USER GUIDE:
MIXING ELEMENTS AND DISSOLVED ISOTOPES IN RIVERS (MEANDIR)

PRESTON COSSLETT KEMENY[1*], MARK ALBERT TORRES[2]

[1] Geological and Planetary Sciences, California Institute of Technology, Pasadena, CA, USA
[2] Earth, Environmental, and Planetary Sciences, Rice University, Houston, TX, USA
*pkemeny@caltech.edu, preston.kemeny@gmail.com

1. INTRODUCTION

This guide explains the organization and use of Mixing Elements ANd Dissolved Isotopes in Rivers (MEANDIR). The goal of this guide is to allow researchers to invert dissolved river chemistry with basic understanding and relative ease, rather than to provide an exhaustive report of the 40 MATLAB files and 4 excel worksheets comprising the model. This guide thus contains sections on required user inputs, inversion variables, and a brief model walk-though. File names are given in blue, variables names in green, and text values in purple. For those interested in deeper understanding, the model is commented and the header of each file provides a summary of its function. Lastly, feel free to contact Preston Cosslett Kemeny with any questions.

2. REQUIRED USER INPUTS

MEANDIR requires four user inputs: a scenario name, observations of dissolved river chemistry, inversion end-members, and parameters for how to perform the inversion. MEANDIR defines these inputs in two MATLAB scripts (MEANDIR_Master, MEANDIR_FindScenarioParameters) and two excel spreadsheets (MyDefaultRiverDataSpreadsheet, MEANDIR_UserEntries). Note that the workbook MEANDIR_UserEntries contains three spreadsheets for defining end-members and conversion factors, but that only the end-members sheet will generally require modifications.

2.1. Scenario Name: MEANDIR_Master is the master script of the model. The only required input in this script is the vector ScenarioList, which is defined near the top of MEANDIR_Master. The model will progress sequentially through entries of ScenarioList and treat each entry as a unique experiment with separate results. The default value of ScenarioList is MyDefaultScenario (line 24). If inverting only one dataset, we recommend starting with this default value. Upon download, MyDefaultScenario is configured to run a short inversion similar to scenario JG-2 (tables 5, 7).

2.2. Scenario Parameters: Users set inversion variables in MEANDIR_FindScenarioParameters. The first part of this script is an *if* statement that searches for the current entry of ScenarioList. Once found, each of 44 variables is defined (see section 3 of this guide). The two parameters Riverdatasource and EMdatasource identify the river data and end-members for the inversion, respectively, and are always required. However, for most scenarios many of the other variables are optional. After defining variables, tests within MEANDIR_FindScenarioParameters attempt to identify common mistakes. Upon download, MEANDIR_FindScenarioParameters contains the full inversion information for all 28 scenarios reported in the main manuscript, plus an entry for MyDefaultScenario. Users can define new scenarios, such as MyDefaultScenario2, by expanding the primary *if* statement of MEANDIR_FindScenarioParameters.

2.3. River Observations: Having found Riverdatasource in MEANDIR_FindScenarioParameters, MEANDIR_Master calls MEANDIR_ReadRiverDataIntoMatlab to match Riverdatasource with instructions on reading the dissolved chemical observations from an excel spreadsheet. The default Riverdatasource is MyDefaultRiverData, which imports MyDefaultRiverDataSpreadsheet using MEANDIR_StandardInput. Upon download, MyDefaultRiverDataSpreadsheet contains the observations from Gaillardet and others (1999). If inverting a single dataset, simply populate MyDefaultRiverDataSpreadsheet with new data and leave ScenarioList = {'MyDefaultScenario'} and Riverdatasource = 'MyDefaultRiverData'. Note that the error of dissolved chemistry should be given as 1σ absolute errors in the data spreadsheet. Missing data can be left blank for individual samples, but a given sample will only be inverted if all variables for the inversion have values in the spreadsheet. For inputting multiple sets of data, the simplest way to enter the second set of observations into MEANDIR is to copy the spreadsheet MyDefaultRiverDataSpreadsheet, save it under a new name such as MyDefaultRiverDataSpreadsheet2, and put the second set of data in the relevant columns. Then, add a new section to the *if* loop of MEANDIR_ReadRiverDataIntoMatlab to evaluate whether or not Riverdatasource is equal to a new value, such as MyDefaultRiverData2. Within the new portion of the loop, copy code from the existing section but give the name MyDefaultRiverDataSpreadsheet2 to xlsread. The river observations are read into the structure variable river within the field observations. MEANDIR_ReadRiverDataIntoMatlab also converts concentrations into charge equivalents using conversion factors from the sheet MEANDIR_conc2equi within MEANDIR_UserEntries, as indicated in the names of the variables Ex. river.observations.Ca_conc and river.observations.Ca_equi give sample Ca^{2+} in μM and equivalents, respectively.

