Loewdin Population Analysis Script

Overview

This Python script processes and visualizes data from the Loewdin population analysis section of an ORCA quantum chemistry output file. The tool allows users to specify atom types, molecular orbitals (MOs), and orbitals for analysis, as well as create visualizations of contributions to MOs. See the SI of (THIS) reference to set up the ORCA input file. The UNO keyword is required

---

Features

- Parses Loewdin population data from ORCA output files.
- Supports filtering by atom types, orbitals, and specific atom numbers.
- Handles pagination in ORCA output files.
- Summarizes contributions to MOs for visualization.
- Creates line plots to visualize contributions by atom types or individual atoms.

Requirements

- ORCA quantum chemistry software package for generating the output file
- Python 3.x
- Required libraries: `pandas`, `matplotlib`

Install the required libraries via pip:

```
Python
pip install pandas matplotlib
```

Usage

Input Setup

Inputs are defined in the script as dictionaries and lists example provided below:

```python
atoms_to_analyze = {
    "Fe": {"orbitals": ["d"], "atoms": [67]},  # Analyze only Fe atom 67
    "O": {"orbitals": ["s", "p"], "atoms": []},  # Analyze all O atoms
    "N": {"orbitals": ["s", "p"], "atoms": [66, 73]},  # Analyze N atoms 66 and 73
    "P": {"orbitals": ["s", "p"], "atoms": []},  # Analyze all P atoms
}

mos_to_analyze = [518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531]
```

- **`atoms_to_analyze`**: Specifies atom types, orbitals, and optional atom numbers for analysis.
- **`mos_to_analyze`**: List of MO indices to analyze.

Running the Script

1. Ensure the ORCA output file is placed in the specified directory.

Update the path to your ORCA output file in the script:

```python
output_file_path = r"path_to_your_orca_output_file"
```

2. Run the script in VS Code or your editor of choice

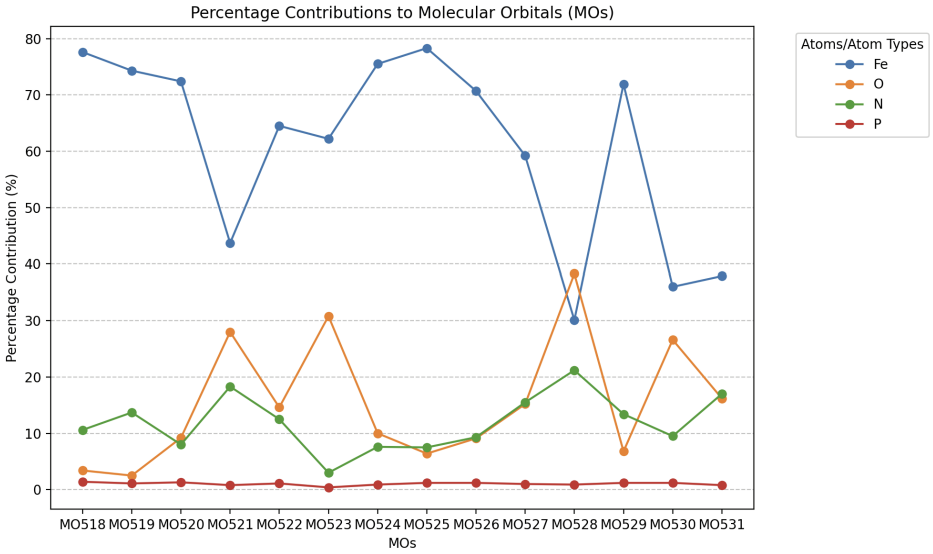
Output

- **Summarized Data**: Contributions by atom types and MOs is printed in the terminal for visualization.
- **Visualization**: Line plot of contributions. Individual atom contributions are shown only if specific atoms are specified.

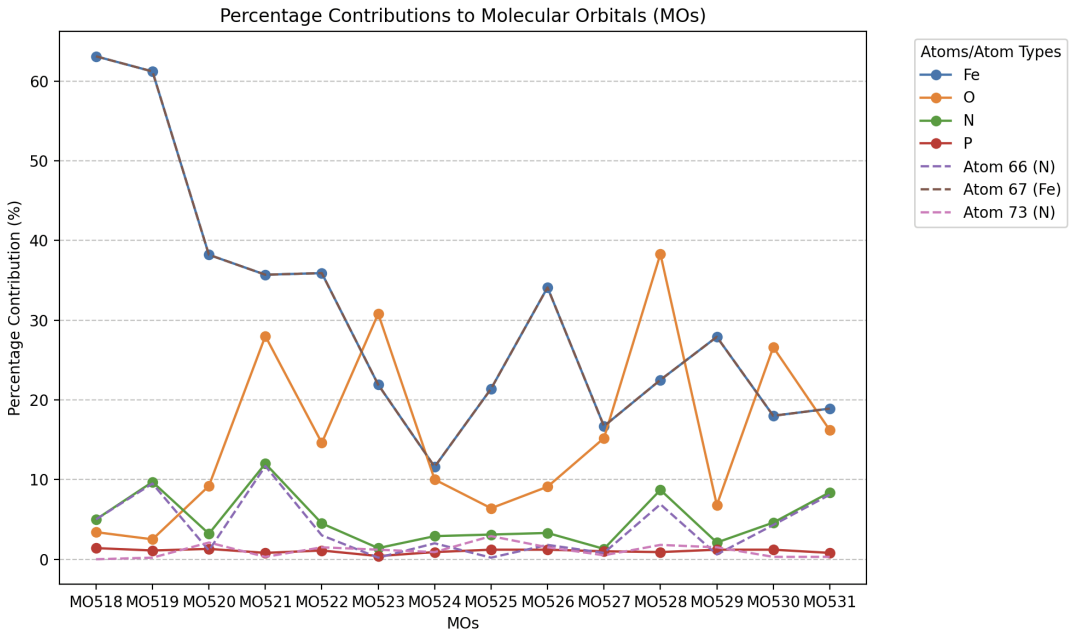
Example Visualization

The script generates a line plot showing the percentage contributions to specified molecular orbitals (MOs) by atom types or specific atoms.

- **Overall Contributions**: Aggregated by atom type.
- **Individual Contributions**: Shown when specific atoms are specified.



If specific atoms are specified those will be indicated by dashed lines.



Functions

1. `parse_loewdin_population(output_lines, mo_filter)`

Parses the Loewdin population section, handling pagination and filtering by specified MOs.

2. `write_loewdin_parsed_to_file(loewdin_parsed, output_filename)`

Writes parsed Loewdin data to a text file for debugging.

3. `process_orca_output(file_path, atom_filter, mo_filter, orbital_filter_map)`

Processes ORCA output to sum contributions for specified atoms and orbitals across MOs.

Customization

To analyze different atom types, orbitals, or molecular orbitals:

1. Update the `atoms_to_analyze` and `mos_to_analyze` variables.

Troubleshooting

- Ensure the ORCA output file exists in the specified path.
- Verify the required libraries are installed.
- Use the debug output file (`parsing_debug.txt`) to review parsed data.
