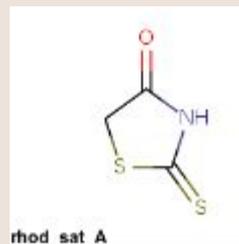
Descriptor/Descriptor Family	Notes	Language
Gasteiger/Marsili Partial Charges	<i>Tetrahedron</i> 36 :3219-28 (1980)	C++
BalabanJ	<i>Chem. Phys. Lett.</i> 89 :399-404 (1982)	Python
BertzCT	<i>J. Am. Chem. Soc.</i> 103 :3599-601 (1981)	Python
Ipc	<i>J. Chem. Phys.</i> 67 :4517-33 (1977)	Python
HallKierAlpha	<i>Rev. Comput. Chem.</i> 2 :367-422 (1991)	C++
Kappa1 - Kappa3	<i>Rev. Comput. Chem.</i> 2 :367-422 (1991)	C++
Phi	New in 2021.03 release <i>Quant. Struct.-Act. Rel.</i> 8 :221-224 (1989)	C++
Chi0, Chi1	<i>Rev. Comput. Chem.</i> 2 :367-422 (1991)	Python
Chi0n - Chi4n	<i>Rev. Comput. Chem.</i> 2 :367-422 (1991)	C++
Chi0v - Chi4v	<i>Rev. Comput. Chem.</i> 2 :367-422 (1991)	C++
MolLogP	Wildman and Crippen <i>JCICS</i> 39 :868-73 (1999)	C++
MolMR	Wildman and Crippen <i>JCICS</i> 39 :868-73 (1999)	C++
MolWt		C++
ExactMolWt		C++
HeavyAtomCount		C++
HeavyAtomMolWt		C++
NHOHCount		C++
NOCCount		C++
NumHAcceptors		C++
NumHDonors		C++

<https://www.rdkit.org/docs/GettingStartedInPython.html#list-of-available-descriptors>

PAINS | Brenk



PAINS



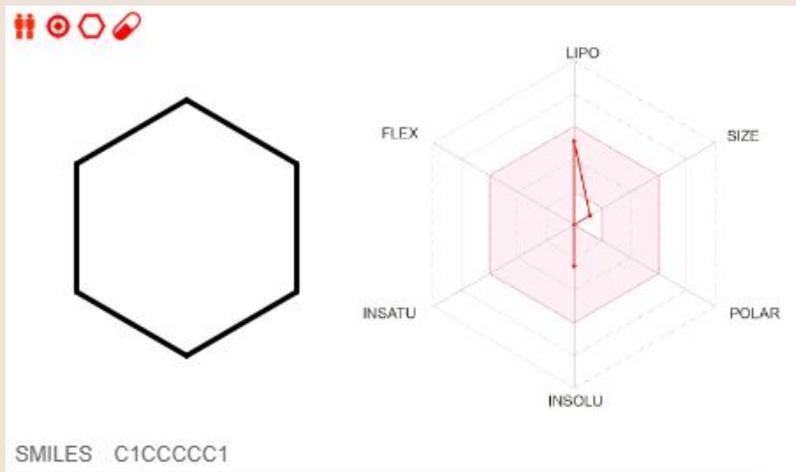
True

Brenk



True

PAINS | Brenk



PAINS



False

Brenk



False

PAINS | Brenk

	PAINS	Brenk
True	Tiene grupos promiscuos que pueden presentar actividad contra varios targets	Tiene al menos un grupo posiblemente reactivo o tóxico
False	No tiene grupos promiscuos que pueden presentar actividad contra varios targets	No tiene grupos posiblemente reactivo o tóxico

SwissADME

[Open Access](#) | [Published: 03 March 2017](#)

SwissADME: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules

[Antoine Daina](#), [Olivier Michielin](#)  & [Vincent Zoete](#) 