2.4. End-member Chemical Distributions: The MEANDIR_Endmembers sheet of the excel workbook MEANDIR_UserEntries defines the elemental and isotopic compositions of end-members. End-members are organized within groups, identified in column A, and the group name should match EMdatasource. Column B identifies the distribution of end-member chemistry (normal, uniform, log-uniform), column C is the active variable, column D is the desired normalization (almost always Na or SumObs), and column E is the type of entry (mean, minimum, etc.). The workbook then constructs a name for the variable in column F. If an entry of F says False, that end-member distribution will not be read correctly into MEANDIR. All columns after F correspond to end-members. The first row gives a long name of each end-member and the second row is an abbreviated name used to identify each end-member in the inversion. Additional end-members are added simply by populating new columns of the spreadsheet. Note that using $\chi^{riv}_{Cl^- critical}$ values requires defining a prec end-member and combining FeS_2 oxidation with rock weathering requires defining a pyri end-member. To define an end-member distribution relative to individual samples, enter sample#X. For example, when an end-member should equal the sample measurement, enter sample#0. To define an end-member as fractionated relative to the inputs from other end-members, the entry should say frac#X where X is the desired fractionation factor Δ (equations 29-32 of main text). For example, when an end-member should equal the inputs from all other end-members, enter frac#0 to represent no fractionation. Entering the code EM = MEANDIR_ReadEndMembersIntoMatlab reads the end-members into the structure EM; for example, EM.AB18_JG99_Na_normal.carb.NOR_MenCaNa would return the mean Ca^{2+}/Na^{+} ratio of the carbonate end-member in the end-member group AB18_JG99_Na_normal. If a new end-member does not appear in EM, check that the first and second rows of the spreadsheet contain names without special characters. If the chemistry of the end-members is missing, confirm that the

variable names in column F of MEANDIR_Endmembers match expectations. After the scenario is complete, the relevant entries of EM are also saved within the EndMembers field of river.

2.5. Model Results: The output structure river contains information on model inputs, the full distributions of end-member fractional contributions, inversion-constrained elemental and isotopic end-member ratios, and weathering variables. Much of the results are condensed into metrics such as the median and $5^{th}/25^{th}/75^{th}/95^{th}$ percentiles, although raw inversion results are also available within most subfields. Some of the following fields are saved for each scenario:

- settings: User-selected inversion parameters
- info: Some variables (name, latitude, etc.) and overview of results
- observations: User-reported river data (information from the spreadsheet)
- model_variable: Data used in the scenario (in same units as EMUnits)
- RiverMatrix: Matrix of river observation ratios
- InvertedRiverValues: Data used in simulations (model_variable if AdjustRiverObs = 0)
- fractionation: Fractionation factors used in the simulations
- inversionmatrix: End-member matrices used in the simulations
- reconstructed: Simulation-constructed sample chemistry
- fraction: Fractional contributions from each end-member to each variable
- misfit: Simulation misfits, both cost function and model-observation
- massbalance: Sum of fractional contributions from end-members to each variable
- EndMembers: Inversion-constrained composition of each end-member
- RZCWY: Calculated values of R, Z, Y, W, and C for each sample
- calculated_other_d34S: Calculated $\delta^{34}S$ of the other end-member and/or FeS_2 end-member
- ExcessSO4: The fractional and absolute amount of excess SO_4^{2-}

3. MODEL VARIABLES

MEANDIR_FindScenarioParameters defines 44 variables divided among 5 groups: (1) river observations, (2) general settings, (3) end-member variables, (4) Cl^- correction, and (5) calculating R, Z, Y, W, and C. Below, we briefly describe each of these variables.

