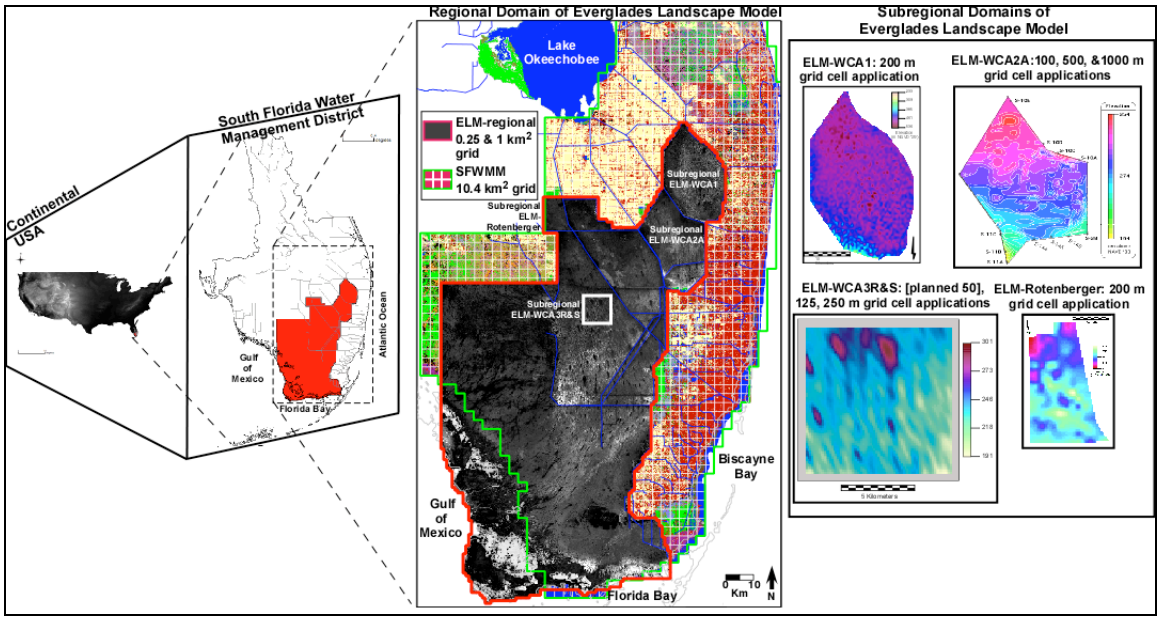
Documentation of the Everglades Landscape Model: ELM v2.8



http://ecolandmod.ifas.ufl.edu

H. Carl Fitz, Assistant Professor
Soil and Water Science Department
Ft. Lauderdale Research and Education Center
IFAS, University of Florida
3205 College Ave.
Davie, FL 33314
email: cfitz@ufl.edu

February 13, 2009

Table of Contents

Preface

Documentation purpose

This documentation report provides the information necessary to fully understand the *goals & objectives, supporting data, algorithms, performance, and application* of the Everglades Landscape Model (ELM). This document, the model source code & data, and further supporting information are maintained on the ELM-development web site:

http://ecolandmod.ifas.ufl.edu

The primary objective of the documentation is to describe the fine-resolution application of the regional ELM, for use in evaluating ecological responses to alternative management scenarios in the greater Everglades landscape. This is a documentation update, limited to describing changes that were made in model design and data during the transition from ELM v2.5 to ELM v2.8. A number of original ELM v2.5 Documentation Chapters are not included here, as their content remains unchanged; the documentation is available in the above ELM-development web site, and at the ELM-application web site:

http://my.sfwmd.gov/elm

The only Chapters included in this ELM v2.8 Documentation Report are those that contain significant new information that is relevant to current application objectives.

Document organization

Each Chapter of this document has its own Table of Contents.

(see ELM v2.5) Chapter 1: ***Introduction*** *to the Everglades and the model* ***Goals & Objectives****.*

(see ELM v2.5) Chapter 2: General overview of ***Wetland Ecological Models****.*

(see ELM v2.5) Chapter 3: Graphical and verbal descriptions of the South Florida and General Ecosystem ***Conceptual Models*** *on which the ELM is based.*

- Chapter 4: Graphical, verbal, and statistical-summary descriptions all of the updates to ***Data*** that are used in the new model application.
- Chapter 5: Graphical, verbal, and mathematical descriptions of the updates to ***Model Structure*** and algorithms (including links to source code).
- Chapter 6: Analysis of ***Model Performance*** relative to the historical period of record in the regional system (1981 - 2000).

(see ELM v2.5) Chapter 7: Aspects of ***Uncertainty*** *in the model and associated data, including sensitivity analysis, appropriate model expectations, and model complexity.*

(see ELM v2.5) Chapter 8: Descriptions of ***Model Application*** *in the regional Everglades system.*

(see ELM v2.5) Chapter 9: Descriptions of past and planned ***Model Refinements****, including an overview of its current limitations.*

- Chapter 10: A **User's Guide** that provides the simple steps to installing and running this Open Source model.

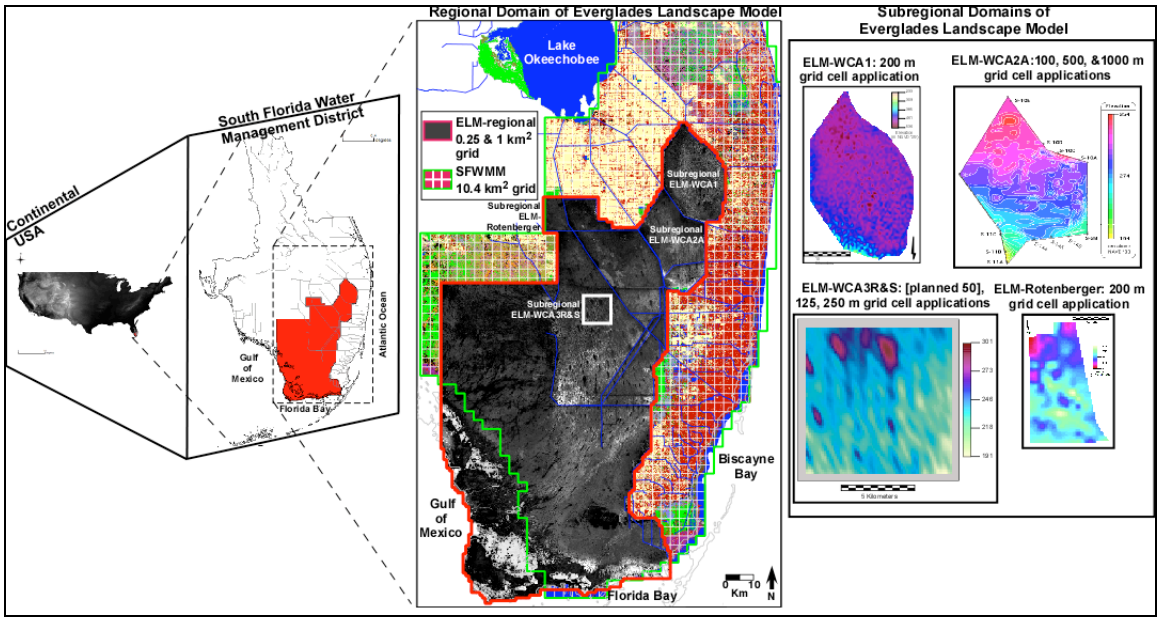
Acknowledgments

Funding for this ELM application came primarily from the U.S. Geological Survey, Priority Ecosystem Science Program, within the auspices of the Joint Ecosystem Modeling partnership. L. Brandt and R. Best provided management and direction for the model refinements described in this documentation update. See the ELM v2.5 Documentation Report for descriptions of other collaborations that were part of the overall ELM development.

Executive Summary

Today's Everglades are significantly different from the landscape that existed a century ago. Humans compartmentalized a once-continuous watershed, altering the distribution and timing of water flows, and increasing the quantity of nutrients that move into the Everglades. The result is a degraded mosaic of ecosystems in a region that is highly controlled by water management infrastructure. However, plans are being developed and implemented to restore parts of this system towards their earlier state.

To support scientific evaluations of restoration plans, computer simulation models can be used to predict the relative benefits of one alternative plan over another. One such tool is the Everglades Landscape Model (ELM). The ELM is designed to improve understanding of the ecology of the Everglades landscape, and can be applied at a range of spatial and temporal scales depending on the project requirements. This model integrates, or dynamically combines, the hydrology, water quality, and biology of the mosaic of habitats in the Everglades landscape. It is a state-of-the-art *model that is capable of evaluating long-term benefits of alternative project plans with respect to hydrology, water quality* and other ecological Performance Measures.



Existing regional and subregional applications of the ELM, including the 500 m grid resolution application developed for the regional Everglades system.

Because the ELM was designed to be explicitly scalable, it is relatively simple to adapt (spatial input map) data to accommodate the scientific objectives that may call for a particular scale of grid resolution or extent. There are a variety of ecological models (i.e., ATLSS (http://atlss.org)) that operate at a 500 m grid resolution, which is 4x finer resolution than that of the regional ELM v2.5. The finer scale was chosen by the model developers to capture more of the variability of animal community responses than possible with 1 km or larger grids. The hydrologic data used to drive these models is output from the South Florida Water Management Model (SFWMM), with 2x2 mile (~10 km^2) grid resolution. While there exist several innovative methods to obtain 500 m

resolution hydrologic data from the SFWMM output and fine scale land elevation data, there was interest in obtaining finer scale hydrology from the regional ELM.

At the request of the Joint Ecosystem Modeling[1] group, and with funding from the USGS Priority Ecosystem Science program, we developed the 500 m resolution regional ELM v2.8[2] in order to provide another alternative for "driving" the other ecological models with fine scale hydrologic data. Moreover, this finer scale regional ELM application is available for evaluating other hydro-ecological responses of the Everglades landscape to alternative scenarios of water and nutrient management. This ELM v2.8 Documentation Report includes the information necessary for scientists and planners to understand this application of ELM, including *a) the ELM objectives, b) how it works, c) how well it works, and d) how to interpret its results*.

The fine spatial scale and very good performance of the resulting model may be useful in a variety of projects involving Everglades synthesis and management. The model "skill" in predicting stage and flow was improved over earlier ELM versions, and continued to be consistent with the SFWMM output. Of particular interest with respect to "driving" fine scale ecological models, this scale of ELM hydrologic output exhibited detailed spatial patterns of flow, with improved connectivity among and within habitats (such as sloughs) relative to the 4x (ELM v2.5) or ~40x (SFWMM v5.4) coarser-scale resolution hydrologic models previously available for the greater Everglades region. In particular for the JEM partnership, making use of a model with a "native" (original) resolution of 0.25 km^2 may likely lead to improved realism in modeling animal communities which respond to hydrologic patterns at these relatively fine scales.

[1] JEM, a partnership among several US Dept. of Interior agencies, several universities, and government and non-government organizations

[2] The tertiary subversion designation of this v2.8 application release is v2.8.3.

Goals

• Develop a simulation modeling tool for integrated ecological assessment of water management scenarios for Everglades restoration

- Integrate hydrology, biology, and nutrient cycling in spatially explicit, dynamic simulations
- Synthesize these interacting hydro-ecological processes at scales appropriate for regional assessments,
- **Understand and predict the relative responses of the landscape to different water and nutrient management scenarios**
- Provide a conceptual and quantitative framework for collaborative field research and other modeling efforts

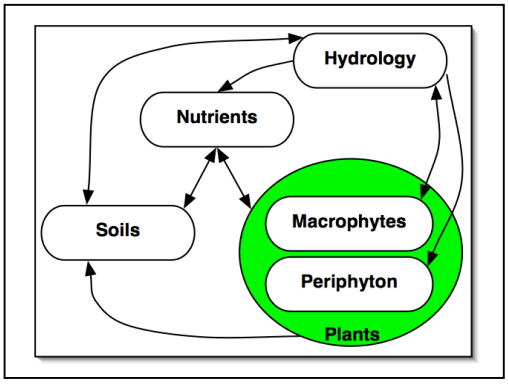
Design

• Can be applied at multiple spatial or temporal scales, for regional or subregional evaluations

- Regional application at fine resolution (40x finer than SFWMM[3])
- Multi-decadal (36-yr) simulation period

• Combine physics, chemistry, biology – *interactions*

- *Hydrology*: overland, groundwater, canal flows
- *Nutrients*: phosphorus cycling and transport
- *Periphyton*: response to nutrients and water
- *Macrophytes*: response to nutrients and water
- *Soils*: response to nutrients and water



• Combine ecological research with modeling

- research advances led to model refinements
- model output aided research designs

Reliability

• Excellent performance (1981 - 2000 history-matching)

- *Hydrology*: the offset (median bias) of predicted and observed values of water stage elevations in the marsh was 0 cm
- *Water quality*: the offset (median bias) of predicted and observed values of phosphorus in the marsh was 0 ug L^{-1}; chloride was 6 mg L^{-1}.

• Tested computer code

- evaluated model response to wide range of conditions (sensitivity analyses)
- years of experience in testing and refining code
- applied at different scales for regional and sub-regional evaluations

• Uses best available data

- comprehensive, unique summary of Everglades ecology
- thorough QA/QC of input data
- continuous interactions with other Everglades scientists and engineers

[3] South Florida Water Management Model, the widely-accepted simulation tool used for regional evaluations of water management alternatives

Model Reviews

- Open Source
  - All ELM data and computer source code freely available on web site
  - Requires only Open Source (free) supporting software
- Publications
  - 1996-2008: Peer-reviewed scientific journals and book chapters
  - 1993-2006: Technical reports published by SFWMD
- CERP[4] Model Refinement Team
  - 2003: Recommended independent peer review
- Independent Panel of Experts
  - 2006: Peer review of ELM by an independent panel of experts
- CERP Interagency Modeling Center
  - 2007: Review of ELM for CERP applications

Application

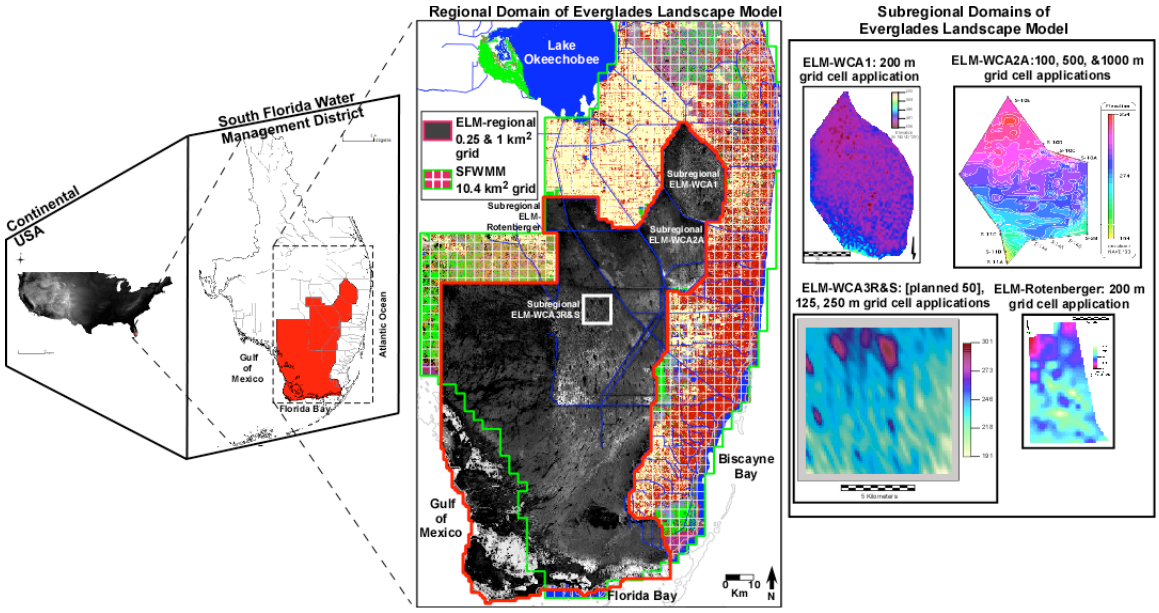
- Specific **model objectives** (Performance Measures)
  - For Joint Ecosystem Modeling partnership, provide fine-scale hydrologic output for use in "driving" other ecological models
  - ELM v2.5 – 2.8: Relative predictions of phosphorus 1) concentrations and 2) net load along spatial gradients in the greater Everglades, over decadal time scales
  - Other ecological Performance Measures proposed, pending model/data updates
- Appropriate applications
  - Relative comparisons of the above Performance Measures under scenarios of alternative water management plans
- Recent applications
  - ELM v2.8.2 application to subregional domain of Water Conservation Area 1, 200 m grid resolution; evaluated hydrologic and water quality responses to simple management scenarios

[4] Comprehensive Everglades Restoration Plan

BLANK PAGE

Documentation of the Everglades Landscape Model: ELM v2.8

Chapter 4: Data



http://ecolandmod.ifas.ufl.edu

February 13, 2009

Chapter 4: Data

4.1 Overview

The focus of this Chapter is the description of changes to data used in the 500 m resolution regional ELM v2.8 application, relative to those documented for the 1 km resolution regional ELM v2.5. The new fine resolution regional model encompasses the identical domain of 10,394 km^2, but with 42,576 active grid cells four times the 10,394 grid cells in the 1 km resolution version). For this ELM v2.8 regional application, most of the data remain the same as those used for the ELM v2.5 regional application. In parallel with the 500 m application, we continue to update/maintain the 1 km application (but which is not specifically covered in this document). The principal changes involved "resampling" data from the 1 km resolution map inputs, and generating new spatial interpolations of the updated land surface elevation data at the 500 m resolution. This ELM v2.8 Data Chapter thus makes extensive reference to the regional ELM v2.5 Documentation Report's Data Chapter.

4.2 Background

4.2.1 Application summary

Because the ELM was designed to be explicitly scalable, it is relatively simple to adapt (spatial input map) data to accommodate the scientific objectives that may call for a particular scale of grid resolution or extent. There are a variety of ecological models (i.e., ATLSS (http://atlss.org)) that operate at a 500 m grid resolution, which is 4x finer resolution than that of the regional ELM v2.5. The finer scale was chosen by the model developers to capture more of the variability of animal community responses than possible with 1 km or larger grids. The hydrologic data used to drive these models is output from the South Florida Water Management Model, with 2x2 mile (~10 km^2) grid resolution. While there exist several innovative methods to obtain 500 m resolution hydrologic data from the SFWMM output and fine scale land elevation data, there was interest in obtaining finer scale hydrology from the regional ELM.

At the request of the Joint Ecosystem Modeling[1] group, and with funding from the USGS Priority Ecosystem Science program, we developed the 500 m resolution regional ELM v2.8[2] in order to provide another alternative for "driving" the other ecological models with fine scale hydrologic data. Moreover, this finer scale regional ELM application is available for evaluating other hydro-ecological responses of the Everglades landscape to alternative scenarios of water and nutrient management.

The primary data change for this new application involved development of a 500 m resolution land elevation map. Most of the other data used in this application remain the same as those used in the regional ELM v2.5, and thus this Data Chapter 4 for this application makes extensive reference to the ELM v2.5 Documentation Report[3].

