**JANPA Software:** NBO 라이선스 없이 NPA(Natural Population Analysis) 분석을 가능하게 하는 오픈소스 도구

2025.04.02 안용상

NPA(Natural Population Analysis)는 분자의 전자 분포를 해석하고, 원자 단위의 전하(atomic charges)를 평가하는 데 사용되는 기법이다. 이 방법은 orca와 같은 양자화학 프로그램을 활용한 계산에서 얻은 전자밀도 행렬로부터 자연 원자 궤도함수(Natural Atomic Orbitals, NAOs)를 구성한 후, 이를 기반으로 각 원자에 전하를 할당하는 방식이다.

이러한 NPA는 일반적으로 NBO(Natural Bond Orbital) 프로그램을 통해 수행된다. NBO는 전자 구조를 화학적으로 직관적인 결합 구조로 해석할 수 있도록 해 주는 분석 도구이며, NPA는 NBO 분석의 첫 단계에 해당한다. 그러나 NBO 프로그램은 상용 라이선스를 필요로 하므로 접근이 제한되는 경우가 많다.

이러한 문제를 해결하기 위해 개발된 소프트웨어가 바로 JANPA (Java-based Natural Population Analysis)이다.

JANPA는 Java 기반의 오픈소스 소프트웨어로, NBO 라이선스 없이도 NPA 분석을 수행할 수 있는 대안이다. JANPA는 다음과 같은 특징을 가진다:

1. 라이선스 프리: 상용 NBO 프로그램 없이 NPA 계산이 가능하다.
2. 오픈소스: 코드가 공개되어 있어 신뢰성과 확장성이 우수하다.
3. 범용성: ORCA, Gaussian, PSI4 등 주요 양자화학 소프트웨어와 호환된다.
4. 플랫폼 독립성: Windows, Linux, macOS 등 다양한 환경에서 사용할 수 있다.

JANPA는 분자의 전자 구조 해석, 전하 분포 분석, 결합 특성 평가 등에 유용하며, 전통적인 NBO 프로그램의 대체재로 기능한다. 특히 교육, 아카데믹 연구, 오픈 연구 환경에서 라이선스 제약 없이 전하 분석을 수행하고자 할 때 적합한 도구이다. 다운로드는 아래 링크에서 할 수 있다.

JANPA 소프트웨어 다운로드 링크<https://janpa.sourceforge.net/>

**JANPA 소프트웨어의 구성**

현재 JANPA 프로그램의 최신 버전 : 2.01 ver (새롭게 개발된 Localized Property-Optimized Orbitals (LPOs) 및 'The Chemist's LPOs' (CLPOs)와 같은 오비탈 지역화 방법이 구현되어 있다)

1. janpa.jar

MOLDEN 호환 포맷의 파일을 입력으로 받아, 자연 원자 궤도함수(NAO) 생성 및 \*\*Natural Population Analysis(NPA)\*\*를 수행하는 핵심 분석 프로그램이다.

1. molden2molden.jar

ORCA, Gaussian, PSI4 등 주요 양자화학 프로그램들이 생성하는 MOLDEN-세미호환(semi-compatible) 형식의 출력을, JANPA가 인식할 수 있는 확장된 MOLDEN 포맷으로 변환하는 도구이다.

1. nwchem2molden.jar

NwChem 프로그램의 계산 결과로부터 MOLDEN 호환 파일을 생성하는 변환기이다

**JANPA 소프트웨어로 Orca의 Output파일(.gbw)로부터 NPA 도출하는 방법**

https://sourceforge.net/p/janpa/wiki/OrcaExamples/

ORCA 프로그램은 .gbw 또는 .47 확장자의 출력 파일을 가지는데, 이를 활용하여 JANPA로 NPA를 수행하는 방법을 설명하겠다. 이 절차는 ORCA v3.x 버전에서 테스트되었으며, SCF 방법(HF, DFT), MP2, CCSD, QCISD 등 다양한 전자 구조 이론 수준에서 활용 가능하다.

**필요 파일 및 jar**

* RCA 계산 완료 후 생성되는 파일들:

1. \*.gbw: Molecular orbital 계수 정보 포함
2. \*.mdcip.47: 밀도 행렬(density matrix) 정보 포함 (post-SCF methods에서 필요)

* JANPA 패키지 구성 요소:

1. orca\_2mkl: .gbw 파일을 MOLDEN-semi-compatible 파일로 변환
2. molden2molden.jar: ORCA 스타일 MOLDEN 파일 → 표준 MOLDEN 파일로 변환
3. janpa.jar: NPA 분석 수행

**NPA 분석 절차**

1. Step 1. .gbw → MOLDEN-semi-compatible 파일 생성

orca\_2mkl myfile -molden

1. Step 2. MOLDEN 파일 정제 및 density matrix 반영
   1. 기본 변환 (SCF 방법)

java -jar molden2molden.jar -i myfile.molden.input -o myfile.PURE -fromorca3bf -orca3signs

* 1. post-SCF (MP2, QCISD, CCSD 등)

java -jar molden2molden.jar -i myfile.molden.input -o myfile.PURE -fromorca3bf -orca3signs -ds47 myfile.mdcip.47

1. Step 3. JANPA 분석 수행

java -jar janpa.jar -i myfile.PURE > myfile.JANPA

**NPA 분석 절차 자동화 (.sh 파일)**

아래와 같은 .sh 파일을 만들어 실행함으로써 위의 절차를 자동화할 수 있다.

janpa\_npa.sh

#!/bin/bash

run\_one() {

  INPUT="$1"

  echo "=============================="

  echo "📁 처리 중: $INPUT"

  # Step 1: ORCA 출력 → Molden 변환

  echo "Step 1: orca\_2mkl 실행 중..."

  orca\_2mkl "$INPUT" -molden

  # Step 2: molden2molden 변환

  echo "Step 2: molden2molden 실행 중..."

  java -jar "$JANPA/molden2molden.jar" -i "$INPUT.molden.input" -o "${INPUT}\_janpa.molden" -fromorca3bf -orca3signs

  # Step 3: JANPA 실행

  echo "Step 3: JANPA 실행 중..."

  java -jar "$JANPA/janpa.jar" -i "${INPUT}\_janpa.molden" > "${INPUT}.janpa.output"

  echo "## ${INPUT}.janpa.output 생성 완료!"

}

# -all 옵션이면 모든 .gbw 파일 처리

if [ "$1" = "-all" ]; then

  echo "📂 현재 디렉토리 내 모든 .gbw 파일 처리 시작..."

  for file in \*.gbw; do

    # 확장자 제거

    base="${file%.gbw}"

    run\_one "$base"

  done

  echo "🎉 전체 작업 완료!"

  exit 0

fi

# 단일 파일 처리

if [ -z "$1" ]; then

  echo "❗ 사용법: ./run\_janpa.sh input\_filename"

  echo "   또는:  ./run\_janpa.sh -all"

  exit 1

fi

run\_one "$1"