3.1. River Observations and Normalization

1. Riverdatasource: Keyword identifying the source of river data to be inverted. Input as a string. Ex. Riverdatasource = 'JG99RiverData' could correspond to Gaillardet and others (1999). As described above, the default for the 'MyDefaultScenario' scenario is that Riverdatasource = 'MyDefaultRiverData'.

2. AdjustRiverObs: Whether or not river measurements should be adjusted to reflect analytical uncertainty. Can be either 0 or 1. If 1, on each simulation river chemistry is drawn from a normal distribution with the mean and standard deviation defined in the data spreadsheet. That is, river dissolved load is modified to reflect user-given analytical error. Ex. AdjustRiverObs = 1.

3. ObsList: Variables included in the mixing model. Inputs given as strings within a cell. Ex. ObsList = {'Na','Cl','Ca','Mg','K','SO4', 'd34S'} would include observations of $\chi^{riv}_{Na^+}$, $\chi^{riv}_{Cl^-}$, $\chi^{riv}_{Ca^{2+}}$, $\chi^{riv}_{Mg^{2+}}$, $\chi^{riv}_{K^+}$, $\chi^{riv}_{SO_4^{2-}}$, and ${}^{34}R^{riv}_{SO_4^{2-}}$ in the inversion. Altering the entries of ObsList modifies the variables included in the inversion. MEANDIR can identify the following dissolved observations:

ALK, DIC, Ca, Mg, Na, K, Sr, Cl, SO4, NO3, PO4, Si, Ge, Li, F, B, Re, Mo, Os, HCO3, d7Li, d13C, d18O, d26Mg, d30Si, d34S, d42Ca, d44Ca, Sr8786, d98Mo, Os8788, and Fmod. Additional observations can be added to MEANDIR or use the place of an otherwise unused variable.

4. CostFunType: Whether or not each variable in ObsList should be evaluated with a relative cost function or absolute cost function. Options are rel and abs. Ex. CostFunType ={'rel','rel','rel','rel'} would use a relative cost function for the four observations in ObsList. CostFunType serves the purpose of the vector **ω** in equation 18 of the main text.

5. WeightingList: Weighting terms for cost function. Ex. WeightingList = [1 1 1 1] would multiply the misfit between model results and observations for each of four observations in ObsList by 1. WeightingList serves the purpose of the vector **ϒ** in equations 16, 17, and 18 of the main text.

6. ErrorCutMinMB: When IterateOver equals Samples, ErrorCutMinMB sets the minimum fraction of concentration observations, or the maximum allowable negative offset for isotopic observations, for a simulation to count as successful. Ex. ErrorCutMinMB = [85 85 85 -2] when ObsList = {'Na','Cl','SO4', 'd34S'} defines the success criteria as simulations where >85% of $\chi^{riv}_{Na^+}$, $\chi^{riv}_{Cl^-}$, and $\chi^{riv}_{SO_4^{-2}}$ are recreated and where the model-constructed $\delta^{34}S$ value is >2‰ lower than the measurement. This vector thus sets the lower boundary of sample matching criteria. When IterateOver equals End-members, all values of ErrorCutMinMB are set equal to $-1x10^{10}$.

7. ErrorCutMaxMB: When IterateOver equals Samples, ErrorCutMaxMB sets the maximum fraction of concentration observations, or the maximum allowable positive offset for isotopic observations, for a simulation to count as successful. Ex. ErrorCutMaxMB = [115 115 115 3] when ObsList = {'Na','Cl','SO4', 'd34S'} defines the success criteria as simulations where <115% of $\chi^{riv}_{Na^+}$, $\chi^{riv}_{Cl^-}$, and $\chi^{riv}_{SO_4^{-2}}$ are recreated and where the model-constructed $\delta^{34}S$ value is <3‰ higher than the measurement. This vector thus sets the upper boundary of sample matching criteria. When IterateOver equals End-members, all values of ErrorCutMinMB are set equal to $1x10^{10}$.