---

[1] JEM, a partnership among several US Dept. of Interior agencies, several universities, and government and non-government organizations

[2] The tertiary subversion designation of this v2.8 application release is v2.8.3.

[3] Fitz, H.C., and B. Trimble. 2006. Documentation of the Everglades Landscape Model: ELM v2.5. South Florida Water Management District. http://my.sfwmd.gov/elm Reviewed by independent expert panel, reported at http://my.sfwmd.gov/elm 664 pages.

4.2.2 Metadata

All of the input data files used in the model have metadata directly associated with them in the project data directories. This Chapter expands on the metadata by further detailing the sources and derivation of the data. The following table lists all of the files that are input to the ELM and described in this Chapter[4].

Type	Input filename	Description
Model domains		
	ModArea	Define spatial domain
	gridmapping.txt	Link coarse-fine grids
Initial condition maps		
	icSfWt	Initial surface water
	icUnsat	Initial unsaturated water
	Elevation	Initial land elevation
	Bathymetry	Initial (and constant) creek bathymetry
	soilBD	Initial (and constant) soil bulk density
	soil_orgBD	Initial (and constant) soil organic bulk density
	soilTP	Initial soil phosphorus
	HAB	Initial habitat type
	icMacBio	Initial total macrophyte biomass
Boundary conditions		
	BoundCond	Grid cells allowing boundary flows
	BoundCond_stage.BIN	Boundary stage/depth time series
	rain.BIN	Rainfall time series
	ETp.BIN	Potential ET time series
	AtmosPdepos	(optional) map, total atomospheric P deposition
	AtmosCLdepos	(optional) map, total atomospheric Cl deposition
	CanalData.struct_wat	Structure: water flow time series
	CanalData.struct_TP	Structure: phosphorus conc. time series
	CanalData.struct_TS	Structure: salt (chloride) conc. time series
	CanalData.graph	Recurring annual time series of stage regulation
Static attributes		
	CanalData.chan	Canal/levee parameters/locations
	CanalData.struct	Water control structure attributes
	basins	Basin/Indicator Region locations
	basinIR	Basin/Indicator Region hierarchy
	GlobalParms_NOM	Parameters: global
	HabParms_NOM	Parameters: habitat-specific
	HydrCond	Parameters: hydraulic conductivity

[4] Two other files, outside of the Project's "Data" directory in the "RunTime" directory, are input to the model and serve to configure the model at runtime. See the User Guide Chapter for information on the "Driver.parm" and "Model.outList" configuration files.

4.3 Model domains

4.3.1 Spatial domain

The ELM can be applied at a variety of grid scale resolutions and extents without changing any source code. For an application at a particular spatial grain and/or extent, the following data files are used to define the model at the desired scale: 1) the appropriate grid resolution/extent of each of the map input files; 2) the grid resolution and geographic (upper left) origin in the two databases that define the canal/levee locations and water control structure attributes; and 3) the linked-list text file that maps coarser-grid data to the selected model application. The User Manual Chapter explains these steps needed to develop an application at a new spatial resolution/extent.

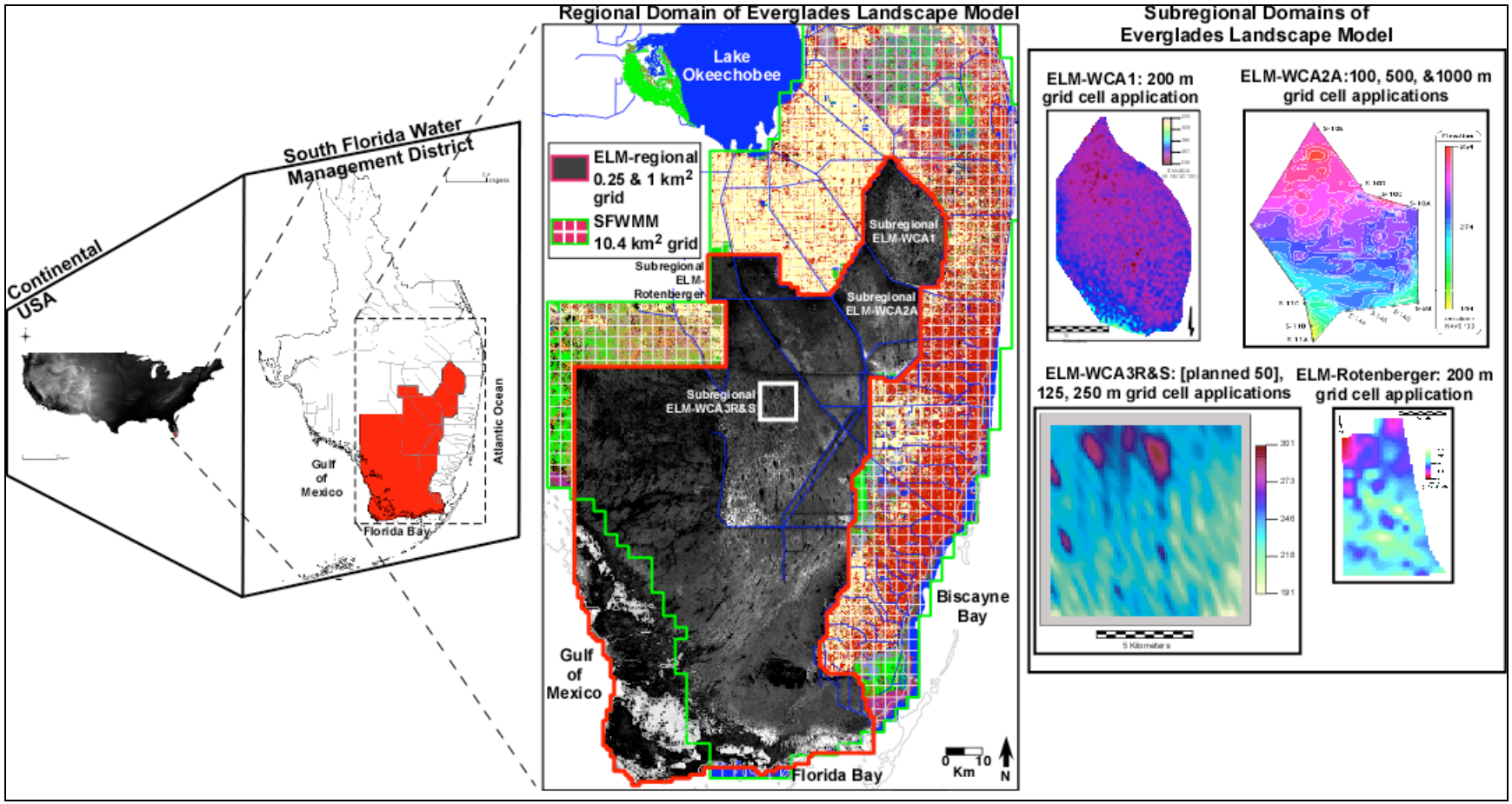
All spatial data are referenced to zone 17 of the Universal Transverse Mercator (UTM) geographic coordinate system, relative to the 1927 North American Datum (NAD).

4.3.1.1 Regional domain (infile = "ModArea")

The focus of this review is on the regional application of ELM to the greater Everglades region, from the northern Everglades marshes along the Everglades Agricultural Area to the mangroves along Florida Bay and the Gulf of Mexico. This region is generally restricted to the "natural" areas of the greater Everglades, including all of the Water Conservation Areas, Holey Land, Rotenberger Tract, most of Everglades National Park, and most of Big Cypress National Preserve (Figure 4.1). This regional application uses 0.25 km^2 square grid cells that encompass an area of 10,394 km^2 (4,013 mi^2). The 1 km regional application uses the same domain extent, but with 1 km grid resolution. All of the maps of the regional application are bounded by the following rectangle of UTM coordinates in zone 17 (NAD 1927):

northing:	2,952,489 m
southing:	2,914,489 m
easting:	578,711 m
westing:	553,711 m

Figure 4.1. The regional and subregional domains of the ELM, and the regional domain of the South Florida Water Management Model (SFWMM).



4.3.1.2 Subregional domains (infile = "ModArea")

The domains of existing sub-regional applications of the ELM are displayed in Figure 4.1. The grain of these subregional applications in the Rotenberger Tract, WCA-1, WCA-2A, and a small area in WCA-3A includes square grid dimensions of 100 m, 125 m, 200 m, 250 m, 500 m, and 1 km.

4.3.1.3 Multi-scale grid-mapping (input = "gridmapping.txt")

A variety of dynamic boundary condition data may be input from coarser model grids. The ELM v2.8 (and 1 km ELM v2.5) uses some dynamic boundary condition data (described in later sections) that are at the scale of the 2x2 mile (10.4 km^2) grid of the SFWMM. For regional or subregional applications of ELM, a "linked list" is generated to map boundary condition data from a coarse grid (usually that from the SFWMM) to the ELM grid. These data are generated from the pre-processor GridMap tool, and input to the ELM via the "gridmapping.txt" file.

4.3.1.4 Basins & Indicator Regions (input = "basins", "basinIR")

The map of the 64 Basins and Indicator Regions defines the spatial distribution of hydrologic Basins and Indicator Regions (BIR). These BIR spatial distinctions do not affect any model dynamics, but are used in summarizing nutrient & water budgets and selected ecological Performance Measures. Budgets and preset Performance Measure variables are output at the different spatial scales defined by the BIR. The Indicator Regions are particularly useful for summarizing model dynamics along ecological gradients.

The largest spatial unit is Basin 0, the "basin" of the entire domain. Hydrologic basin(s) within the domain are regions with either complete restrictions on overland flows (such as Water Conservation Area 1 surrounded by levees) or partial restrictions of overland flows (i.e., Water Conservation Area 3A is bounded by levees except along part of its western boundary). Hydrologic basins are "parent" regions that (may) contain "child" Indicator Regions. Indicator Regions are drawn within a hydrologic basin boundary (but an Indicator Region may not belong to two parent basins). In reporting BIR output data, parent basins' data include (e.g., sum) the data on all child Indicator Regions contained within them. When re-drawing the BIR ("basins") map, the user must edit the "basinIR" text file that defines the inheritance characteristics and allowable surface flows of the BIRs (such as the flow allowed to/from Water Conservation Area 3A through the gap mentioned above).

4.3.2 Temporal domain

The ELM can be applied at a variety of time scales, depending on the objective and the availability of boundary condition data. The temporal extent of the historical period used in evaluating model performance (calibration/validation) is 1981 – 2000.

The temporal extent of the available meteorological record (used in future alternative model evaluations) is 1965 – 2000. As detailed later in this Chapter for each boundary condition data file, the temporal grain of these input data is 1-day. As described in the

Model Structure chapter, the time step (dt) of the vertical solutions is 1-day, while the time step for horizontal solutions varies with the model grid resolution.

While the 1-km resolution applications of ELM utilize 12 horizontal time slices per day (2-hr dt), the 500-m resolution applications utilize 40 horizontal time slices per day (36-min dt).

4.4 Initial condition maps

There are a number of map data files that are necessary to implement this spatially explicit landscape model. Those that are used in defining the initial conditions of the simulation were developed using the methods described either below (if newly developed), or in the ELM v2.5 Data Chapter, for each specific data set. Note that the initial conditions for some variables do not have individual input map files (see the descriptions of the Global and the Habitat-specific parameter databases).

For many of the spatial data sets of the 500 m resolution ELM v2.8, we simply resampled, or resampled and then filtered, the 1 km resolution data used in ELM v2.5.

- **Map resample operation:** the 1-km resolution map data was resampled[5] at a 500 m grid resolution, resulting in an identical spatial distribution of values, but in a (larger) finer scale array at the 500 m spatial grain.
- **Map filter operation:** following a resample operation at 500 m grid resolution, a 3 x 3 (500 m) cell neighborhood-mean filter was passed across the entire domain with a moving window, returning the mean of the neighborhood for each (500 m) grid cell[6]

For the input maps whose underlying data were unchanged from ELM v2.5, we below indicate whether resampling alone, or resampling and filtering, was applied to generate the required 500 m resolution map for ELM v2.8.

4.4.1 Water depths

4.4.1.1 Surface water depth (input = "icSfWt")

ELM v2.5 1-km resolution data, resampled and filtered.

4.4.1.2 Unsaturated water depth (input = "icUnsat")

ELM v2.5 1-km resolution data, resampled and filtered.

4.4.2 Land surface elevation

An initial goal of this development was to directly mosaic the current EDEN[7] and TIME[1] elevation maps into the larger ELM domain, reconciling the differences in grid

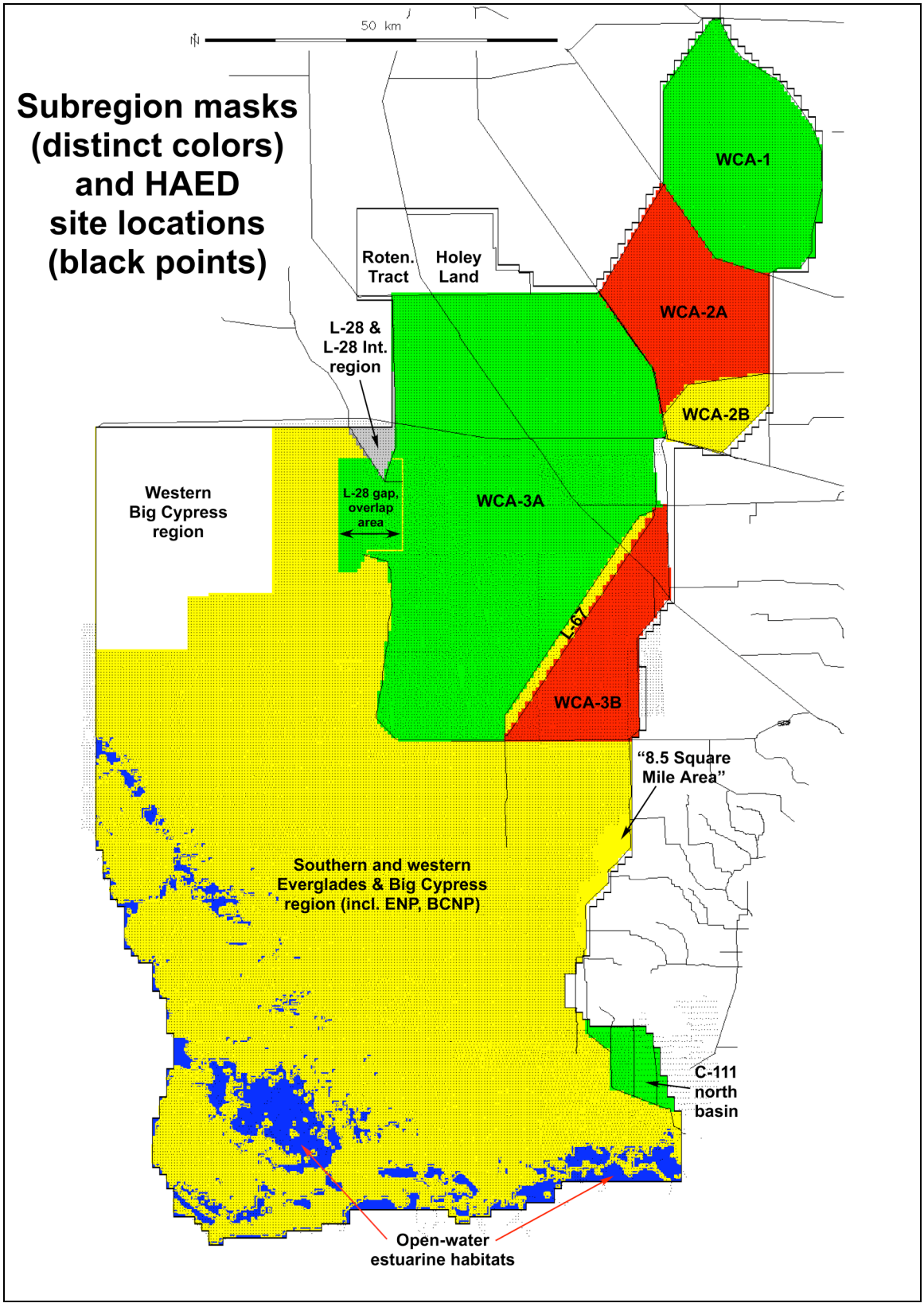
[5] GRASS v6.2.1, r.resample command, using the entire spatial domain defined above, and 500 meter resolution for both the easting and northing cell dimensions.

[6] GRASS v6.2.1, r.neighbors command (in the 500 m resolution domain), with neighborhood size = 3 cells, and neighborhood method = average

[7] Everglades Depth Estimation Network model; Tides and Inflows in the Mangroves model

resolutions and topology. Those EDEN (October 2007 version) and TIME (current as of December 2007) data were combined into an integrated elevation surface at 400 m resolution by USGS staff (J. Jones, December 2007 data sharing). Including the October 2007 version of the updated EDEN elevation map, this map was a significant improvement over earlier data that did not have accurate data for several regions, including northern Water Conservation Area 3A and Big Cypress National Preserve. However, the spatial rescaling and spatial offset necessary to convert those data to match the 500 m ELM grid resulted in a variety of scaling artifacts. Some of these artifacts had the potential to affect hydrologic flows, particularly along the critical edges of hydrologic basins. While some spatial filtering operations could alleviate some of these patterns, that step was deemed undesirable from the perspective of maintaining accuracy of the data product.

Figure 4.2. The raw point observations (black points in figure) were interpolated separately within multiple basins (depicted as separate colors), delineated by levees.