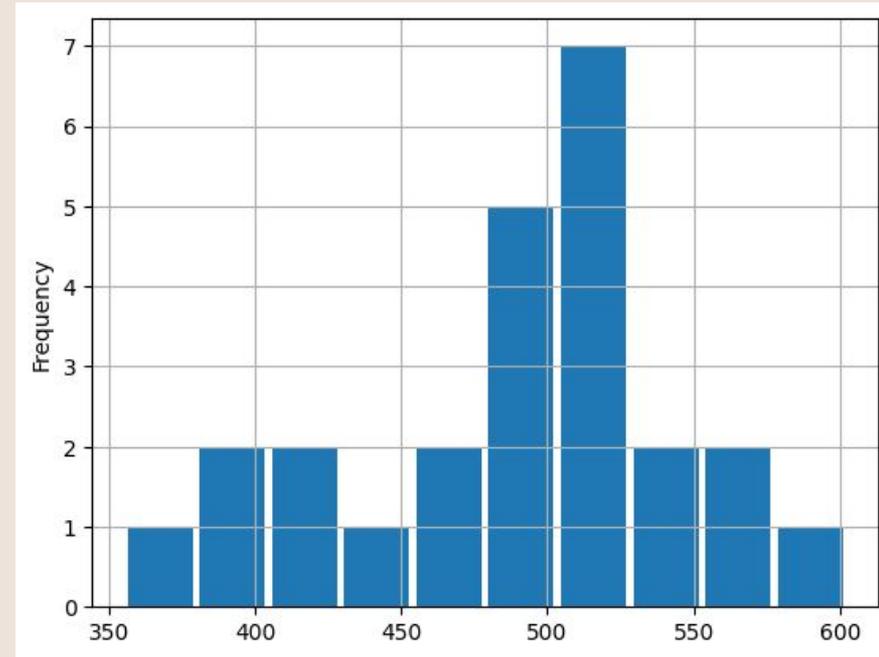
Daina, A., Michielin, O. & Zoete, V. SwissADME: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. *Sci Rep* 7, 42717 (2017).

<https://doi.org/10.1038/srep42717>

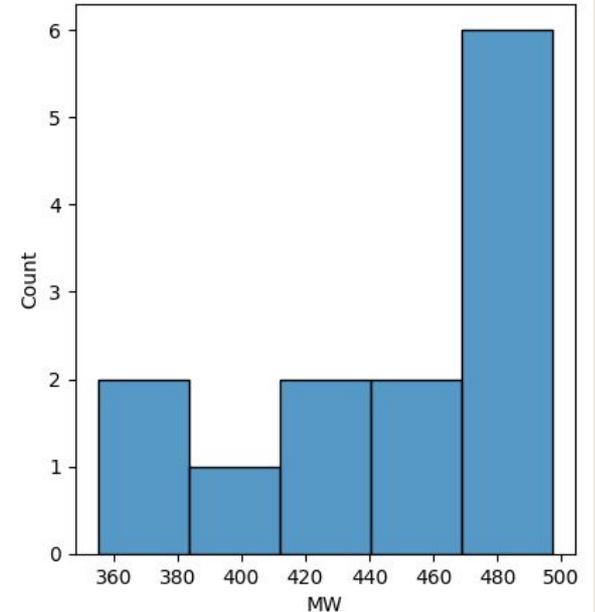
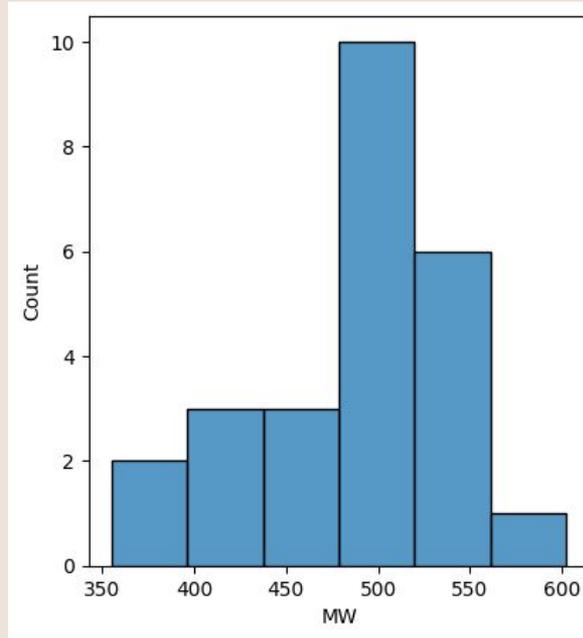
Parte 2