8. nCFList: Variables in ObsList that should not be included in the cost function but should be reconstructed by the model. Ex. nCFList = {'HCO3'}. nCFList serves the purpose of the vector ξ in equation 18 of the main text, and its role is demonstrated in the comparison of AB-2 and AB-3 (fig. 6D in the main text). nCFList will often be empty.

9. ConvertDelta2RList: Isotopic variables in δ notation to be converted into isotopic ratios for inversion, typically to allow meaningful evaluation with a proportional cost function. Inversion results are converted back to δ notation for ease of interpretation. Ex. ConvertDelta2RList = {'d34S'} would convert $\delta^{34}S$ values to $^{34}S/^{32}S$ ratios prior to inversion. Values to convert from δ notation to ratios are in MEANDIR_DeltaNotationToR within MEANDIR_UserEntries.

10. AllIonsExplicitlyResolved: Indicates if all major cations and anions are included in ObsList. Values are 1 or 0. If 1, ListChargeClosure ensures that the sum of normalized positive charge equals the sum of normalized negative charge. Ex. AllIonsExplicitlyResolved = 1.

11. ObsInNormalization: Variables in the normalization. Entries can include cations, anions, and neutral species. Ex. ObsInNormalization = {'Ca','Mg','Na'}. To ensure internal consistency of end-members, one ratio contributing to the normalization variable is solved using other entries.

12. ImposeNormalizationCheck: Whether to impose a constraint that the sum of fractional contributions to the normalization variable must be between the lowest non-isotopic value of ErrorCutMinMB and highest non-isotopic value of ErrorCutMaxMB. Values are 0 or 1. Ex. ImposeNormalizationCheck = 1.

3.2. Simulation Settings

13. Solver: The numerical technique used to invert each simulation. Options are: mldivide, lsqnonneg, mldivide_optimize, lsqnonneg_optimize, and optimize. The choice mldivide refers to standard least squares inversion and lsqnonneg is a non-negative least squares inversion, both of which optimize absolute cost functions. When followed by _optimize, the solver is a constrained optimization using the first solver's result as the initial condition. Depending on the values of CostFunType, mldivide_optimize, lsqnonneg_optimize, and optimize can optimize for either absolute or proportional misfit. If Solver equals optimize the model uses an initial condition of equal fractional contributions from each of the end-members. Input as a string. Ex. Solver = 'mldivide_optimize'. See fig. 4 of the main text for comparison of the different solvers.

14. IterateOver: Options are Samples and End-members. If equal to Samples, each sample is treated fully independently. If End-members, all samples are inverted using the same end-members. MEANDIR supports parallel processing at two mutually exclusive locations within MEANDIR_Master. The first location is a loop from 1 to the number of iterations (lines 176 and 177 of MEANDIR_Master). When IterateOver equals End-members, the code will run faster when this loop is set to be *parfor*. The second location is a loop from 1 to the number of samples (lines 215 and 216 of MEANDIR_Master). When IterateOver equals Samples, the code will run faster when this second loop is set to *parfor*. Moreover, when IterateOver equals Samples, the code will generate a transparency error if the first loop is *parfor* and the second loop is *for*. Input as a string. Ex. IterateOver = 'End-members'.

15. maxiterations: When IterateOver equals Samples, this is maximum number of simulations attempted per sample. Define as a number. Ex. maxiterations = 10000. If IterateOver equals End-members, MEANDIR sets maxiterations to equal 1.

16. maxsuccess: When IterateOver equals Samples, this is the desired number of successful simulations for each sample. MEANDIR stops testing each sample when this number of successful simulations is found or the maximum number of simulations is reached. Input as number. Ex. maxsuccess = 500. If IterateOver equals End-members, MEANDIR sets maxsuccess to equal 1.