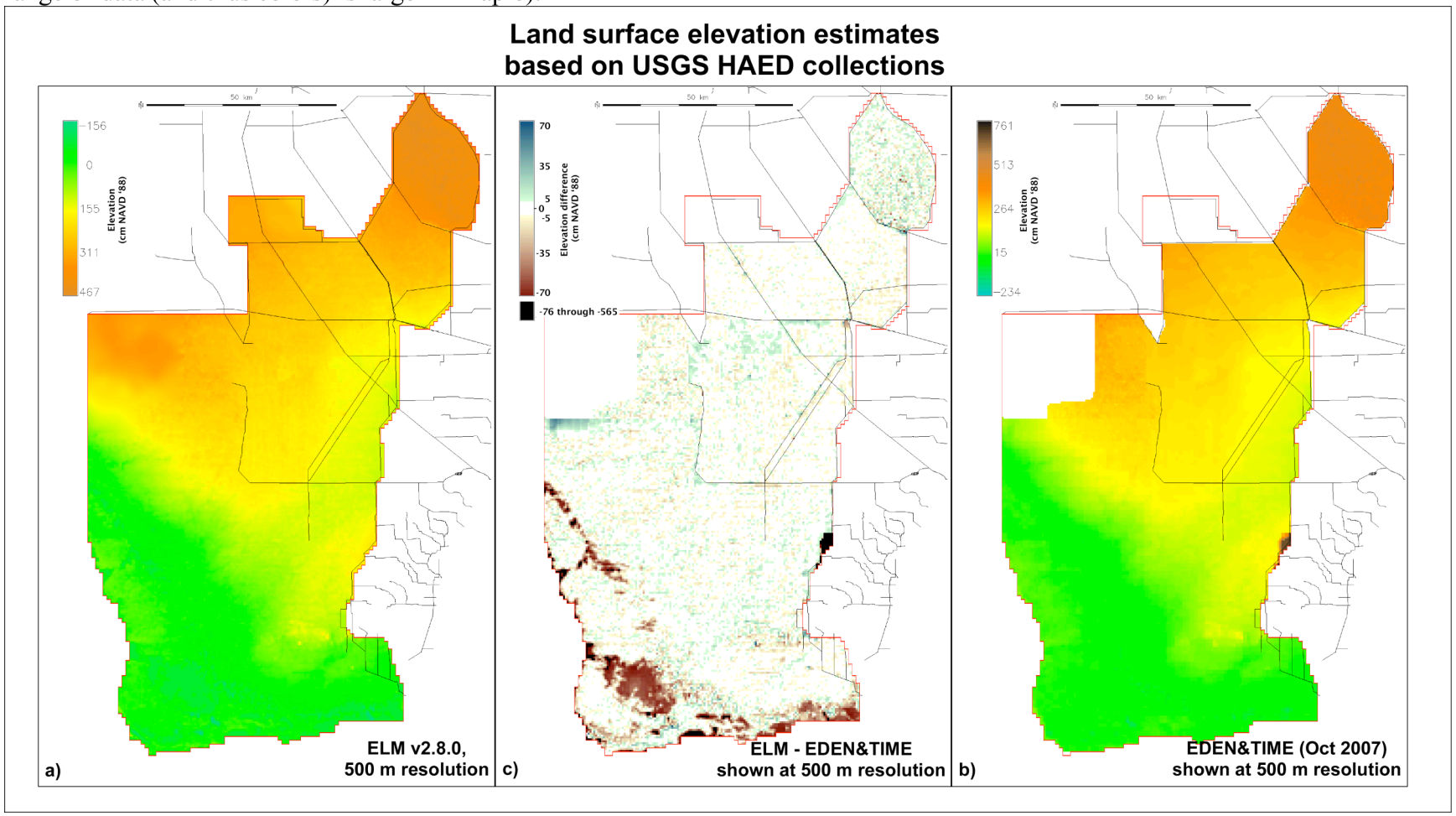


Thus, a new set of interpolations were made from High Accuracy Elevation Data (HAED) Project (Desmond 2003) observations within ELM hydrologic basins. The complete set of point observations was obtained from the USGS project's web site in December 2007. As shown in Figure 4.2, separate hydrologic basins (generally bounded by levees) were "masked" at a 250 m resolution, and within each raster basin mask, a "regular spline with tension" method[8] was used to generate a 250 m resolution elevation map. This 250 m resolution map was then run through a 3-neighbor (9 cell) mean filter, and resampled (within the basin mask) at 500 m resolution. These individual elevation maps were mosaic'd together, and filled in with elevation data from ELM v2.5 (converted to the NAVD '88 datum and scaled to 500 m resolution). All conversions from NGVD '29 to NAVD '88 were made using the "Corpscon" program, v. 6.0.1[9].

[8] Using GRASS GIS, v.surf.rst command, no smoothing, tension parameter at default value=40. This method was developed, and documented within GRASS manual pages, specifically for interpolations of elevation data sets at a variety of scales. Further analysis of observed-modeled differences and cross-validation products may be provided with the final version of the elevation map.

[9] http://crunch.tec.army.mil/software/corpscon/corpscon.html

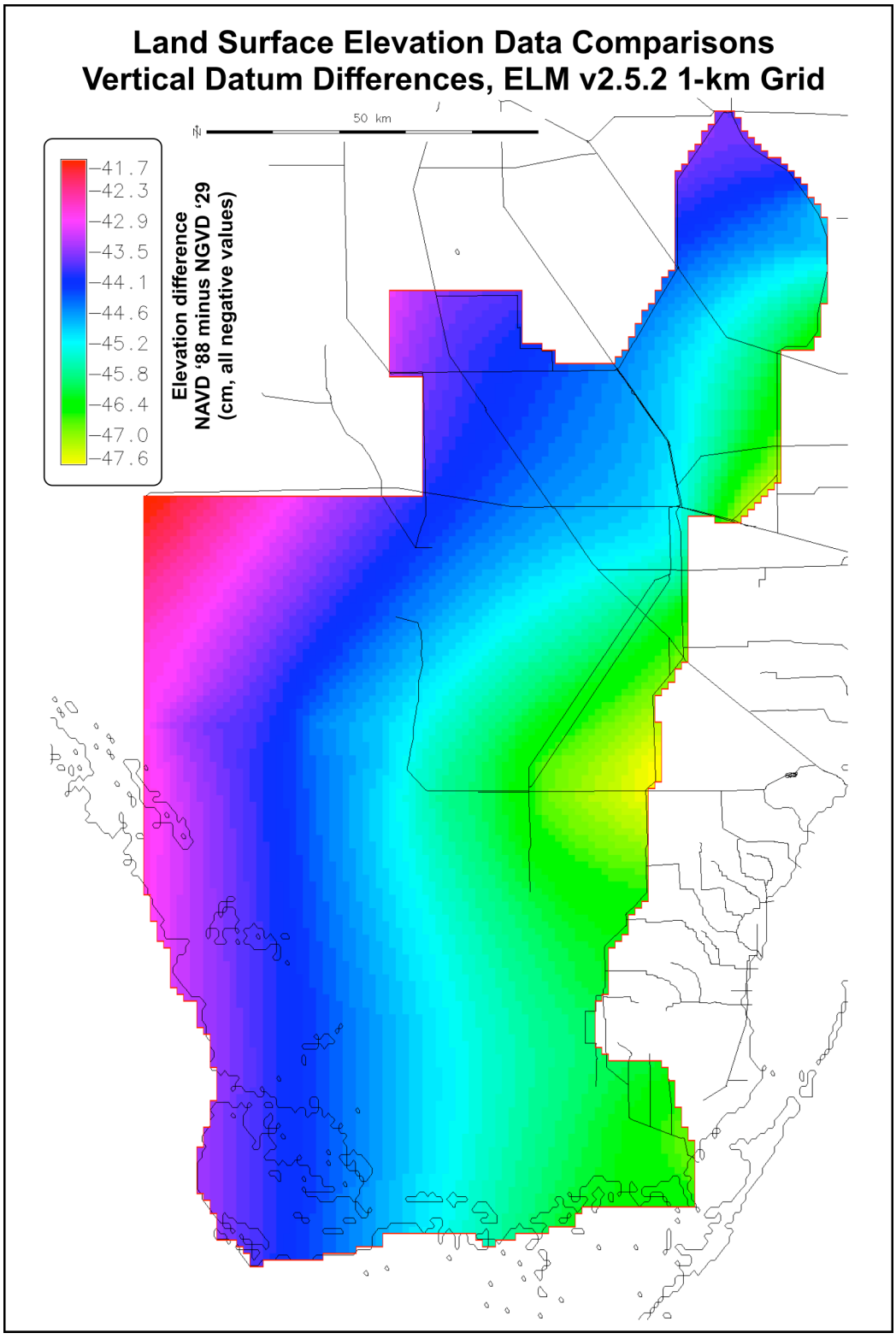
Figure 4.3. Interpolated surfaces representing land surface elevation: a) map indicating the ELM regional-500m v2.8.3 initial elevation, b) map indicating the combined EDEN and TIME elevation (shown at 500 m resolution), and c) the difference between the two maps (shown at 500 m resolution). Note that maps a) and b) share a common color table for the same data values, although the range of data (and thus colors) is larger in map b).



A simple visual analysis of the differences between the final regional elevation map and the data in the EDEN&TIME map (Figure 4.3) showed a high degree of consistency between the elevation data sets, particularly removed from edge areas of hydrologic basins. Some of the most significant differences appeared in the open water estuarine areas. These deep-water regions were almost entirely omitted by the HAED collections (Figure 4.2), and the TIME elevation data set that was provided exhibited elevations that were generally little different from surrounding land. To generate the ELM elevation map (Figure 4.2), estuarine areas were delineated as a special "basin" (masked by an ELM map of open water estuarine habitat type). All of the (very sparse) data points within this mask were interpolated as described earlier, and then all raster cells in the masked region that were shallower than (negative) -75 cm NAVD '88 elevation were recoded to that elevation value, i.e., if necessary, an estuarine cell was assigned an elevation that made it at least 30 cm below the (approximate) mean sea level (~ -45 cm NAVD '88). The other major difference between the EDEN&TIME and ELM elevation maps involved the LIDAR (i.e., not HAED) data used in the 8.5 Square Mile Area. Those data were deemed to be significantly in error, and the 500 m grid cells in the ELM map were assigned (NAVD '88) values from ELM v2.5.

Note that the development of the elevation and bathymetry maps utilized the NAVD '88 vertical datum, which is the most accurate datum. In order to have the option of making simulations consistent with the SFWMM (which will drive ELM boundary conditions for all future scenarios), we converted the final maps from NAVD '88 to NGVD '29 using "Corpscon", v. 6.0.1. Unpublished results from a comparison of historical (calibration/validation) simulations using the different datums showed no effective differences in the statistical comparisons of observed and simulated stages at the multiple gage locations within the model domain. However, potentially significant differences in flow were observed in regions where the non-linear differences between the two datums (Figure 4.4) resulted in a gradual slope of differences across tens of kilometers.

Figure 4.4. The difference in elevation between the two vertical datums in common use in south Florida projects, comparing the 1 km resolution DEM of ELM v2.5 within the two datums. Note the nonlinear, spatial trend (error) of differences in the NGVD '29-based data relative to the more accurate NAVD '88 based data.



4.4.3 Soils

All of the soil map data were unchanged from ELM v2.5, but were resampled and filtered.

4.4.3.1 Bulk density (input = "soilBD")

ELM v2.5 1-km resolution data, resampled and filtered.

4.4.3.2 Organic bulk density (input = "soil_orgBD")

ELM v2.5 1-km resolution data, resampled and filtered.

4.4.3.3 Total phosphorus concentration (input = "soilTP")

ELM v2.5 1-km resolution data, resampled and filtered.

4.4.4 Vegetation

4.4.4.1 Habitat type (input = "HAB")

ELM v2.5 1-km resolution data, resampled.

4.4.4.2 Macrophyte biomass (input = "icMacBio")

ELM v2.5 1-km resolution data, resampled and filtered.

4.5 Static attributes

4.5.1 Water management infrastructure

4.5.1.1 Canal and levee network (input = "CanalData.chan")

The canals and associated levees (defined in a text data file CanalData.chan) have a spatial topology that uses exact geographic coordinates (UTM zone 17, North American Datum of 1927). Thus, modifying the spatial resolution of the model (and thus the model grid's row/column attributes) does not require modification of the geographic coordinates or other attributes of this input file. However, a small number of coordinate locations (defining a continuous canal vector segment) needed some slight (ca. tens of meters) movement to rectify details of hydrologic basin locations relative to the "basins" map described above.

4.5.1.2 Water control structures (input = "CanalData.struct")

The attributes of all water control structures are maintained in a relational database using "FilemakerPro" software (which exports the model input file, CanalData.struct). Among the definitions in this database are the attributes of sources and destinations of structure flows, which necessarily uses grid cell row-column attributes. The relational database has a range of functionalities, including calculations of grid cell locations for any model scale (grain and extent) using geographic coordinates.

The same water control structures and structure attributes were used in the ELM v2.8 and ELM v2.5.2. The grid cell row-column references were different (via the database geometric calculations) between the models at the different resolutions.

4.5.2 Model parameters

None of these parameters have been updated from ELM v2.5; please see the ELMv2.5 Documentation Report, Chapter 4.

4.5.2.1 Global parameters (input = "GlobalParms_NOM")

None of these parameters have been updated from ELM v2.5; please see the ELMv2.5 Documentation Report, Chapter 4.

4.5.2.2 Habitat-specific parameters (input = "HabParms_NOM")

None of these parameters have been updated from ELM v2.5; please see the ELMv2.5 Documentation Report, Chapter 4.

4.5.2.3 Aquifer hydraulic conductivity (input = "HydrCond")

ELM v2.5 1-km resolution map data, resampled and filtered (see methods above in Initial condition maps).

4.6 Boundary conditions

4.6.1 Meteorological

4.6.1.1 Rain (input = "rain.BIN")

No change from ELM v2.5 (and SFWMM v5.4); please see the ELMv2.5 Documentation Report, Chapter 4.

4.6.1.2 Evapotranspiration (input = "ETp.BIN")

No change from ELM v2.5 (and SFWMM v5.4); please see the ELMv2.5 Documentation Report, Chapter 4.

4.6.2 Hydrologic

4.6.2.1 Flow constraints (input ="BoundCond")

ELM v2.5 1-km resolution map data, resampled (see methods above in Initial condition maps).

4.6.2.2 Stage/depth (input = "BoundCond_stage.BIN")

No changes from ELM v2.5.

4.6.2.3 Tidal height (input = "CanalData.graph")

The tidal stage heights were converted from NGVD '29 to NAVD '88 using "Corpscon", v. 6.0.1. No other changes from ELM v2.5.

4.6.2.4 Managed flows (input = "CanalData.struct_wat")

No changes from ELM v2.5.

4.6.3 Nutrient/constituent inflows

4.6.3.1 Atmospheric phosphorus & chloride deposition

For phosphorus, there were no change from ELM v2.5; please see the ELMv2.5 Documentation Report, Chapter 4.

We added chloride inputs to the model from atmospheric deposition, using a rainfall concentration that was constant in time, at 1.7 mg l^{-1}.

4.6.3.2 Phosphorus in structure inflows (input = "CanalData.struct_TP")

No changes from ELM v2.5.

4.6.3.3 Chloride in structure inflows (input = "CanalData.struct_TS")

No changes from ELM v2.5.

4.7 Performance assessment targets

4.7.1 Hydrologic

4.7.1.1 Stage

The observed stage heights at all gage stations were converted from NGVD '29 to NAVD '88 using "Corpscon", v. 6.0.1. No other changes were made to the observed data that were used in assessing the calibration-validation performance of ELM v2.5. The grid cell row-column coordinates were necessarily changed (from the 1-km grid row-col coordinates) for the new grid resolution of the ELM v2.8, using the ModelOutListCreator.xls (see Chapter 10 User's Guide) to automate the process for all geographic locations.

4.7.2 Water quality

4.7.2.1 Surface water quality constituents

No changes from ELM v2.5. The grid cell row-column coordinates were necessarily changed (from the 1-km grid row-col coordinates) for the new grid resolution of the ELM v2.8, using the ModelOutListCreator.xls (see Chapter 10 User's Guide) to automate the process for all geographic locations.

4.7.3 Ecological

4.7.3.1 Other ecological targets

No changes from ELM v2.5. The grid cell row-column coordinates were necessarily changed (from the 1-km grid row-col coordinates) for the new grid resolution of the ELM v2.8, using the ModelOutListCreator.xls (see Chapter 10 User's Guide) to automate the process for all geographic locations.

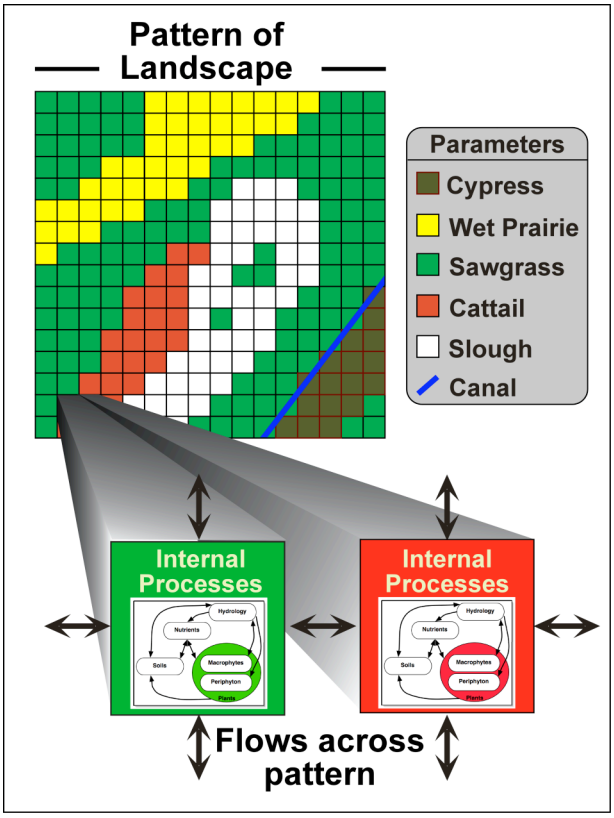
4.8 Literature cited

Desmond, G. 2003. Measuring and Mapping the Topography of the Florida Everglades for Ecosystem Restoration. Open File Report 021-03, U.S. Geological Survey, Reston, VA.

BLANK PAGE

Documentation of the Everglades Landscape Model: ELM v2.8

Chapter 5: Model Structure



http://ecolandmod.ifas.ufl.edu

February 13, 2009

Chapter 5: Model Structure

5.1 Overview

The Everglades Landscape Model (ELM) is a spatially distributed simulation using integrated hydro-ecological process modules. With a structured programming approach, the hydrologic, biogeochemical, and biological processes (such as evapotranspiration, soil oxidation, and plant growth) are contained in code modules that are activated by the user at runtime. Being "data-driven", the model relies on databases to modify scenarios of water management, while computer source code remains constant.

This Chapter on Model Structure for ELM v2.8 serves to update the Model Structure Chapter 5 of the complete ELM v2.5 Documentation Report. Therefore, this is not a "stand-alone" document on the model structure, but simply updates any algorithm which has changed to meet the objectives of different projects.

The major change from ELM v2.5 to ELM v2.8 was the restructuring and refinement of algorithms that defined schedule-based managed flows through water control structures. Integral with the goals and objectives of new modeling projects for evaluating local management alternatives, these modifications allowed evaluations of simple alternatives to hydrologic and water quality management of the landscape.

This Chapter also includes an Appendix on the numerical dispersion attributes of the code, with results. That document was created during the 2006 Independent Peer Review of ELM v2.5, and is included verbatim for this ELM v2.8 update.

A separate User's Guide Chapter includes information on the required computing environment[1] and the basic steps needed to install and use an ELM project.

Using an Open Source[2] philosophy, we hope to encourage collaboration in the modeling community. Towards that end, all source code (and data) necessary for an ELM project is available for download on the ELM web site, and all code in the ELM project is documented in detail using the automated "Doxygen" documentation system. This online, source-code level documentation extends beyond the scientific algorithms described in this Chapter, including details of all of the functions that are compiled in the (ANSI C) code project.

We recommend viewing the hyper-linked version of the algorithm interactions and equations at this location on the ELM development web site (http://ecolandmod.ifas.ufl.edu/models/algorithms_ELM/MainControllerModule.html).

[1] Unix operating system (Linux, Darwin, or Solaris) using Open Source software.

[2] http://www.opensource.org/

5.2 Update summary, ELM v2.5 – v2.8

This Model Structure Chapter 5 describes ONLY changes that were made to algorithms and source code between the regional ELM v2.5 (July 2006, ELM v2.5 Documentation Report[3]) and the current ELM v2.8 (specifically, the public release of ELM v2.8.3). Therefore, this is not a "stand-alone" document on the model structure, but simply updates any algorithm which has changed to meet the objectives of different projects.