Cálculo de propiedades fisicoquímicas de una lista de moléculas

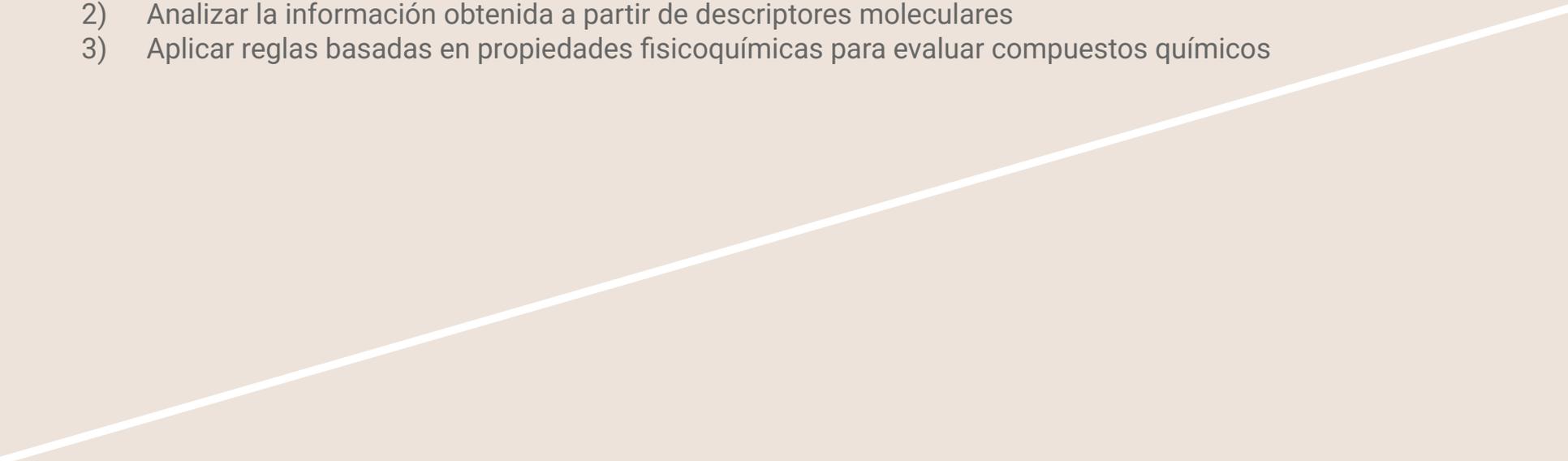
Visualización de la distribución de las propiedades fisicoquímicas



Visualización de la distribución de las propiedades fisicoquímicas



Objetivos

- 1) Calcular propiedades fisicoquímicas de moléculas
 - 2) Analizar la información obtenida a partir de descriptores moleculares
 - 3) Aplicar reglas basadas en propiedades fisicoquímicas para evaluar compuestos químicos
- 

Cierre

Chemprop: A Machine Learning Package for Chemical Property Prediction

Esther Heid, Kevin P. Greenman, Yunsie Chung, Shih-Cheng Li, David E. Graff, Florence H. Vermeire, Haoyang Wu, William H. Green, and Charles J. McGill*



Cite This: *J. Chem. Inf. Model.* 2024, 64, 9–17



Read Online

<https://pubs.acs.org/doi/10.1021/acs.jcim.3c01250?ref=pdf>

<https://github.com/chemprop/chemprop>