17. numberiterations: When IterateOver equals End-members, numberiterations is the number of iterations to be performed. Because this number of simulations will be stored and later subsampled, large values can cause MEANDIR to run slowly. Input as a number. Ex. numberiterations = 10000. If IterateOver equals Samples, MEANDIR sets numberiterations to equal 1.

18. MisfitCuts: When IterateOver equals End-members, this fraction of samples with lowest misfit will be kept (in units of %). Input as a number. Ex. MisfitCuts = 1 keeps the 1% of simulations with lowest misfit between models results and observations. MisfitCuts = 100 will keep all results. If IterateOver equals Samples, MEANDIR sets MisfitCuts to equal NaN.

19. CullOn: When IterateOver equals End-members, this is whether inversion results should be cut to the lowest MisfitCuts fraction based on misfit at the level of each individual samples or the entire sample set. Values are EachSample or AllSample. Input as string. Ex. CullOn = 'AllSample'.

20. saveuncutdata: When IterateOver equals End-members, this variable controls if all results are saved (in addition to the fraction defined by MisfitCuts) in a different subfield. Value is 0 or 1. Ex. saveuncutdata = 0. This is useful for evaluating the impact of culling the model results, but can result in very large files.

3.3. End-member Variables

21. EMdatasource: Identifies the end-member group (column A of MEANDIR_Endmembers in MEANDIR_UserEntries). Input as string. Ex. EMdatasource = 'AB18_JG99_SumObs_normal'.

22. EMList0: End-members to be included in the inversion. Each end-member is identified using the code from row 2 of the MEANDIR_Endmembers spreadsheet. Input as strings within a cell. Ex. EMList0 = {'prec','carb','dolo','slct','biot','clay'} will include six end-members in the inversion.

23. MinFractionalContribution: For constrained optimization, this is the list of minimum fractional contributions from each end-member to the normalization, corresponding to entries of EMList0. Ex. MinFractionalContribution = [0 0 0 -inf] constrains optimization to have a minimum contribution of 0 from each of the first three end-members and no minimum from the fourth (contributions outside the range of 0 to 1 may be required when modeling secondary phase formation).

24. MaxFractionalContribution: For constrained optimization, the list of maximum fractional contributions from each end-member to the normalization, corresponding to entries of EMList0. Ex. MaxFractionalContribution = [inf inf inf 0] constrains optimization to have no maximum contribution from the first three end-member and a maximum contribution of 0 from the last entry.

25. ListNormClosure: When normalizing to the sum of variables, define the variable for each end-member that will be calculated by mass balance to ensure internal consistency. This variable will often be the most abundant cation for that end-member. Inputs given as strings corresponding to each entry of EMList0. Ex. ListNormClosure = {'Na','Ca','Cl','Ca'}.

26. ListChargeClosure: When AllIonsExplicitlyResolved equals 1, this vector lists the variable for each end-member solved through charge balance. Often, this is the anion at highest abundance. Inputs as strings corresponding to EMList0 entries. Ex. ListChargeClosure ={'Cl','SO4','HCO3'}.

27. EMUnits: Whether end-member chemistry is given as concentration ratios or charge-equivalent ratios. Must be conc or equi. Input given as string. Ex. EMUnits = 'equi'.

28. EMsources: End-members that count as sources of dissolved variables. Without secondary phases, EMsources is typically equal to EMList0. Ex. EMsources = {'prec','bslt', 'carb','htsp'}.

29. EMsinks: End-members that count as sinks of dissolved variables, such as clays and other secondary phases. EMsinks is often empty or only contains one entry. Ex. EMsinks = {'clay'}.

30. EndMembersWithNegativeRatios: List of end-members where chemical ratios may be negative. Be careful to account for these selections relative to MinFractionalContribution and MinFractionalContribution. Ex. EndMembersWithNegativeRatios = {'pyri'}.