Several changes were made to accommodate specific objectives of evaluating local scale management alternatives. As described later in this chapter, the principal changes were made to increase the functionality of the model in simulating managed flows through water control structures. In maintaining its design goals, the ELM v2.8 code remains general in scope, such that a change made to accommodate such new functionality does not affect other applications if that functionality is not needed. Thus, when referring to v2.8.3 of the ELM code, it does not matter whether the model project of interest is a regional or subregional application – the algorithms and code are general to all.

As summarized in Table 5.1, a variety of other modifications were made to the ELM between v2.5 and v2.8. None of these changes resulted in significant differences in the performance characteristics of a regional application, but all provided either enhanced model functionality or incremental improvement to the predictive performance capabilities of the model at subregional and regional scales (see Model Performance, Chapter 6).

The ANSI C language source code of the entire ELM project is fully documented using the automated documentation tool Doxygen[4]. All ELM source code (and requisite data) is available for download from the ELM web site[5], and the Doxygen-generated documentation is available from that web site. This web-based source code documentation is primarily targeted to an audience of programmers, but its easy navigation can be useful to clarify a user's understanding of details of dependencies, methods, etc.

---

[3] Fitz, H.C., and B. Trimble. 2006. Documentation of the Everglades Landscape Model: ELM v2.5. South Florida Water Management District. http://my.sfwmd.gov/elm Reviewed by independent expert panel, reported at http://my.sfwmd.gov/elm 664 pages.

[4] The Open Source Doxygen application is available at http://www.stack.nl/~dimitri/doxygen/

[5] http://ecolandmod.ifas.ufl.edu/models/doxy_ELM.html

Table 5.1. Summary of updates to code for ELM applications, v2.5 through 2.8.

Version	Date	Purpose	Description/*detail*
2.5.2	Jul-06	Public release	Complete documentation, source code, data for regional application
2.6.0	Nov-06	Expand functionality	In response to Peer Review Panel requests, modified input/output utility functions, for greater flexibillity in boundary conditions *a) new data for Ridge&Slough subregional application, century time scales*
2.6.1	Jan-07	Documentation update	Following Peer Review project, misc updates to code and data documentation, for finalizing results of Peer Review project
2.7.a	Jul-07	No code changes	New spatial data, for prototype of new regional application at 500 m grid resolution; improved model-installation methods
2.7.0	Oct-07	Expand functionality; bug fixes	Formalize velocity calculations for sediment transport; enhance multi-grid modeling capabilities *a) increased number of point time series locations that may be output;* *b) corrected stage vs. depth code for overland flows from SFWMM at domain periphery (identified during Peer Review)* *c) corrected code that was intended to "auto-scale" constituent dispersion at different grid resolutions (identified during Peer Review)* *d) option to output surface water flow velocities in grid cells*
2.7.1	Nov-07	Expand functionality	Prototyping for increased flexibility in water management options (designing to be limited in scope/complexity) *a) prototype restructuring of modules for rule-based water control structure flow* *b) option to output grid-cell information from boundary-condition model (e.g., SFWMM)*

Table 5.1 (continued). Summary of updates to code for ELM applications, v2.5 through 2.8.

Version	Date	Purpose	Description/*detail*
2.8.0	Dec-07	No code changes	New land surface elevation map & new vertical datum, for optional use in new regional application at 500 m grid resolution
2.8.1	Feb-08	Expand functionality	Completed update to rule-based water management modules; other extensions to capabilities
			a) increased modularity to support expanded capabilities in triggering rule-based managed flows
			b) added chloride atmospheric deposition equation and supporting dbase change
			c) added option to output new Basin/Indicator-Region file; extended option to output boundary-condition model data (e.g., NSM/SFWMM)
2.8.2	Jul-08	Expand functionality	Additional spatial array (map) output capabilities
			a) added floating point spatial array output options
			b) added self-documenting netCDF spatial array output options
			c) added units to Model.outList (runtime configuration) file, to support self-documenting netCDF format
2.8.3	Feb-09	Public release	Documentation for public release, regional and subregional applications

5.2.1 ELM v2.6

5.2.1.1 Summary

The update from ELM v2.5 to v2.6 was made during the Independent Peer Review (July-Dec 2006) of the regional ELM v2.5. The primary updates involved new source code utilities that increased functionality of boundary conditions, and new supplements to the model documentation that further described model performance under significant "perturbations"[2]. In order to maintain consistency of model results between v2.5 and v2.6, the ELM v2.6 did not involve changes to existing algorithms, with the secondary version attribute having been incremented to v2.6 in order to avoid confusion with the v2.5 public release. There was no new public release of v2.6 code and data.

5.2.1.2 Specifics

During the independent peer review of the ELM v2.5, a variety of requests were made by the review Panelists[3]. To meet one of the requests, source code changes were made to

[2] See http://my.sfwmd.gov/elm , page at the "Implementation: v2.5" tab, in the "Supplemental Results" section.

[3] Mitsch, W. J., L. E. Band, and C. F. Cerco. 2007. Everglades Landscape Model (ELM), Version 2.5: Peer Review Panel Report. Submitted January 3, 2007 to the South Florida Water

increase the number of options for defining boundary condition flows along the periphery of the model grid cell domain. These modifications were targeted towards new applications that were run for century time scales, under hypothetical overland inflow conditions. The modifications were verified to have not affected standard regional or subregional applications of ELM.

Several code bugs were identified (Oct 2006) during this update, and corrections were planned for a subsequent version update (v2.7, after completion of the Peer Review project).

5.2.2 ELM v2.7

5.2.2.1 Summary

In the update from ELM v2.6 to v2.7, two code bugs were corrected, and several refinements made to the model functionality. The update did not include changes to existing algorithms (beyond the bug fixes), but did include several enhancements to model output options and some initial re-structuring of source code involving water control structure flows. There was no new public release of code and data for v2.7, but several subregional and regional applications were developed and refined as part of this interim update.

5.2.2.2 Specifics

The correction to the "auto-scaling" algorithm of constituent dispersion did not affect regional or subregional application results. Before and after the correction, the algorithm returned the intended scale of constituent (i.e., phosphorus and chloride) dispersion in surface water flows[4], provided that the correct dispersion scaling parameter was input to the model. The correction to the code bug was effectively a modification that aligned the code with the original documentation intent, allowing the user to apply the same model parameter file (Global_Parms_NOM) to model applications of any grid resolution, instead of a customized parameter file for each model application grid scale.

The correction to the code bug involving surface water flows across (un-leveed) domain boundaries could have had significant impacts on simulation results, but the overall statistical summaries of the regional application were negligibly affected due to the limited region of such surface water exchanges across the domain boundaries in that application. Such exchanges were generally limited to the Big Cypress National Park (BCNP) subregion, which is an area with low hydroperiods in the historical record. In such areas where overland flows were allowed across (un-leveed) boundaries, the algorithm evaluated differences in water stage elevations between internal and external grid cells, with stage in the external grid cells being derived from SFWMM output. However, (in ELM v2.5), an estimate of the external land surface elevation was not

---

Management District, West Palm Beach, FL. http://my.sfwmd.gov/elm (Peer Review: Comments tab). 35 pp.

[4] Fitz, H.C. Nov 22, 2006. Addendum to: ELM v2.5: Model Structure Chapter 5. http://my.sfwmd.gov/elm , page at the "Implementation: v2.5" tab, in the "Supplemental Results" section.

added to the SFWMM output data, which actually represented positive or negative water depths relative to local (SFWMM) land surface elevation. In ELM v2.5, hydrologic performance was poorest in the BCNP region relative to the other regions in the model domain. These performance characteristics were originally associated with the very high uncertainties that existed in the BCNP land surface elevation data that were used in the model. As noted in the ELM v2.5 Documentation Report, Data Chapter 4, the ELM v2.5 used different sources of data from the SFWMM in BCNP. The correction encoded into the ELM v2.7 simply involved applying the within-domain land surface elevation to the external cell, for an estimate of external stage. Using the original ELM v2.5 land surface elevation for BCNP, hydrologic performance did not improve after the correction to the code bug. However, with data from new land surface elevation surveys in this region (ELM v2.8, below) and correct boundary condition code for SFWMM-driven overland flows across the domain boundary, model evaluations showed significant improvements in hydrologic performance in this BCNP region. Those regional performance characteristics will be documented in the regional ELM v2.8 release.

Other changes made to code for ELM v2.7 involved enhancing the functionality of the model, providing several new variables as options to output. The first variable was that of surface water flow velocities, and another variable was that of the depth of the water table relative to land surface elevation.

5.2.3 ELM v2.8

5.2.3.1 Summary

The principal change between ELM v2.7 and the current ELM v2.8 was the restructuring and refinement of algorithms that defined rule-based managed flows through water control structures. Integral with the goals and objectives of new modeling projects for evaluating local management alternatives, these modifications allowed evaluations of simple alternatives to hydrologic and water quality management of the landscape.

The primary focus of this Model Structure Chapter 5 for the ELM v2.8 is the description of those code changes.

Another change to code (and data) that was made in the v2.8 update was the addition of one equation to represent atmospheric deposition of chloride. In addition, output functionality was enhanced to meet new Performance Measure requirements imposed for evaluating hydrologic changes. Finally, two additional map output capabilities were developed: self-documenting file(s) of netCDF spatial time series, and time series of individual floating point maps. (Prior versions allowed only scaled output of integer-only data).

5.2.3.2 Specifics

The equation for atmospheric deposition of chloride of a similar form as that for phosphorus, which was modified during the v2.6 update conducted during the ELM v2.5 Peer Review. In ELM v2.8, both phosphorus and chloride are (independently) input into the model domain by one of two options selected by the user:

1. Assume a constant concentration in rainfall inputs (wet deposition) to the model

domain, resulting in spatial and temporal variation in constituent loads; or

2. Assume a temporally-constant loading rate of total (wet plus dry) deposition, which may vary in space (via a single input map of deposition).

If the user assigned a negative concentration value to the constituent (chloride or phosphorus) rainfall concentration parameter in the GlobalParms_NOM input parameter file, a domain-wide (constant or spatially variable across space) input map of the long-term daily mean of the mass loading rate was assigned to the (cell-specific) atmospheric load variable for the particular constituent. Otherwise, the non-negative constituent concentration was applied to the daily rainfall volume (in each grid cell), with the mass load assigned to the atmospheric load variable for the particular constituent.

The remainder of this chapter describes the specific changes to algorithms in the Horizontal solutions: Water Management: Structure Flows module. Please see the ELM v2.5 Documentation Report for the current documentation of other code modules, which did not change for ELM v2.8.

5.3 Horizontal solutions (updates)

The horizontal solution modules calculate spatial flows of surface water, groundwater, and associated constituents (phosphorus and salt/tracer) in the (mostly) horizontal dimensions across raster grid cells and vector canals.

For this ELM v2.8 update, only the Water Management: Water Control Structure Flows module descriptions are updated.

See the ELM v2.5 Documentation Report for full descriptions of the other Vertical Solutions and Horizontal Solutions.

5.3.1 Water management: Water control structure flows

5.3.1.1 Overview

The Water Management Modules provide the mechanisms for distributing managed flows of water and constituents (phosphorus and salt/tracer) in a network of canals, levees, and water control structures. The ELM code for quantifying water control structure flows was significantly restructured in the process of the ELM v2.8 update, in order to increase the flexibility and modularity of these water management components. The Water Control Structure Flows set of modules includes eight methods (modules) to quantify water control structure flows, plus one controller module. The method defined for each structure flow module depends on its source-destination relationship, and whether the structure flow is data-driven or calculated internal to the model.

5.3.1.2 Controller Module

The attributes of the water control structures are defined in a relational (FilemakerPro) database, and exported into an ASCII (text) input file for the model. Among the variety of attributes in this database (CanalData.struct, see Data Chapter 4, ELM v2.5

Documentation Report) are the definitions of the source (canal ID or cell ID[5]) and destination (canal ID or cell ID) water and constituent storages. The database also defines whether flows are to be driven by time-series input data or to be calculated in the model.

There are two basic classes of water control structures in the ELM: a) structures that involve regulated flows, emulating "real-world" water management from either ELM calculations of flows, or input data on flows; and b) un-regulated "virtual" structures which are model constructs to support hydrologic assumptions, and which do not correspond to "real world" infrastructure.

Daily water and constituent flows are passed through a water control structure using one of four source-destination relationships: 1) flow from a canal to a canal, 2) flow from a cell to a cell, 3) flow from a canal to a cell, or 4) flow from a cell to a canal. Depending on the nature of the source-destination relationship, and the regulated or unregulated class of structure, one of eight (8) function methods are invoked by the controller.

Table 5.2 provides an overview of the decision matrix that is used by the controller to invoke the required method (module) for a water control structure defined in the input database.

Managed/regulated water control structures (i.e., "real-world" structures) may be either: 1) driven by daily time series flow data that is derived from historical observations or from output from other models such as the SFWMM; 2) driven by ELM-calculated structure flows based on targets of stage from other models such as the SFWMM; or 3) driven by management rules (via stage target "schedules") which determine whether a structure is "open" for flow that is calculated by the ELM.

For any water control structure, external boundary condition flows (of water into or out of the active domain of the model) are fluxes to or from a reserved grid cell location (row 1, column 1) that always denotes a cell that is outside of the active model domain. In the case where the source of the water is outside of the model domain (i.e., "new" water), the values of concentration of constituents (phosphorus and salt/tracer) are defined (in the input attribute file CanalData.struct) by either a temporally-constant value, or a link to a time series of daily values. In the case of a time series of daily concentrations, the data were previously developed from either the output of another model, or from interpolations of observed data (see Data Chapter 4). A structure flow whose source water is internal to the model will always have a constituent concentration that is available from internal model calculations. In all cases, the source water's constituent concentration (mass volume^{-1}) is multiplied by the structure flow (volume) to determine the mass of constituent that is associated with the flow.

[5] The cell ID is the row and column grid location, which is calculated in the database from the geographic coordinates of the structure, and is thus independent of the scale of the model application.

Table 5.2. Decision matrix for invoking different water control structure Modules in the WatMgmt.c code. The database (CanalData.struct) that is input to the model defines these attributes for each water control structure. Headwater and Tailwater denote the source and destination, respectively, of positive flows. Canals are always internal to the model domain; a grid cell external to the model domain is denoted by "CellExt". "Use Flow Data" indicates an available time series of daily flows that is input to the ELM (in lieu of internally-calculated flows). The Headwater or Tailwater Targets are stage data targets from either Regulation Schedules or other model outputs. A Trigger Cell may be either an internal or external cell (CellExt) location that is distant from the structure, evaluating stage at that location to "trigger" structure operations. A non-zero Flow Parameter drives the "generic pump" flux equation for calculated flows.

Headwater	Tailwater	Use Flow Data?	Headwater Target	Tailwater Target	Headwater Trigger Cell	Tailwater Trigger Cell	Flow Parameter	Module Invoked (8 total)	Description	Typical usage
Canal	Canal	yes						flowData_CanCan	Data-driven mgmt; internal only	Historical- or SFWMM- driven simulations
Canal	Cell	yes						flowData_CanCel	Data-driven mgmt; internal, or outflow from domain	Historical- or SFWMM- driven simulations
Cell	Canal	yes						flowData_CelCan	Data-driven mgmt; internal, or inflow to domain	Historical- or SFWMM- driven simulations
Cell	Cell	yes						flowData_CelCel	Data-driven mgmt; internal, or in/outflow to/from domain	Historical- or SFWMM- driven simulations
Canal	Canal						0	flowCalc_CanCan	Virtual structure (unregulated): "Instantly" equilibrate Canal-Canal stages; internal only	Connects multiple model canal reaches to represent one long "real world" canal.
Canal	Canal		Schedule	Schedule	Cell	Cell	>0	flowCalc_CanCan	Internal rule-based mgmt, minimize both Schedule-Cell stage differences	**New v2.8, within-basin water flows**
Canal	CellExt			OtherModel		CellExt	>0	flowCalc_CanCel	Outflow from domain, minimize Canal-CellExt (other model data) stage difference	Seepage management
Canal	CellExt			Schedule			>0	flowCalc_CanCel	Outflow from domain, minimize Schedule-Canal stage difference	Creek (vector) tidal boundary conditions
Canal	CellExt		Schedule		Cell		>0	flowCalc_CanCel	Outflow from domain, minimize Schedule-Cell stage difference	**New v2.8, managed outflows**
CellExt	Canal		Schedule				>0	flowCalc_CelCan	Inflow to domain, minimize Schedule-Canal stage difference	Creek (vector) tidal boundary conditions
CellExt	Canal			Schedule		Cell	>0	flowCalc_CelCan	Inflow to domain, minimize Schedule-Cell stage difference	**New v2.8, managed inflows**
Cell	Cell						0	flowCalc_CelCel	Virtual structure (unregulated): Manning's equation of overland marsh flow	Marsh flows thru levee gap (e.g., under I-75 Alligator Alley bridges)

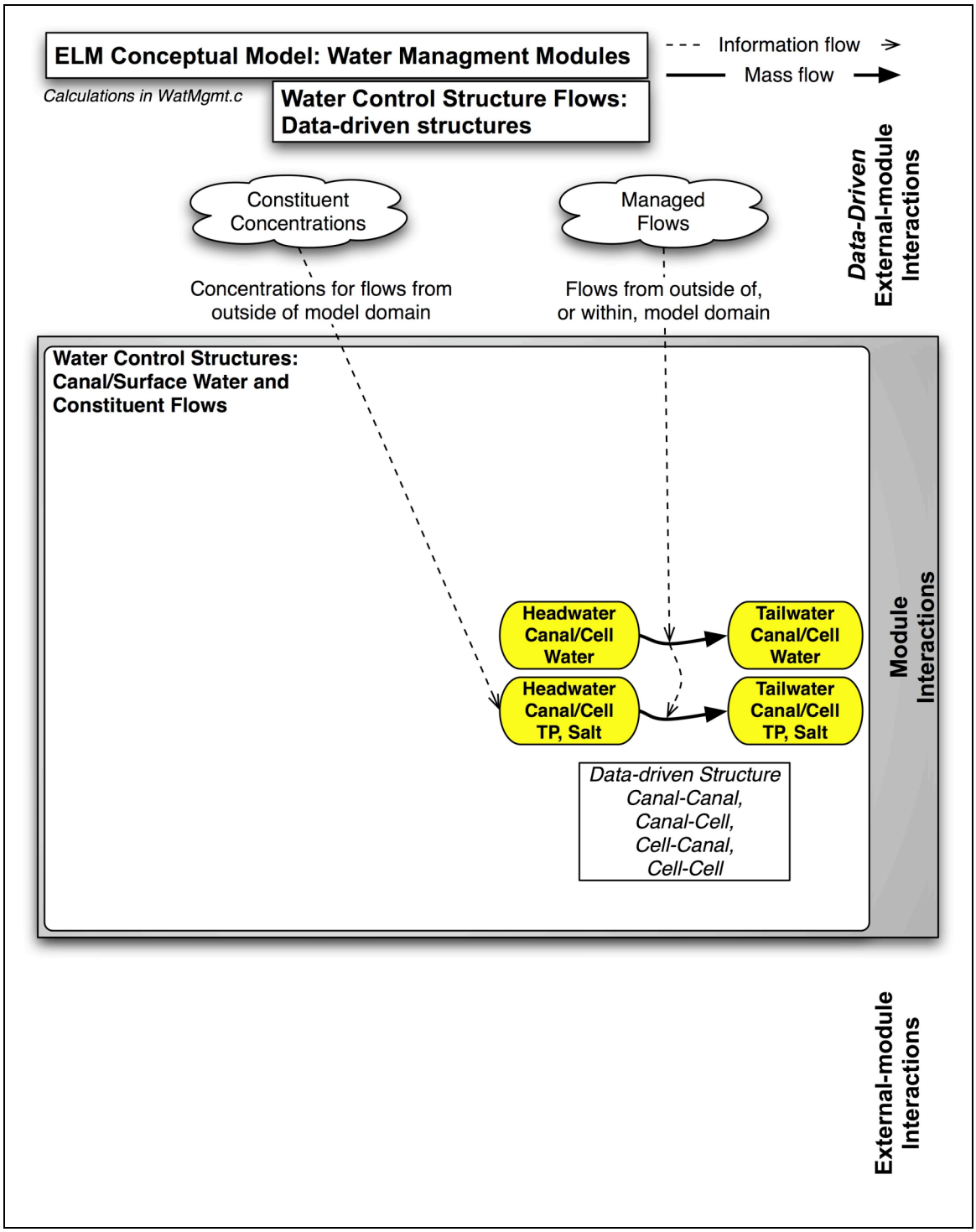
5.3.1.3 Data-driven structures

For data-driven structures, no changes were made to the methods in ELM v2.5 – v2.8.

