SOUPY

Author: Néstor Rovira Yélamos
CSIC IDAEA, Carrer Jordi Girona 29, 08034 Barcelona, Spain.

I. INTRODUCTION

Groundwater systems often present a complex mixture of different recharge sources. Identifying the relative contributions of these sources is fundamental in hydrogeology, as it enables a better understanding of aquifer dynamics, the origin of contaminants, and the processes controlling water quality. Quantifying mixing ratios is therefore a central task for groundwater management.

In this context, we present Soupy, a Python program specifically designed to solve the mixing ratio problem in groundwater samples. The software implements mathematical methods to analyze chemical and isotopic data and provides estimates of source contributions.

This article introduces the conceptual basis of Soupy.

II. MODEL

The mixing problem can be formulated as a matrix factorization task, where the source concentration matrix $W \in \mathbb{R}^{m\times k}$ is multiplied by the mixing ratio matrix $H \in \mathbb{R}^{k\times n}$ to approximate the observed concentration matrix $X \in \mathbb{R}^{m\times n}$:

$$X \approx WH.$$

Here, m represents the number of chemical or isotopic tracers, k the number of potential sources, and n the number of groundwater samples. The goal is to estimate the unknown matrix H, subject to physical constraints:

- $0 \leq H_{ij} \leq 1$ for all i, j, ensuring that mixing ratios are bounded between 0 and 1.
- $\sum_i H_{ij} = 1$ for all j, so that each sample is represented as a linear combination of the sources.

Because both the observed concentrations X and the source signatures W are affected by measurement uncertainty and natural variability, the problem cannot be solved analytically. Instead, it is formulated as an optimization problem that seeks matrices W and H minimizing the mismatch between observed and reconstructed concentrations.

A straightforward objective function is the Frobenius norm of the residuals:

$$\phi = \sum_{ij}((WH - X)_{ij})^2,$$

which corresponds to a least-squares fit. However, this formulation tends to emphasize species with higher absolute concentrations, potentially introducing significant errors in less abundant tracers.

To overcome this limitation, a weighted objective function is adopted, where residuals are scaled by their associated standard deviations:

$$\phi = \sum_{ij} \frac{((WH - X)_{ij})^2}{\sigma_{ij}^2}$$

This approach is equivalent to generalized least squares, and ensures that all tracers, regardless of magnitude, contribute equally to the optimization according to their analytical uncertainty. After this change there is a remaining problem, minimizing this function doesn't ensure that the final values of W are compatible with the measurements. Therefore, we can add another term to the function to take into account the measurements of the sources.

$$\phi = \sum_{ij} \frac{((WH - X)_{ij})^2}{\sigma_{X,ij}^2} + \lambda_W \sum_{ij} \frac{((W - W_{measure})_{ij})^2}{\sigma_{W,ij}^2}$$

Where λ_W is a weight parameter that can be adjusted by the user.

An advanced feature in the program is the variable p that allows the user to modify the exponent in the function. By default, the objective function uses $p = 2$, corresponding to a least-squares approach that is optimal under the assumption of normally distributed errors. However, the program also allows the exponent p to be set by the user. Values of $p < 2$ (e.g., $p = 1$) reduce the influence of large deviations and provide a more robust fit in the presence of outliers, while values $p > 2$ place stronger emphasis on minimizing large residuals. This flexibility enables the user to adapt the optimization to the statistical properties of the data and the specific requirements of the application.

$$\phi = \sum_{ij} \left| \frac{(WH - X)_{ij}}{\sigma_{X,ij}} \right|^p + \lambda_W \sum_{ij} \left| \frac{(W - W_{measure})_{ij}}{\sigma_{W,ij}} \right|^p$$

III. CODE

This section outlines the optimization procedures used in order to build the sample concentrations matrix X from the original measurements. The problem dimension is high enough that the optimization is subject to local minima. The solution is that the procedure is run numerous times (as the user specifies) with different initial conditions both for H and for W. The initial conditions for W are generated from perturbation of the original measurements with random noised scaled with their standard deviation σ_W.

After all iterations, the solution with the smallest objective function value is chosen as the optimal solution. Additionally, the average and standard deviation over iterations in the estimated H matrices are reported in order to assess the consistency and stability of the inferred mixing ratios.

A. Optimization of W and H

The objective is the reconstruction of X through the simultaneous optimization of W and H. Allowing variation in W allows some flexibility in the estimation of H, as the latter is in itself an erroneous measurement. Varying its values prevents convergence upon locally optimal solutions, particularly when the weight parameter λ_W is more than unity. λ_W is initialized at 0.5 but may be varied manually based on relative reliability of the measurements of the compared variables W and X.

B. Two-Level Optimization Scheme

There are two related loops in each iteration of the optimization. The outer one changes both W and H, and the inner one improves certain variable subsets in the matrices and hence injects randomness as well as reduces the chance of convergence at local minima.

For H, the rows (or equivalently the columns of W) are ordered at random and optimized sequentially using *scipy.optimize.minimize* and the *SLSQP* algorithm. The algorithm supports constraints on H and is founded on a quasi-Newton estimate of the Hessian. For W, the *L-BFGS-B* algorithm is chosen because this is more efficient and no constraints are required.

C. Rationale for Stochastic Column-wise Optimization

Column-by-column optimization of individual variables is employed mostly as insurance against local minima. If all variables were optimized at the same time, the system potentially would converge as a group to some suboptimal solution. However, sequential optimization reduces this chance because the variables that have not converged yet have mobility in order to escape local minima. Individual variable optimization is more costly because more optimization steps are employed but each step is computationally cheaper because there are fewer active variables.

D. Comparison and Limitations

The algorithm currently lacks the implementation of uncertainty quantification. However, in comparison to established methodologies, Soupy demonstrates a significant benefit over traditional End-Member Mixing Analysis (EMMA), as it possesses the capability to concurrently process high-dimensional datasets, rather than being limited to low-dimensional tracer spaces.