31. CoupleFeS2SO4intoEM: List of end-members where SO_4^{2-} values are overwritten by the pyri end-member. Input given as strings in a cell. Ex. CoupleFeS2SO4intoEM = {'carb','slct'}. Note that end-member SO_4^{2-} values are overwritten by the distribution of FeS2SO4 values in end-member pyri.

32. CoupleFeS2d34SintoEM: List of end-members where SO_4^{2-} $\delta^{34}S$ values are overwritten by the pyri end-member. Input given as strings in a cell. Ex. CoupleFeS2d34SintoEM = {'carb', 'slct'}. Note that end-member SO_4^{2-} δ^{34} values are overwritten by the distribution of FeS2d34S values in end-member pyri.

33. RecordFullFeS2Distribution: Whether or not to record the full distributions of calculated other $\delta^{34}S$ and FeS_2 $\delta^{34}S$ when $\delta^{34}S$ is not in the inversion. Either 0 or 1. Ex. RecordFullFeS2Distribution = 0. Recording the full distribution will result in larger file sizes.

34. BalanceEvaporite: Whether or not evaporite $SO_4^{2-} = Ca^{2+} + Mg^{2+} + Sr^{2+}$ and evaporite $Cl^- = Na^+ + K^+$. Either 0 or 1. Ex. BalanceEvaporite = 1.

3.4. Cl^- and Precipitation:

35. PrecProcessing: Whether to use values of $\chi^{riv}_{Cl^-\ critical}$. Options are ClCrit or EndMember. If ClCrit, $\chi^{riv}_{Cl^-\ critical}$ are subtracted from $\chi^{riv}_{Cl^-}$ and other measurements are corrected by precipitation ratios. If PrecProcessing equals EndMember, no special correction is made. Only used if prec is included in EMList0. Input is given as a string. Ex. PrecProcessing = 'ClCrit'.

36. ClCriticalValuesGiven: Indicates whether or not $\chi^{riv}_{Cl^-\ critical}$ values are provided. Only used if prec is included in EMList0. Must be either 0 or 1. If $\chi^{riv}_{Cl^-\ critical}$ values are known, this value should be 1. If PrecProcessing is ClCrit but ClCriticalValuesGiven equals 0, 100% of river Cl^- will be subtracted (MEANDIR will assume that $\chi^{riv}_{Cl^-\ critical} = \chi^{riv}_{Cl^-}$). Ex. ClCriticalValuesGiven = 0.

3.5. Weathering Parameters:

37. CalculateRZCWY: Whether or not MEANDIR should calculate R, Z, C, W, and Y. Options are: 0 or 1. Ex. CalculateRZCWY = 1 will attempt the calculations.

38. R_Numerator_EMList: End-members that contribute to the numerator of R (carbonate end-members). Inputs given as strings in a cell. Ex. R_Numerator_EMList = {'carb','dolo'}.

39. R_Numerator_IonList: Ions that contribute to the numerator of R. Inputs given as strings contained within a cell. Ex. R_Numerator_IonList = {'Na','Ca','Mg'}.

40. Z_NumeratorType: How the numerator of Z is calculated. Values are ZfromSO4excess, ZfromriverSO4, ZfromEM, or Znotcalculated. If ZfromSO4excess, the Z numerator is SO_4^{2-} in the river sample not attributable to end-members. If ZfromriverSO4, the Z numerator is SO_4^{2-} in the river. If ZfromEM, the Z numerator is SO_4^{2-} derived from end-members listed in Z_Numerator_EMList. If Znotcalculated, Z is not calculated. Ex. Z_NumeratorType = {'ZfromEM'}.

41. Z_Numerator_EMList: End-members that contribute to the numerator of Z. Use if Z_NumeratorType is ZfromEM. Input given as strings contained within a cell. Ex. Z_Numerator_EMList = {'carb', 'slct'} or Z_Numerator_EMList = {'pyri'.}

42. C_Numerator_EMList: End-members that contribute to the numerator of C (organic carbon end-members). Inputs given as strings in a cell. Ex. C_Numerator_EMList = {'corg'}.

43. RZC_Denominator_EMList: End-members that contribute to the denominator of R, Z, and C (all weathering end-members). Ex. RZC_Denominator_EMList = {'carb','dolo','slct','biot'}.