For data-driven structure flows (Figure 5.1), external data sources (such as historical observations, or SFWMM output) are used for the daily flow values. Such flows may apply to structures which have both the source and the destination within the model domain, or to flows with either the source or the destination being external to the model domain.

Dependent on the (four) source-destination relationships, there are four modules that define water control structure flows that are "data-driven", for which the flows are not calculated by the model. In these modules (Table 5.2), the daily flow values (adjusted for the canal time step) from data sources (such as historical observations, or SFWMM output) are directly assigned to the model flow variable after being converted from English units (cfs, or cubic feet per second) to metric units ($m^3\ d^{-1}$). Another computation that is made is an evaluation of any source-volume constraint. If the data-driven flow demand exceeds the source-volume, the flow is reduced to the volume that is defined to be available in the source. In such a case, a warning is printed to a debugging output file (Driver1.out).

Figure 5.1. Data-driven water control structures. Daily flows are input from data sources such as historical observations, or another model's forecasts for future managed flows (i.e., SFWMM). For a structure that introduces "new" water from outside of the ELM domain, input data are used to assign concentrations of (phosphorus and chloride) constituents to the flow.



5.3.1.4 Virtual structures

For virtual structures, no changes were made to the methods in ELM v2.5 – v2.8.

As indicated in the Water Management Canal-Marsh Flux Module section (see ELM v2.5 Documentation Report), because some canals extend over large distances, the model segments a number of "real world" Everglades canals into separate model canal reaches that are linked by "virtual" water control structures which equilibrate the stages in the two canal reaches at every canal time step (Figure 5.2). This segmentation minimizes the potential grid-cell dispersion of constituents (nutrients and salt/tracer) along canals spanning long distances, as constituents are assumed to be homogenous along the entire length of a canal reach.

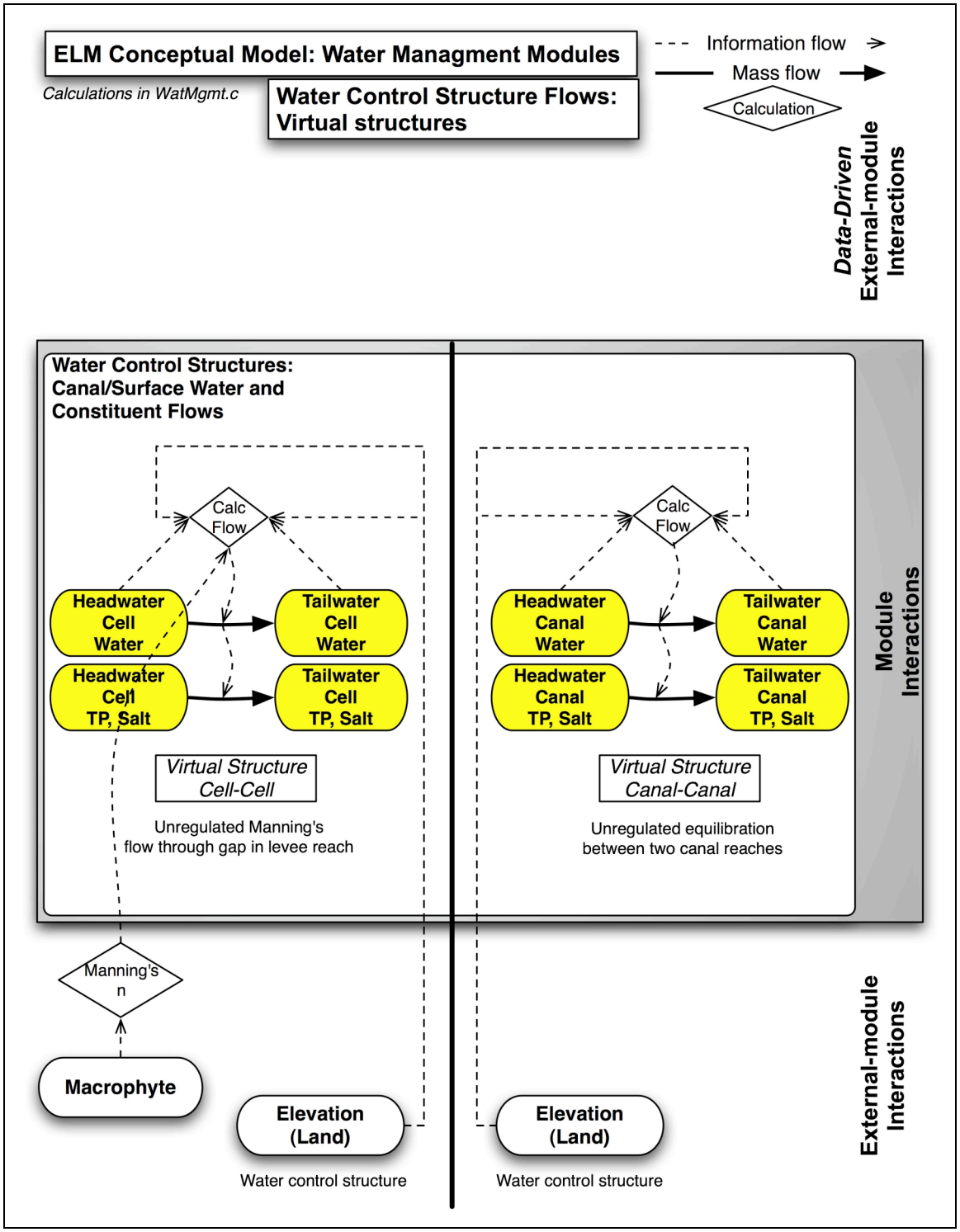
In the case of "virtual" structures that equilibrate two canal reaches (that are portions of a longer, continuous "real-world" canal), a simple mass-balance equilibrium is sought between the two canal reaches during each canal time step:

$$flux = \frac{A_s \cdot A_d}{A_s + A_d} H_{delta}$$

where *flux* is the flow volume (m^3) during a canal time step, A_s and A_d are the surface areas (m^2) of the source and destination canal reaches, respectively, and H_{delta} is the head difference (m) between the two canal reaches. This difference is taken to be the difference in stage elevations at the midpoints of the two reaches, with a constant depth assumed along the length of the reach. For each canal reach, the elevation drop along the length of the reach from the upstream end to downstream end is known from the initialization of the canal network topology. To obtain stage elevations, the depth (m) of water stored in each canal reach is added to the land surface elevation at the midpoint each canal reach: stages based on those elevations are equilibrated at every time step (in the positive downstream direction only).

In the case of an under-bridge "virtual" structure between wetland grid cells (Figure 5.2), the overland flow equation for grid cell fluxes is called to calculate the overland flow using an open-water Manning's n coefficient (see Surface Water Raster Flux Module for module and equation descriptions, ELM v2.5 Documentation Report).

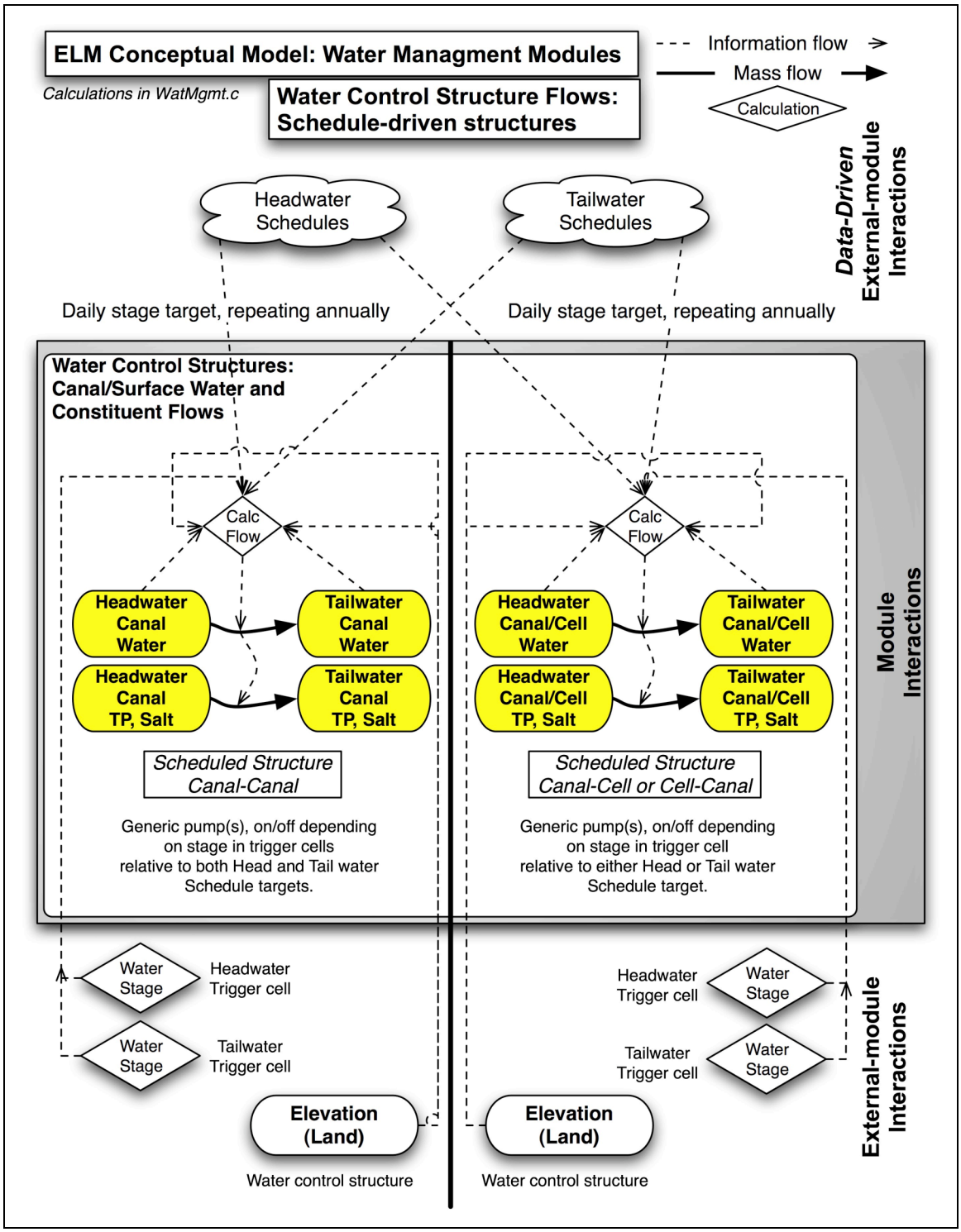
Figure 5.2. Virtual water control structures. Flows are calculated by the model to equilibrate stages between two canal reaches (right), or to provide a method for calculating overland marsh flows through a gap in a levee (left).



5.3.1.5 Schedule-driven structures

For ELM v2.8, a variety of new methods were developed for regulated, schedule-driven water control structure flows (Figure 5.3). While the new methods were associated with source code changes, the structure (fields per record) of the input database (CanalData.struct) required no associated changes.

Figure 5.3. Schedule-driven water control structures. Flows are calculated by the model to regulate water stage elevations in the marsh (grid cells), with the goal of minimizing the difference between the schedule target(s) and the trigger grid cell(s). Stage in the trigger cell(s) is (are) evaluated relative to the target stage. If the target is not met, a generic pump is invoked to move water between the head- and tail- water (source and destination) storage locations, with either a canal-canal, canal-cell, or cell-canal flow relationship.



Tidal boundary conditions

In ELM v2.5 – v2.8, tidal boundary conditions[6] were imposed with a "schedule"-driven head or tail water target stage for structures[7] that associated with vectors of tidal rivers/creeks (aka "canals") and cells external to the model domain. Long-term mean (Jan – Dec) monthly tidal stages recurred annually through use of a 12-month input graph function, and the data were interpolated to provide daily head- or tail- water target stages in an external source or destination grid cell associated with the creek/river vector (see Table 5.2).

A potential flux was calculated from the stage difference between the external (cell) schedule target and the internal river/creek vector, moving water and constituents between the source and destination (cell-canal, or canal-cell flow relationship). See the below section for the equation definitions that were common to these schedule-driven structures. If the source water was an external cell, a constant salinity that was input by the user was imposed on each tidal flux. As with any "canal" vector, creek/river vectors were segmented and linked by canal-canal (creek-creek) virtual structures as described above.

Managed flows

In ELM v2.8, the new water control structure methods managed water relative to the operational requirements dictated by schedule-driven head- and/or tail- water target stages relative to marsh stages in "trigger" cells (Figure 5.3). Coincident with the ecological goals of ELM applications, the water management algorithms are kept very simple, avoiding *any* level of design engineering for the infrastructure associated with "real world" water management in the Everglades. Therefore, the methods used in ELM water management are considered to be idealized "water movers" , and assume that engineering constraints and capacities of water control structures can be formally quantified with other (simulation and/or analytic) quantitative tools. The goals of water management methods in ELM are to simply move water in response to commonly-used schedules and triggers, but ignore possible hydraulic constraints associated with moving volumes of water between remote regions.

There were three managed water control structures that were encoded for ELM v2.8:

1) managed losses from the system, fluxing water and constituents from a canal to an external cell (canal-cell);
2) managed gains to the system, fluxing water from an external cell into a canal (cell-canal); and
3) managed flows within the system (of a single hydrologic basin).

---

[6] Not applicable to subregional applications lacking connections to an estuary.

[7] These tidal structures are "virtual" in that they are model constructs in order to most simply provide time-varying tidal boundary conditions. These implementations of schedule-driven "structures" do not correspond to "real world" infrastructure. However, for ease of categorization for this documentation, we consider them under the schedule-driven structure category. No changes in tidal boundary condition methods were made between ELM v2.5 and v2.8.

In a method analogous to that of the tidal boundary conditions, long-term mean (Jan – Dec) monthly target stages recurred annually through use of a 12-month input graph function (see Model Application Chapter 8 for data examples), and the data were interpolated to provide daily head- or tail- water target stages in internal and/or external source or destination grid cells (Table 5.2).

For structures with either a canal-cell or a cell-canal relationship of managed flows, an evaluation was made of the stage difference between the scheduled target stage and the trigger cell in the marsh. In the case of canal-cell managed losses from the system, if the stage in the marsh trigger cell exceeded that of the headwater target, the structure was classified as open, to provide inflow of water into the system. Similarly, in the case of cell-canal managed inputs to the system, if the headwater target stage exceeded that of the marsh trigger cell, the structure was classified as open, to provide inflow of water into the system

In all cases, the objective function of the water management structure was that of minimizing the difference between the schedule's target stage and the stage in the remote marsh trigger cell. The water control structures in these cases were assumed to be an idealized, generic pump (or set of pumps), with variable RPM that decreased with decreasing difference between the existing (remote trigger cell) and targeted water levels:

$$flux = Q_{\max} \cdot H_{N\_delta} \cdot dt_{can}$$

where *flux* is the potential flow volume (m^3) during a canal time step, Q_{max} is the maximum pump capacity ($m^3\ d^{-1}$), H_{N_delta} is the normalized head difference ($m\ m^{-1}$) between the target and existing stages, normalized to a 1-m deficit at maximum pump RPM. The actual flow volume (m^3) during a canal time step was constrained to not exceed the maximum available volume that was defined for the source water storage. This simple equation of potential *flux* ignores the engineering constraints of the head differential between the source and destination storages, assuming only that the idealized pump will increase in throughput as a linear function of the management demand.

5.4 Appendix: Numerical Dispersion

This Appendix is a document that was created during the 2006 Independent Peer Review of ELM v2.5, and is included verbatim for this ELM v2.8 update (updating the ELM v2.5 Documentation Report).

5.7.3 Overland flow module [addendum]

Surface Water Raster Flux Module Description

ELM v2.5 Documentation Report (July 2006). p. 5-90. Starting with paragraph #2 of section:

[paragraph #2] The flow between two adjacent cells is determined from a simplification of the well-known open channel, diffusion flow model in an explicit, finite-difference framework. Omitting any inertial or acceleration terms, the continuity equation is simply a two-dimensional flux driven by differences in slope of the water surfaces. The flux between a pair of grid cells in the model domain's array is described by the empirical Manning's equation for overland flow:

$$Q = \frac{D^{\frac{5}{3}} L^{\frac{1}{2}} \Delta h^{\frac{1}{2}}}{n} \qquad \text{Eqn 1}$$

where Q is the volumetric flow velocity ($m^3\ d^{-1}$), D is the water depth (= hydraulic radius, m) above ground elevation, L is the length of a grid cell (m), Δh is the difference (m) in water stage between the source and destination cells, and n is the empirically-derived Manning's roughness coefficient. Using an explicit numerical method, the solution is iterated in both the row-wise and the column-wise directions during each time step, the direction alternates (east-west and west-east, north-south and south-north) after each time step. This Alternating Direction Explicit solution minimizes the directional bias that is associated with a uniform- direction solution. Constraints for stability and mass balance are imposed on the calculated flux during each time step, preventing head reversals or flows greater than the volume available in the donor grid cell. The mass of constituents (nutrients, salt/tracer) is passed along in a mass-balance calculation based upon the water volume flux between cells.

[Dispersion: new text starts below]

Calculations of dispersive flux of constituents takes advantage of the properties of the numerical dispersion that is a known property of finite difference solutions. The approach is analogous to that described for the WASP water quality model (Wool et al. in press), in which the advection term of the water flows is adjusted to decrease or increase the dispersive flux of constituents, for a resultant combined advective and dispersive mass transfer among grid cells.

The (horizontal) flow velocity u ($m\ d^{-1}$) of water among grid cells is determined by:

$$u = \frac{Q}{L \cdot D} \qquad \text{Eqn 2}$$

where Q, L, and D are given previously (and $L \cdot D$ is the interfacial area of flow).

Numerical dispersion associated with the solution method (reference in Wool et al. in press) is calculated by:

$$disp_{num} = 0.5 \cdot u \cdot (L - u \cdot sfstep) \quad \text{Eqn 3}$$

where *sfstep* is the horizontal solution's time step (days). Numerical dispersion is a non-linear function of velocity, and increases with increasing grid size, while decreasing with longer time steps. Using this equation, numerical guidelines for selection of the *sfstep* in the expected velocity regimes of the Everglades[8] were demonstrated in Figure 7.5.1 of the ELM v2.5 documentation: the *sfstep* is chosen for each scale of application to maintain a similar trade-off between decreased numerical dispersion and increased Courant λ (with theoretical instabilities in the solution when $\lambda > 1.0$) for the Everglades applications.

The estimate of numerical dispersion is then used to adjust the velocity term. In this step, the numerical dispersion component of potential constituent flux is removed by:

$$u_{adj} = \frac{(u \cdot L - disp_{num})}{L} \quad \text{Eqn 4}$$

such that u_{adj} is the velocity (m d^{-1}) adjusted to represent that associated with potential advection of constituents, without the influence of potential numerical dispersion. (Note that the transfer of water volume/mass is not affected by any part of these dispersive flux calculations).

The u_{adj} is then put into the form of a water volume potential:

$$Flux_{adj} = Parm_{agg} \cdot u_{adj} \cdot D \cdot L \quad \text{Eqn 5}$$

where $Flux_{adj}$ is the volume (m^3 d^{-1}) of potential water flow that is specific to advective transfer of constituents, and $Parm_{agg}$ is a positive or negative, dimensionless parameter that includes the dispersion number, and a grid scale conversion[9]. This parameter is calculated by:

$$Parm_{agg} = \left(1 - \frac{l_{disp}}{L}\right) dispParm \quad \text{Eqn 6}$$

---

[8] Generally << 5 cm sec^{-1}, as discussed in the ELM v2.5 Documentation Report, Uncertainty Chapter 7.

[9] A bug was found in the implementation of this aggregated parameter in ELM v2.5, resulting in the definition of the use of the l_{disp} (GP_dispLenRef) parameter in the GlobalParms database to be incorrect. The aggregated parameter does not "auto-scale" the dispersion among different grid applications, which was the original intent. The actual implementation of this parameter in ELM v2.5 leads the model to scale the amount of dispersion as the ratio of the GP_dispLenRef length to the grid cell length, as described in this section. The amount of dispersion in the regional (1km grid length) ELM v2.5 (Model Performance Chapter 6) was of the intended magnitude, reducing the potential numerical dispersion in the 1km grid. This bug was found during the quality-control tests of the multi-grid scale implementations of the century-scale perturbation experiments, during the 2006 ELM Peer Review.

where l_{disp} is the dispersion mixing length (m), and *dispParm* is a calibration parameter (set to 1.0 in ELM v2.5).

In determining the actual mass of constituent to flux from cell to cell, the total Flux volume is compared to that representing advective transfer of constituents:

$$P = \frac{(Q - Flux_{adj})}{D \cdot L \cdot L} \cdot sfstep \qquad \text{Eqn 7}$$

where P (dimensionless) is the proportion of the total available (donor) water volume that will be associated with constituent flux. The available (donor) constituent mass is multiplied by that proportion for cell-cell flux, thus completing the constituent mass advection and dispersion flux for a time step. Note that if the $Parm_{agg}$ is equal to 0.0, no correction to numerical dispersion occurs; a negative $Parm_{agg}$ increases the simulated dispersion, while a positive parameter value decreases simulated dispersion.

To demonstrate the use of the parameters to scale the magnitude of actual dispersion in ELM applications, we used a simple subregional domain for a 250 m and a 1 km grid application[10]. A series of Indicator Regions (Figure 1a) were established along a gradient of decreasing elevation (NorthNW to SouthSE), in order to monitor the mass of a conservative tracer along the primary flow path of the simulated system(s). (The mass of the tracer was summed within the multiple grid cells of each Indicator Region). In both applications, the system was inoculated with an initial tracer concentration in the surface water of a 1 km^2 region near the high-elevation (northern) boundary (Figure 1b). With no (northern) water inflows to the domains, the systems had significant water outflows in the downslope, SouthSE section of the domain, inducing the landscape flows. Flows of the tracer mass were monitored on a daily basis as they transited through the Indicator Regions and exited the model domain.

The l_{disp} was set to ½ of the grid cell length in both applications (500 and 125 m for the 1 km and 250 m applications, respectively), to match the dispersion length used in the regional (1 km grid) ELM v2.5 application. (Note that, as explained above, the parameter definition incorrectly implies a dispersion mixing length; to strive for consistency/clarity, we maintained the terminology as defined in the ELM v2.5 documentation). The simulation was run for three months. At the end of the simulation, 41.1% of the original mass of tracer remained in the 250 m grid domain, while 47.9% remained in the 1 km grid domain. Given that the fine scale application was 16x finer resolution relative to the 1 km application, this difference of approximately 7% was indicative of effective control of numerical dispersion that will be present to some degree in any "large" grid. In the 1 km grid application, no adjustment for numerical dispersion (l_{disp} = grid width) resulted in 22.1% of the original mass remaining (or approximately half that under the lower dispersion implementation).

For the finer-scale, 250 m application, we performed a series of model experiments in which we modified the l_{disp} parameter to double- and half- that of the 250 m grid length

[10] Further information on the data behind this subregional application is provided in the Perturbation Experiments Chapter 11 (November 2006 addendum to July 2006 ELM Documentation Report).

(and width), in order to demonstrate the effect of altering the simulated dispersive flux. Water velocity varied in time and space in the model, but was on the order of 0.5-1.0 cm/sec, representing moderately high velocities for the present day Everglades landscape. In comparing the model experiments, the time at which the maximum tracer mass was found in each Indicator region was used as a quantitative indicator of the difference in dispersion under each parameter set (Figure 2). The time for the maximum mass to be attained in Indicator Region 16 was 17 d, 25 d, and 34 d for the l_{disp}= 500, 250, and 125 m, respectively. Thus, if the magnitude of actual dispersion becomes better understood (quantified) in these vegetated wetlands (see Uncertainty Chapter 11), the dispersion fluxes in ELM can be further refined through appropriate parameter adjustments.

Literature cited

[includes only the references used in this Addendum]

Wool, T. A., R. B. Ambrose, J. L. Martin, and E. A. Comer. *in press*. Water Quality Analysis Simulation Program (WASP) Version 6.0 Draft: User's Manual. US Environmental Protection Agency - Region 4, Atlanta, GA

Figures

Two figures follow.

Figure 1. The model experiments were conducted in a 10x10 km subregion (in common with the Chapter 11 Model Perturbation experiments). The Indicator regions (a) used to monitor the tracer mass followed the downslope elevation gradient. A 1 km^2 area (b) was inoculated with an initial concentration (green-yellow) of conservative tracer in the ponded surface water. A snapshot of the (yellow) surface water tracer concentration (c) in the 250 m and 1 km grid applications after 25 days of simulation provides a visualization of the relative differences in dispersal at the two grid scales.

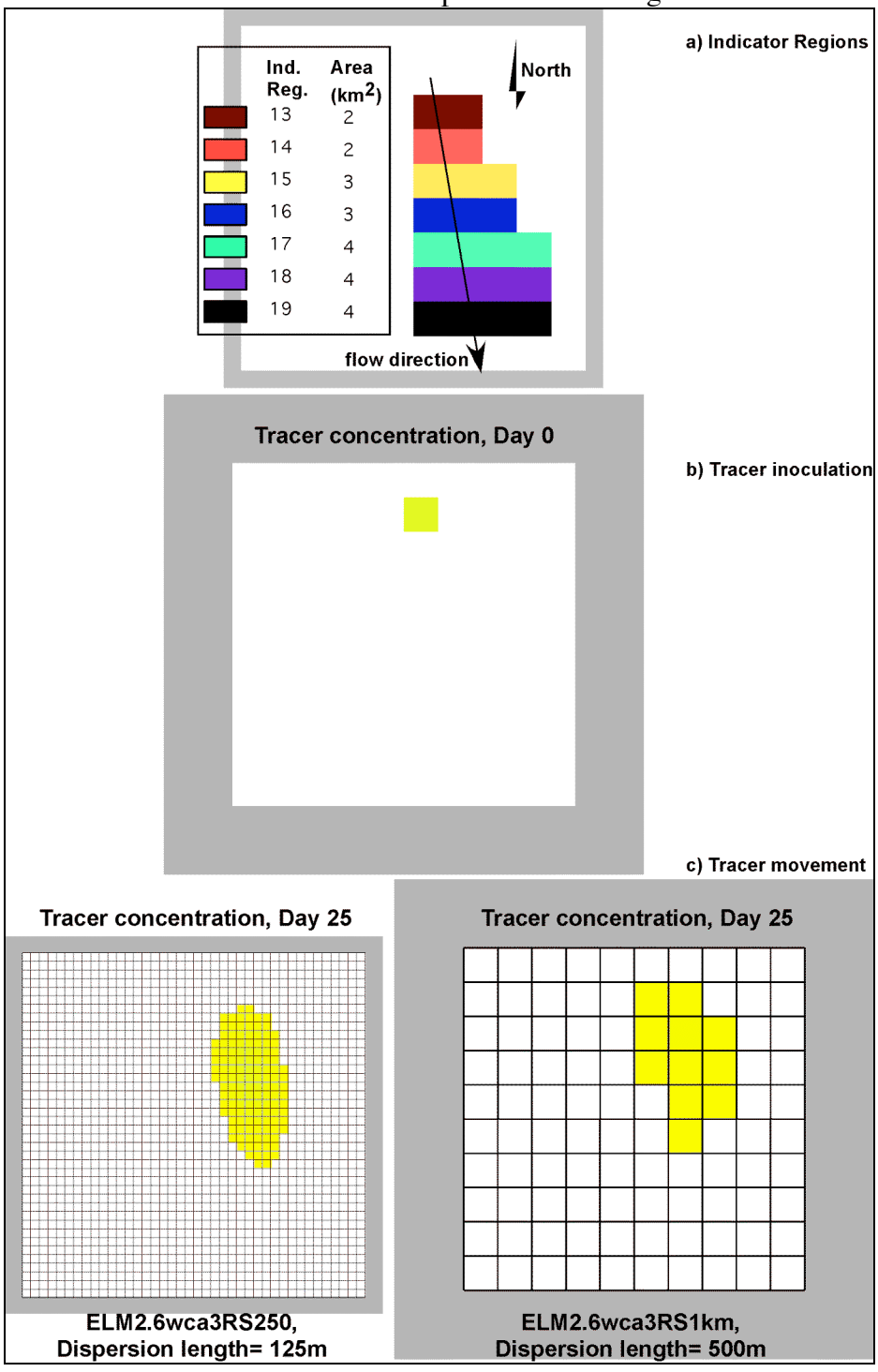
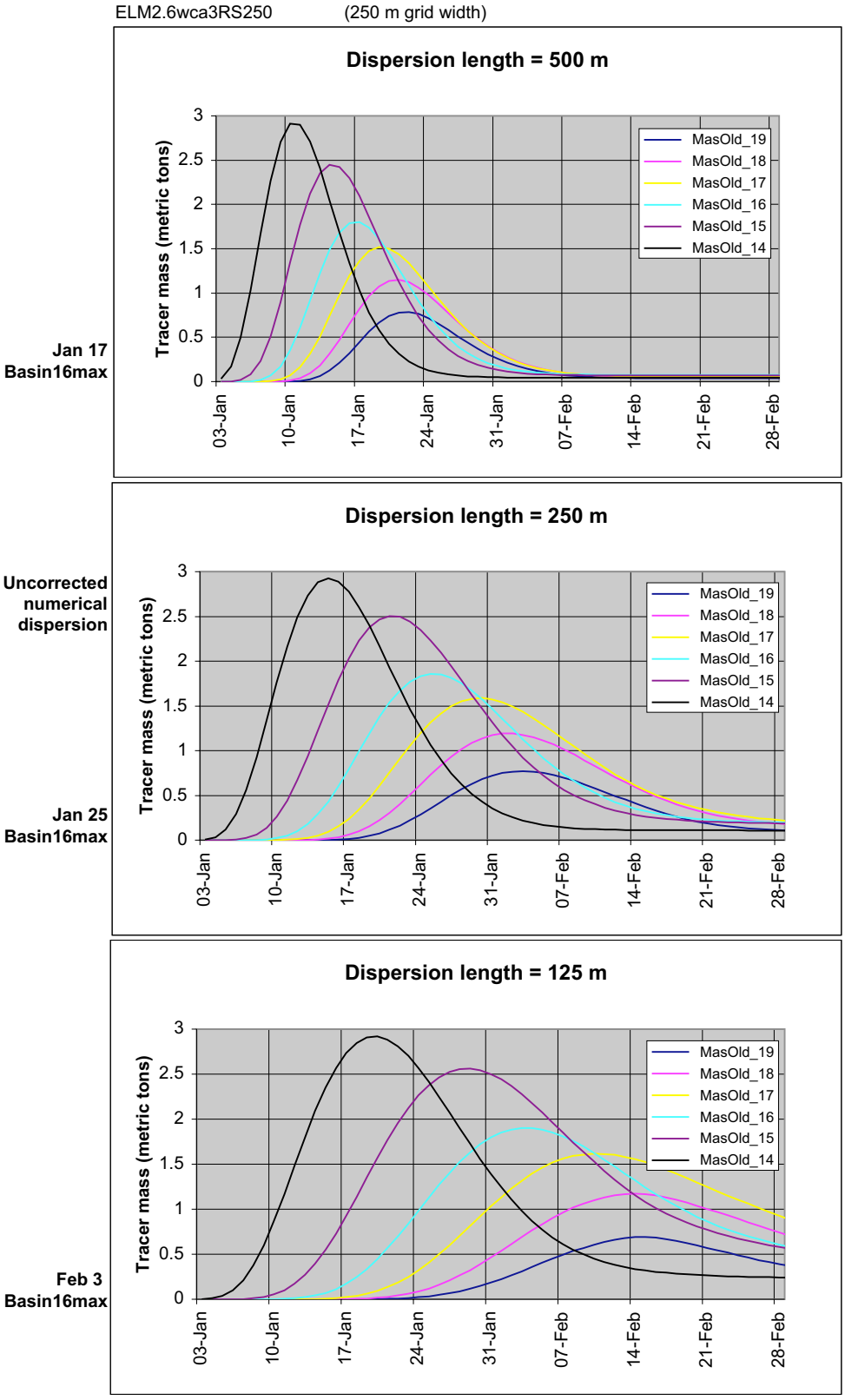


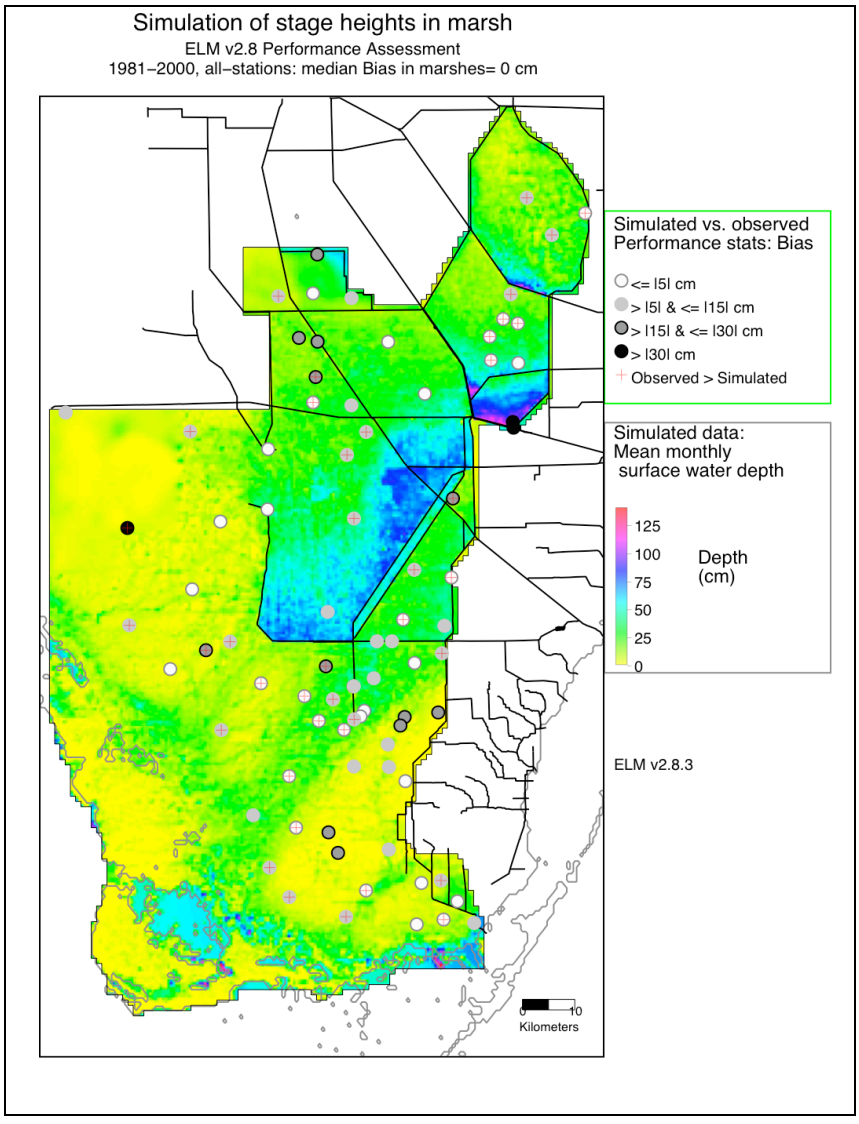
Figure 2. Time series plots of tracer mass in each Indicator Region, for three different values of the "dispersion length" parameter in the 250 m grid application. See the text and Figure 1 for details.



BLANK PAGE

Documentation of the Everglades Landscape Model: ELM v2.8

Chapter 6: Model Performance



http://ecolandmod.ifas.ufl.edu

February 13, 2009

Chapter 6: Model Performance

6.1 Overview

As described in the Introduction Chapter 1 of the ELM v2.5 Documentation, an overarching Goal of the ELM is to understand and predict ecological dynamics across the greater Everglades landscape. For the current ELM v2.8, the primary goal was to provide fine scale (500 m grid resolution) hydrologic output for use in driving other ecological models (see the Executive Summary of this documentation report). For this specific project under the auspices of Joint Ecosystem Modeling partnership, the ecological performance measures were of secondary importance. For future water quality/ecological applications, specific Objectives involve Performance Measures of the "water quality" and other ecological aspects of ecosystem dynamics across the landscape.