44. RZC_Denominator_IonList: Ions that contribute to the denominator of R, Z, and C. Ex. RZC_Denominator_IonList = {'Na','Ca','Mg','K'}. Our opinion is that this vector should contain all inverted cations, but there is debate in the literature on whether or not Na^+ and K^+ should be included.

4. MODEL WALK-THROUGH

MEANDIR defines inversion parameters in MEANDIR_FindScenarioParameters, uses MEANDIR_ReadEndMembersIntoMatlab to identify inversion end-members, and calls on MEANDIR_ReadRiverDataIntoMatlab to find river data. These steps, which entail most of the user involvement in the inversion, were our focus in the preceding text. We now briefly summarize the remainder of MEANDIR_Master.

4.1. Define the River Water Matrix: MEANDIR_GenerateRiverMatrix generates the matrix RiverMatrix0 containing normalized river observations. The rows of RiverMatrix0 correspond to the normalized elemental or isotopic ratios and columns correspond to samples.

4.2. Define the End-member Distributions: MEANDIR_makeEMdistributions generates an array of probability distributions for end-member chemistry, as well as multiple additional arrays encoding information on which end-members are defined relative to samples and which end-members are defined through fractionation relative to the inputs from other end-members.

4.3. Initiate Primary Model Loop: MEANDIR then initiates a loop to either iterate over samples or over end-member values. MEANDIR supports parallel processing at two locations: lines 176/177, and lines 215/216.

4.4. Adjust Observations Accounting for Analytical Error: If AdjustRiverObs equals 1, MEANDIR_AdjustRiverDataForAnalyticalError modifies observations of river chemistry and recalculates the normalization. In this case river observations are pulled from normal distributions with user-supplied means and standard deviations, the latter of which should be given in the data spreadsheets as 1σ absolute errors. MEANDIR_AdjustRiverDataForAnalyticalError is called but does not alter the river observations if AdjustRiverObs equals 0.

4.5. Calculate End-members: MEANDIR_PullEndMemberRatios calculates an internally consistent set of end-member values. MEANDIR_PullEndMemberRatios will attempt to generate a viable set of end-members up to maxtrycount times per function call (default value is 500).

4.6. Correct River Data for Precipitation: If PrecProcessing is ClCrit and EMList0 contains prec, MEANDIR_ClCriticalCorrection adjusts river observations for inputs from precipitation using values of $\chi^{riv}_{Cl^-\ critical}$. If the removal of the specified amount of Cl^- would cause another element to become negative, the end-member matrix is regenerated up to maxredefinition times per simulation (default value is 100).

4.7. Perform River Inversion: MEANDIR_InvertActiveSimulation sets the initial conditions of optimization and defines MEANDIR_CostFunction as an optimization problem for fmincon.

4.8. Evaluate Inversion Results: MEANDIR_EvaluateInversionInstance evaluates if the simulation generated results that meet user-supplied criteria for success. If IterateOver equals Samples, results are only saved if they satisfy the values defined in ErrorCutMinMB and ErrorCutMaxMB. If IterateOver equals End-Members, nearly all simulations are saved at this step.

4.9. Unpack Results: Following the inversion, MEANDIR_UnpackInversionResults aggregates the inversion results for subsequent calculations.

4.10. Cull Results: If IterateOver equals End-Members, MEANDIR_CalculateCullDatabyMisfit isolates the fraction of simulations with lowest misfit as set by MisfitCuts.

4.11. Calculations and Saving: MEANDIR calculates and saves variables of interest into subfields of the variable river. Relevant programs include MEANDIR_CalculateMassBalance, MEANDIR_SaveFractionalContributions, MEANDIR_SaveInvertedRiverValues, MEANDIR_SaveFractionations, MEANDIR_CalculateReconstructedObservations, MEANDIR_CalculateExcessSO4, MEANDIR_CalculateOther_d34S, MEANDIR_CalculateEndMemberValues, and MEANDIR_CalculateRZCWY.