The overall approach of (developing and) calibrating the ELM was described in Chapter 6 of the ELM v2.5 Documentation Report. In this update to ELM v2.8, the only parameter changes involved the segmentation of the canal reach vectors; there was relatively little effort involved in "tweaking" the calibration/validation performance. However, a number of code changes (Chapter 5 of this document) and initial condition data (primarily land elevation, Chapter 4 of this document) were made for this update to ELM v2.8.

In its regional (~10,000 km^2) application at 0.25 km^2 grid resolution, the current ELM version 2.8 is available to assess relative differences in hydro-ecological performance of Everglades water management plans - at decadal time scales. Hydrologic performance of the ELM improved over that of ELM v2.5, with a median Nash- Sutcliffe Efficiency statistic was 0.61 and median bias of 0 cm, comparable to the South Florida Water Management Model (SFWMM) within the Everglades. The chloride tracer is an indicator of how well flows were simulated relative to observed data: the median seasonal relative bias of all stations was 10% in the marshes, and 11% in canals (and the bias was 6 mg L^{-1} in the marsh and 13 mg L^{-1} in canals), again showing improvement over the ELM v2.5. While not a focus of this project, the simulated and observed phosphorus concentrations were very closely related, with a bias of 0 ppb in the marsh and canal locations (which again was an improvement over that of ELM v2.5).

Thus, the model "skill" in predicting stage and flow was improved over earlier ELM versions, and continued to be consistent with the SFWMM output. Of particular interest with respect to "driving" fine scale ecological models, this scale of ELM hydrologic output exhibited detailed spatial patterns of flow, with improved connectivity among and within habitats (such as sloughs) relative to the 4x (ELM v2.5) or ~40x (SFWMM v5.4) coarser-scale resolution hydrologic models previously available for the greater Everglades region. Making use of a model with a "native" (original) resolution of 0.25 km^2 may likely lead to improved realism in modeling animal communities which respond to hydrologic patterns at these relatively fine scales.

6.2 Performance expectations

6.2.1 Model application niches

For model users and stakeholders, a fundamental concern is simply: how well does the model work? To be useful, it is critical that model goals and objectives are clearly stated, and that the design and performance of the model is shown to meet those goals. Towards this end, it is critical that a model is understood within the context of its "application niche". The application niche should be a juxtaposition of A) the real or perceived needs of the "users" and B) the realistic capabilities portrayed by the model developers. The intersection of A & B is the intended target of the model application – a basic point that is sometimes lost in practice as a result of inadequate communication.

6.2.2 ELM v2.8 application niche

The ELM application niche was described in Chapter 6 of the ELM v2.5 Documentation Report. The application niche remains essentially the same for ELM v2.8, but the emphasis in this Chapter is on the hydrologic performance, for use in driving other ecological models. This does not reflect a "new" application niche for ELM. Rather, the hydrologic emphasis reflects a need in the scientific community for another method to obtain fine scale hydrologic data to support other ecological models. The SFWMM continues to be the only regional model for simulating water management alternatives, and continues to be required[1] to drive the regional ELM in any future scenarios of water management. The overall goals (application niche) of the ELM remain in the ecological evaluations, with hydrologic performance being a fundamental component of that.

6.2.3 Establishing performance expectations

6.2.3.1 ELM

The expectations of hydrologic simulations in the Everglades are reasonably well-understood by most users. Perhaps this is largely due to the context of hydrologic modeling in south Florida, which has a multi-decadal history of applications, with a relatively well monitored system in which the physics are reasonably well understood.

6.2.3.2 Other Everglades hydrologic models

The South Florida Water Management Model (SFWMM, sometimes referred to as the "2x2") is the primary tool used to evaluate managed hydrology in the south Florida landscape, including the greater Everglades region. This model was used to evaluate relative hydrologic benefits under different water management alternatives for the Comprehensive Everglades Restoration Plan, in addition to a wide variety of other planning applications. The two-mile by two-mile square (~10.4 km^2) grid of the

[1] While historical (calibration/validation) ELM simulations primarily utilize observed flows through water control structures, any regional ELM simulations of future water management scenarios require output from the SFWMM: daily flows through all water control structures, and stage heights along the ELM domain periphery. See Chapter 10 User's Guide.

SFWMM has a relative accuracy in predicting stage that has been well-accepted for evaluating water management alternatives for the greater Everglades and much of south Florida in general. The documentation for the SFWMM v5.5 is available at:

http://www.sfwmd.gov/ (click on "What we do", "Simulation modeling", "Models")

which includes statistical evaluations of the model performance in predicting stage in the greater Everglades. For the 82 marsh stage monitoring locations common to the ELM domain, the statistical comparisons of SFWMM daily output data to daily observed data indicated very good performance, as indicated by the median values for each statistic: R^2 = 0.81, Nash-Sutcliffe Efficiency = 0.67, Root Mean Square Error = 0.12 m, and Bias = 0.0 m. The computational methods used in these statistics are the same as those defined later in this chapter.

As a "second generation" simulator of managed hydrology in south Florida, the South Florida Regional Simulation Model (SFRSM,

http://www.sfwmd.gov/ (click on "What we do", "Simulation modeling", "Models")

is designed to have significantly increased flexibility and performance relative to the current SFWMM. While portions of the SFRSM are still under development, its advanced design, and the very good performance of early prototypes, indicate that it will provide significant improvements as a replacement for the SFWMM in the future.

6.3 Performance evaluation methods

The methods of evaluating and improving the performance of a distributed, integrated ecological model are wide ranging, usually involving both analytic tools and science-based judgments. Ultimately one seeks to communicate the cumulative evidence of how well the model meets its objectives: an evaluation of the model performance in history-matching is a fundamental component of that communication.

Please see Chapter 6 of the ELM v2.5 Documentation Report for the discussion of the calibration process, validation process, and performance evaluation methods (pp. 6-5 through 6-14). The statistical metrics used in evaluating model performance are repeated in this ELM v2.8 chapter as Appendix A.

6.4 Model updates

As described in other Chapters, the current release[2] ELM v2.8 has a number of improvements over the last release, ELM v2.5. For the source code (Chapter 5), several changes were made to accommodate specific objectives of evaluating local scale management alternatives. As described in that chapter, the principal changes were made to increase the functionality of the model in simulating managed flows through water control structures. In maintaining its design goals, the ELM v2.8 code remains general in scope, such that a change made to accommodate such new functionality does not affect other applications if that functionality is not needed. Thus, when referring to v2.8 of the

[2] For simplicity, any full public release version is denoted only by the primary and secondary version attributes (see Model Refinement Chapter, ELM v2.5 Documentation). The tertiary version attribute of this model release is ELM v2.8.3.

ELM code, it does not matter whether the model project of interest is a regional or subregional application – the algorithms and code are general to all.

The new fine resolution regional model (ELM v2.8) encompasses a domain identical to the regional ELM v2.5 (10,394 km^2), but with 42,576, 0.25 km^2, active grid cells (four times the 10,394 grid cells in the 1 km^2 resolution version). For this ELM v2.8 regional application, most of the data (Chapter 4) remain the same as those used for the ELM v2.5 regional application. The principal changes involved "resampling" data from the 1 km resolution map inputs, and generating new spatial interpolations of the updated land surface elevation data at the 500 m resolution.

6.5 Model configuration

In ELM v2.8, the model was configured to simulate historical conditions inclusive of the years 1981 – 2000. The domain was that of the regional ELM, employing a 0.25 km^2 grid mesh encompassing all of the Water Conservation Areas, Holey Land, Rotenberger Tract, parts of the Model Lands near the C-111 canal region, and most of Everglades National Park and Big Cypress National Preserve. The vector topology of the canal/levee network and the point locations of water control structures were constant during the simulation period. The habitat succession module was operating, as were all other ecological modules, providing dynamic feedbacks among the physics, chemistry, and biology of the mosaic of ecosystems in the landscape. Dynamic boundary conditions included daily data on rainfall, potential evapotranspiration, managed water control structure flows with associated constituent concentrations, and stage (along the borders of the domain, including annually-recurring, monthly mean tidal amplitudes). Full descriptions of the requisite data and the functionality of the algorithms and source code are provided in other Chapters of this documentation.

6.6 Performance results

6.6.1 Hydrologic performance

6.6.1.1 Stage: statistical metrics

The marsh stage monitoring locations used in evaluating the model performance are mapped in Figure 6.1. Table 6.1 shows the statistical performance metrics for the simulated vs. observed stage data at each location during the 1981-2000 historical simulation period. The median bias of predicted stages was 0 cm. The median Nash-Sutcliffe Efficiency statistic was 0.61 for the simulation. The spatial distribution of the mean surface water depths (background of Figure 6.1) indicates relatively deep inundation in southern portions of Water Conservation Areas and large slough features draining to the southwest and south in Everglades National Park. Biases do not appear in any spatial trend, but boundary conditions along the model periphery resulted in higher biases in and immediately adjacent to canals and estuarine regions.

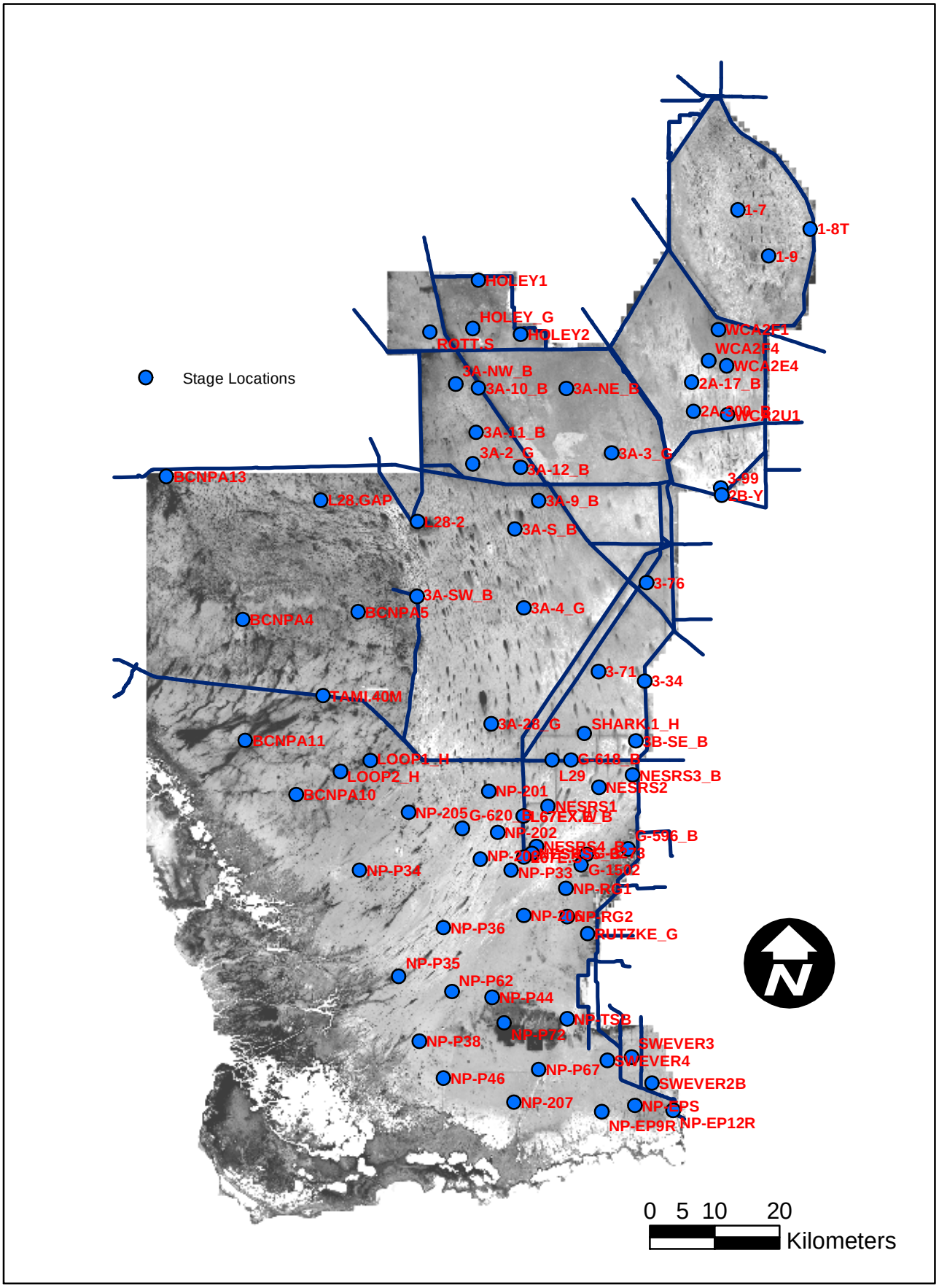


Figure 6.1. Map of stage monitoring site locations..

Figure 6.2 Map of statistical bias in model predictions of observed water stage elevations in marsh locations. Background map is the simulated mean daily surface water depth during 1981-2000. Statistics are detailed in Table 6.1.

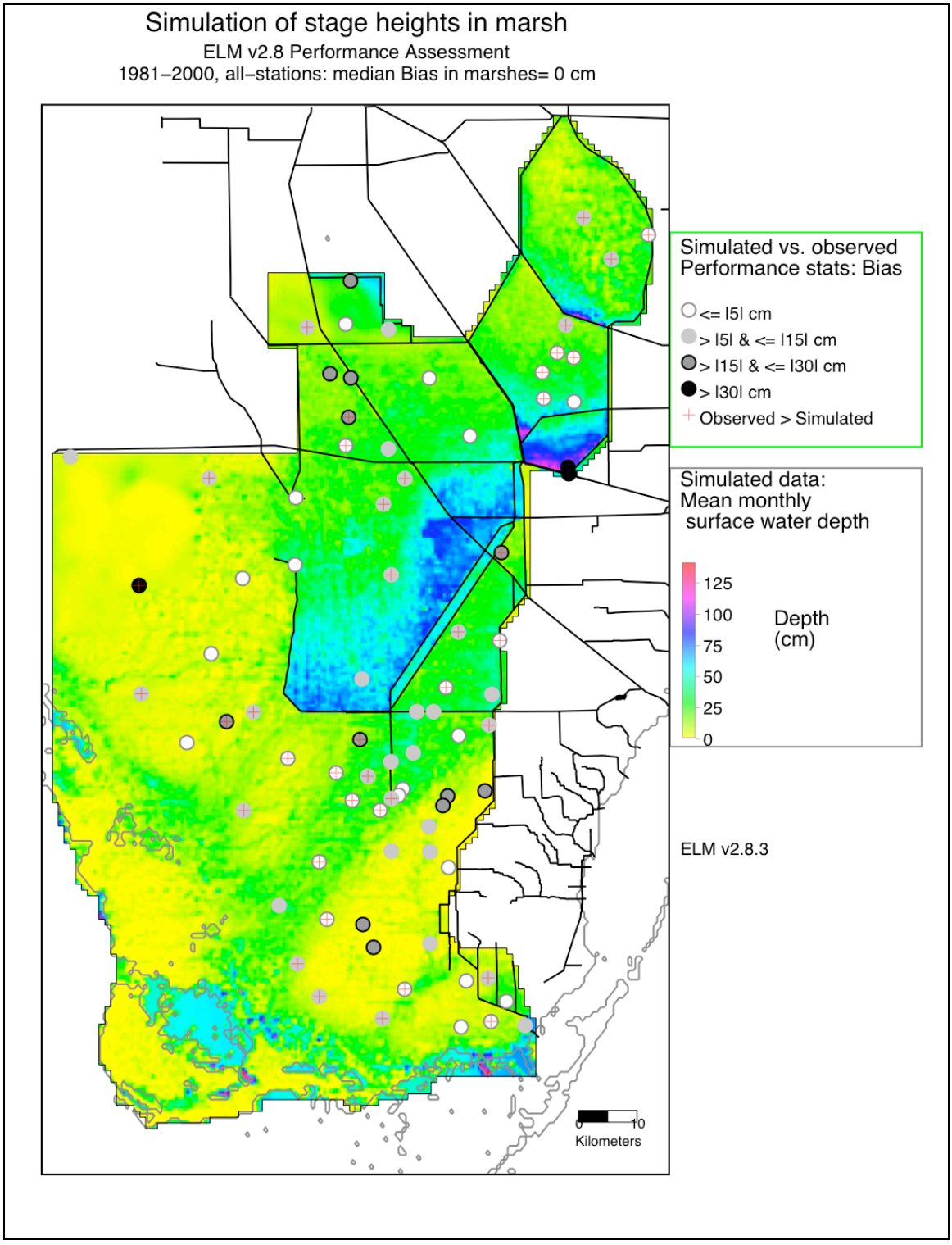


Table 6.1. Statistical evaluation of simulated vs. observed stage, 1981 – 2000. Units of Bias (observed minus simulated) and RMSE are meters.

		Stage 1981-2000				
Site	Basin	N	Bias (m)	RMSE (m)	R2	NS Eff.
_1-7	WCA1	7046	0.12	0.17	0.73	0.12
1-8T	WCA1	6869	0.05	0.14	0.79	0.63
_1-9	WCA1	6879	0.05	0.12	0.77	0.56
WCA2F1	WCA2A	2259	0.06	0.16	0.80	0.64
WCA2F4	WCA2A	1941	0.01	0.16	0.74	0.62
WCA2E4	WCA2A	2260	0.03	0.17	0.76	0.60
2A-17_B	WCA2A	7305	0.01	0.18	0.76	0.58
2A-300_B	WCA2A	7278	0.00	0.20	0.70	0.62
WCA2U1	WCA2A	2150	-0.03	0.19	0.70	0.64
3A-NW_B	WCA3A	7035	-0.22	0.26	0.71	0.06
3A-10_B	WCA3A	6445	-0.16	0.20	0.77	0.08
3A-NE_B	WCA3A	6813	-0.03	0.21	0.69	0.68
3A-11_B	WCA3A	6487	0.17	0.20	0.85	-0.01
3A-3_G	WCA3A	7305	0.00	0.14	0.86	0.86
3A-2_G	WCA3A	7145	0.01	0.11	0.88	0.87
3A-12_B	WCA3A	6738	-0.07	0.17	0.68	0.53
3A-9_B	WCA3A	6969	0.07	0.13	0.86	0.79
L28-2	WCA3A	4007	-0.04	0.16	0.57	0.47
3A-S_B	WCA3A	6871	0.07	0.13	0.85	0.74
3A-4_G	WCA3A	7305	0.08	0.14	0.88	0.80
3A-28_G	WCA3A	7295	-0.10	0.14	0.89	0.77
_3-99	WCA2B	3338	-0.35	0.40	0.67	-0.50
2B-Y	WCA2B	5515	-0.54	0.62	0.82	0.01
_3-76	WCA3B	3390	0.23	0.25	0.54	-1.90
_3-71	WCA3B	3454	0.11	0.15	0.64	0.20
_3-34	WCA3B	1633	0.04	0.09	0.84	0.77
SHARK.1_H	WCA3B	6684	0.05	0.12	0.83	0.79
3B-SE_B	WCA3B	6029	-0.09	0.15	0.88	0.80
HOLEY1	Holey L.	4041	-0.18	0.22	0.65	-0.02
HOLEY_G	Holey L.	5599	-0.05	0.22	0.50	-0.52
HOLEY2	Holey L.	4046	-0.11	0.20	0.56	0.35
ROTT.S	Roten. T.	5208	0.08	0.16	0.56	0.39
BCNPA13	BCNP	1923	-0.10	0.20	0.49	0.33
L28.GAP	BCNP	6393	0.10	0.17	0.65	0.42
3A-SW_B	BCNP/3A	6641	0.00	0.09	0.86	0.83
BCNPA5	BCNP	3636	-0.02	0.13	0.74	0.62
BCNPA4	BCNP	3601	0.33	0.38	0.58	-1.27
TAMI.40M	BCNP	7305	-0.02	0.17	0.75	0.68
BCNPA11	BCNP	3549	0.12	0.25	0.44	0.15
continued next page …						

Table 6.1 continued. Statistical evaluation of simulated vs. observed stage, 1981 – 2000. Units of Bias (observed minus simulated) and RMSE are meters.

		Stage 1981-2000				
Site	Basin	N	Bias (m)	RMSE (m)	R2	NS Eff.
G-618_B	ENP	7124	-0.11	0.16	0.82	0.56
L29	ENP	7305	-0.06	0.14	0.73	0.61
LOOP1_H	ENP	5938	0.06	0.13	0.69	0.61
LOOP2_H	ENP	5972	0.16	0.22	0.74	0.29
NESRS3_B	ENP	5579	0.07	0.16	0.66	0.58
NESRS2	ENP	6228	-0.02	0.09	0.75	0.75
NP-201	ENP	5723	0.17	0.21	0.83	0.43
BCNPA10	ENP	3637	-0.02	0.14	0.53	0.47
NESRS1	ENP	6536	-0.05	0.10	0.76	0.66
NP-205	ENP	7149	0.04	0.14	0.81	0.79
L67EX.W	ENP	6319	0.00	0.17	0.80	0.62
L67EX.E_B	ENP	6187	-0.08	0.14	0.72	0.54
G-620_B	ENP	6264	0.03	0.11	0.79	0.78
NP-202	ENP	7069	0.07	0.13	0.83	0.68
NESRS4_B	ENP	4854	-0.04	0.11	0.72	0.59
G-596_B	ENP	7282	-0.25	0.30	0.59	-0.42
NESRS5_B	ENP	4953	-0.02	0.08	0.78	0.67
G-3273	ENP	6137	-0.19	0.26	0.73	0.38
L67E.S	ENP	3631	0.06	0.16	0.61	0.54
NP-203	ENP	7049	0.04	0.12	0.74	0.71
G-1502	ENP	7305	-0.16	0.24	0.75	0.53
NP-P33	ENP	7147	0.02	0.12	0.67	0.67
NP-P34	ENP	6971	0.09	0.17	0.82	0.63
NP-RG1	ENP	1570	-0.09	0.13	0.87	0.68
NP-206	ENP	6641	-0.06	0.20	0.75	0.71
NP-RG2	ENP	1502	-0.08	0.13	0.88	0.73
NP-P36	ENP	6952	0.03	0.12	0.68	0.63
RUTZKE_G	ENP	2369	-0.01	0.14	0.79	0.60
NP-P35	ENP	6851	-0.05	0.12	0.74	0.61
NP-P62	ENP	6851	0.03	0.13	0.80	0.78
NP-P44	ENP	6440	-0.20	0.29	0.79	0.52
NP-TSB	ENP	7299	-0.09	0.16	0.89	0.77
NP-P72	ENP	7186	-0.20	0.28	0.77	0.48
NP-P38	ENP	6896	0.07	0.13	0.88	0.56
SWEVER3	ENP	5330	0.07	0.11	0.81	0.29
SWEVER4	ENP	5582	-0.01	0.15	0.78	-0.06
NP-P67	ENP	7107	0.02	0.10	0.80	0.75
NP-P46	ENP	6680	0.06	0.14	0.74	0.32
SWEVER2B	ENP	5488	-0.02	0.07	0.79	0.77
NP-207	ENP	6755	0.07	0.11	0.86	0.66
NP-EPS	ENP	5240	0.03	0.06	0.77	0.63
NP-EP12R	ENP	2828	-0.05	0.07	0.70	0.40
NP-EP9R	ENP	2608	-0.01	0.06	0.84	0.81
	Median:	6356	0.00	0.15	0.76	0.61

6.6.1.2 Stage: graphical indicators

These visualizations of the temporal trends in simulated and observed data are an important component of understanding the model performance, particularly with respect to recognizing any unique aspects of the data dynamics at a particular site. Figure 6.3 shows an example of the time series of stage hydrographs in long and in short hydroperiod areas. The model effectively captured the spatial differences between southern Everglades marl prairie region that is periodically flooded, and a Water Conservation Area 3A location that is virtually always inundated with relatively deep surface water.

Figure 6.3 (following page). Example plots of time series and Cumulative Frequency Distributions (CFD) of simulated and observed stage in short hydroperiod (NP-206, Everglades National Park) and long hydroperiod (3A-SB, WCA-3A) sites.

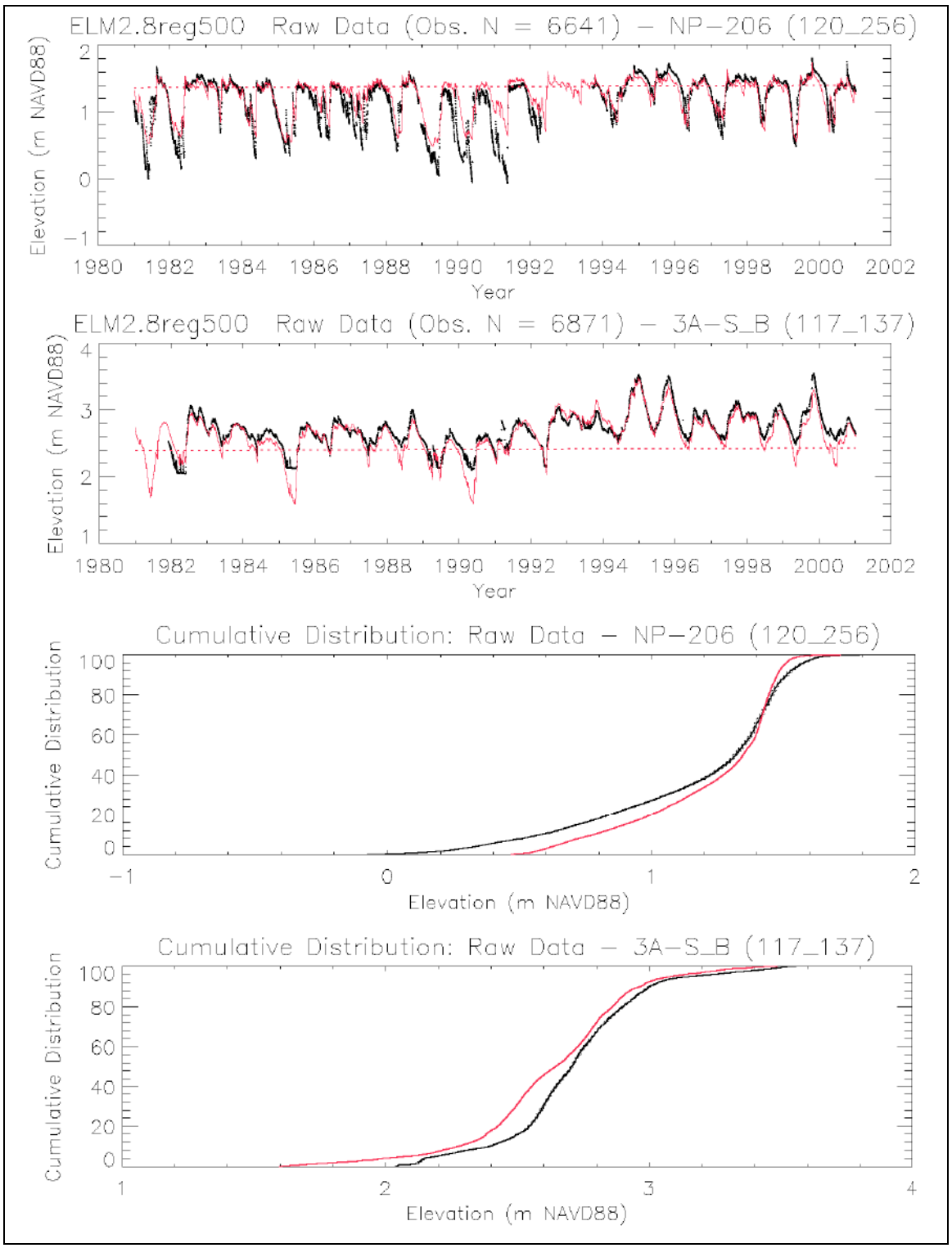
The red dashed line in the stage hydrographs is the model grid cell's land surface elevation, which is a time-varying output variable of the model. The model grid cell column and row locations are shown in parentheses (col_row) of each plot's title.

Time series plots: All data, with no temporal aggregation, of daily observations (black dots) and model results (red line).

Cumulative Frequency Distributions: The CFDs of the simulated and observed (raw, un-aggregated) data; the 95% confidence interval for observed data is shown in the dashed black lines. Note that only paired simulated and observed data points are used.

Appendix B. The complete set of graphics for all monitoring sites in the greater Everglades is provided in Appendix B.

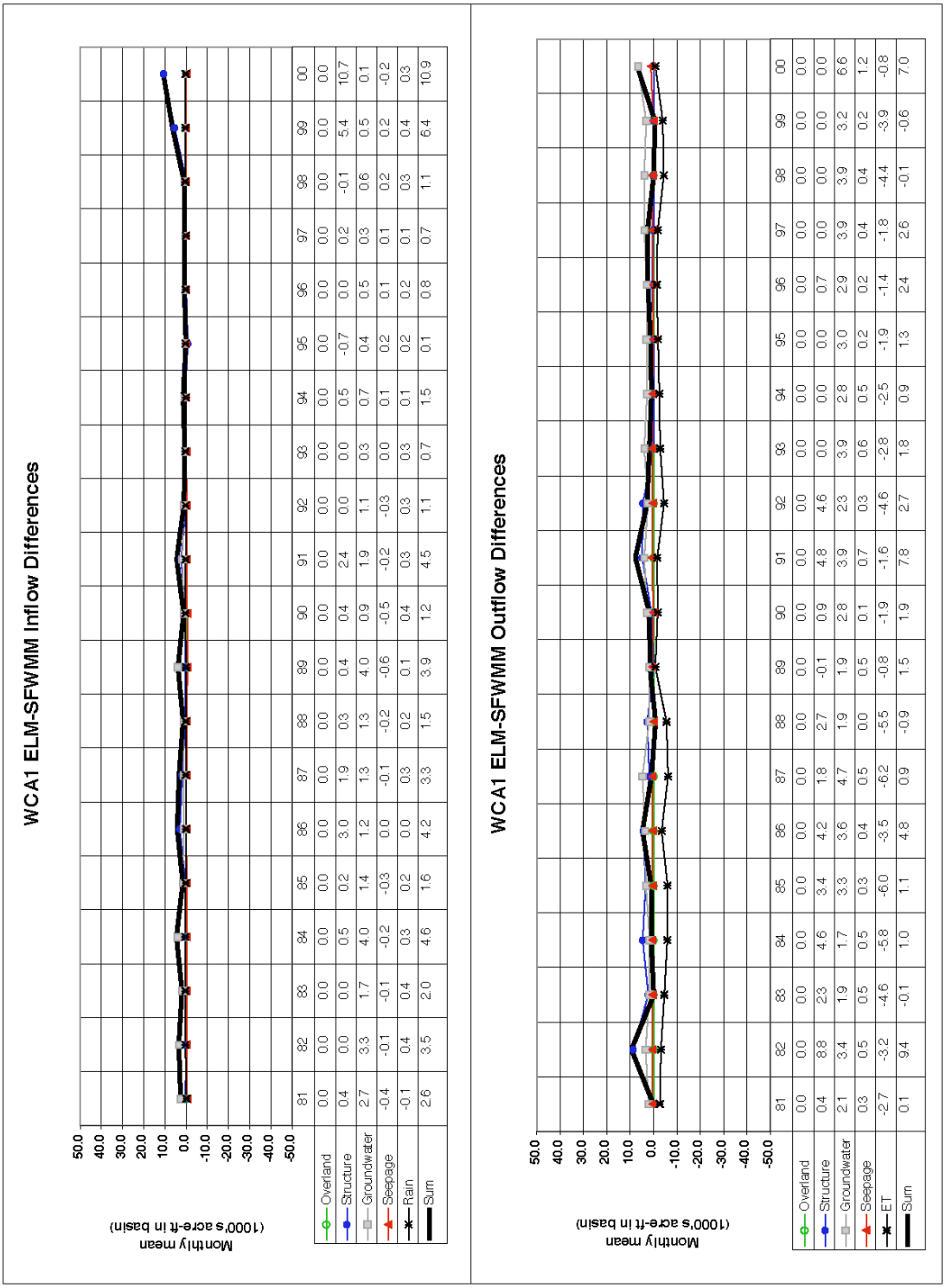
Figure 6.3. Time series and Cumulative Frequency Distributions of simulated and observed stages for long and short hydroperiod sites. See full Figure legend above.



6.6.1.3 Consistency: inter-model water budget indicators

The water budgets of the ELM were generally similar to those of the SFWMM. For each of the major hydrologic basins, we compared the annual flows into and out of each Water Conservation Area. Figure 6.4 shows an example of such a comparison. Very minor differences in rainfall are due to the different spatial scales and discretization of grid cells. Other differences are observable in some years for other flows, but do not represent significant volumes (relative to the size of the basin). For each Water Conservation Area, Appendix C provides the actual hydrologic budgets for ELM, and the differences between the SFWMM and ELM.

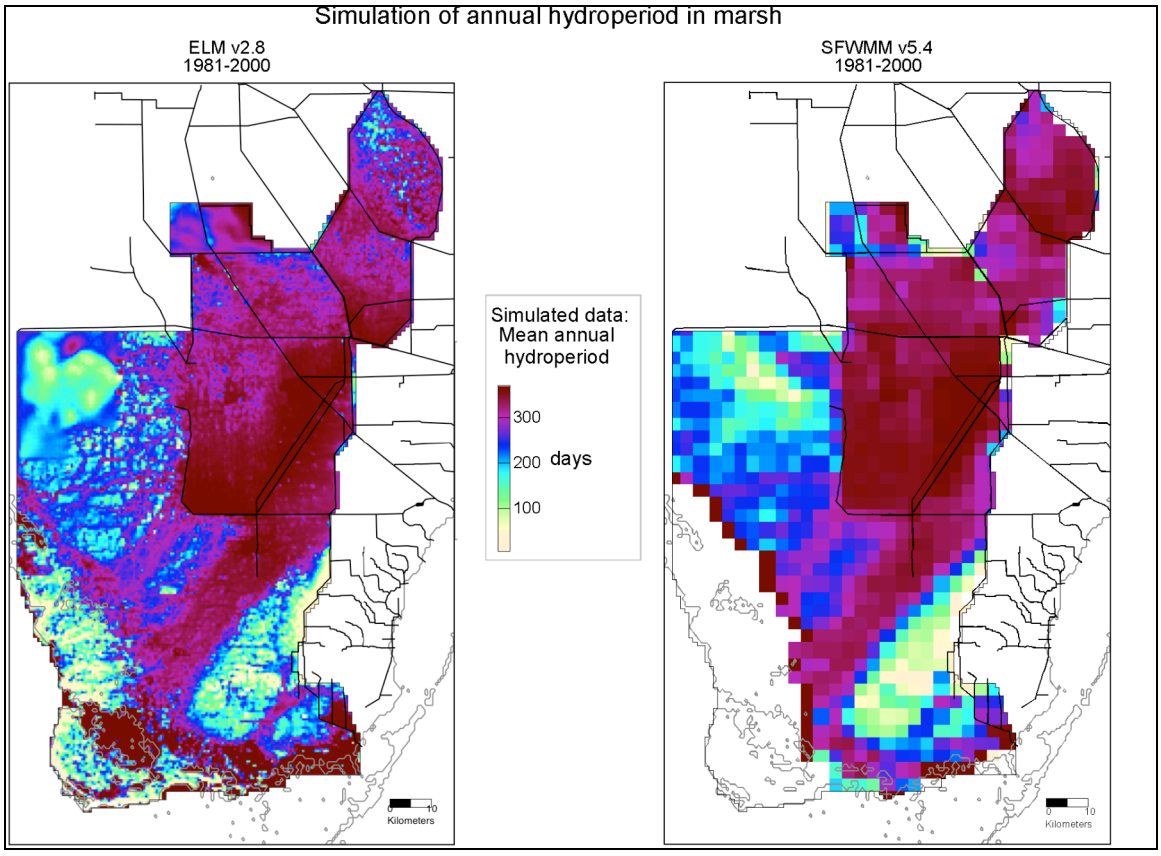
Figure 6.4. Annual differences between the SFWMM v5.4 and ELM v2.8 for all inflows and outflows for Water Conservation Area 1 (WCA-1). Horizontal inflows and outflows are via overland surface water (none for this basin), water control structures, subsurface groundwater, and levee seepage flows. Rainfall and evapotranspiration (ET) are vertical inflows and outflows, respectively.



6.6.1.4 Consistency: inter-model hydroperiod indicators

Another indicator of consistency between the ELM and the SFWMM is a comparison of the maps of the mean annual hydroperiod that is simulated by each model. Figure 6.5 indicates that the ELM generally mimics the distribution of hydroperiods, with some differences in the ELM capturing finer scaled features (largely due to finer scaled land surface elevation input data). Importantly, the ELM v2.8 utilizes newer land surface elevation data (see Data Chapter 4 of this document), resulting in some further differences (principally Big Cypress, WCA-3A, and WCA-1).

Figure 6.5. Mean annual hydroperiod simulated by the ELM and by the SFWMM, displaying only the portion of the SFWMM domain that overlaps with that of the ELM. The SFWMM grid cells are approximately 10.4 km^2, compared to the 0.25 km^2 grid resolution of the ELM. (As indicated, the SFWMM domain does not extend to the southwestern mangrove-dominated region along the Gulf of Mexico).



6.6.1.5 Consistency: Flow tracer (chloride) indicators

The marsh and canal water quality monitoring locations used in evaluating the model performance are mapped in Figure 6.6. The distribution of chloride (Cl) concentrations throughout the freshwater Everglades showed patterns of long-term flow regimes that were consistent with our understanding of major flow paths (Figure 6.7), most notably the "ring" of higher Cl encircling WCA-1, and high concentration inputs into WCA-2A. Other canal inputs within WCA-3A transported the tracer into Everglades National Park[3]. The seasonal relative bias metric indicated a distribution of relative errors that tended to be higher in close proximity to higher concentrations in canals, similar to the trends of phosphorus concentrations. The median seasonal relative bias of all stations was 10% in the marshes, and 11% in canals (Table 2). The median seasonal bias was 6 mg L^{-1} in the marsh, and 13 mg L^{-1} in the canals.

[3] The distribution of Cl concentrations go "off-the-freshwater-scale" in the estuarine southern Everglades, with Cl concentrations that were << 1 parts per thousand generally corresponding to the extent of mangrove and other estuarine habitat types.

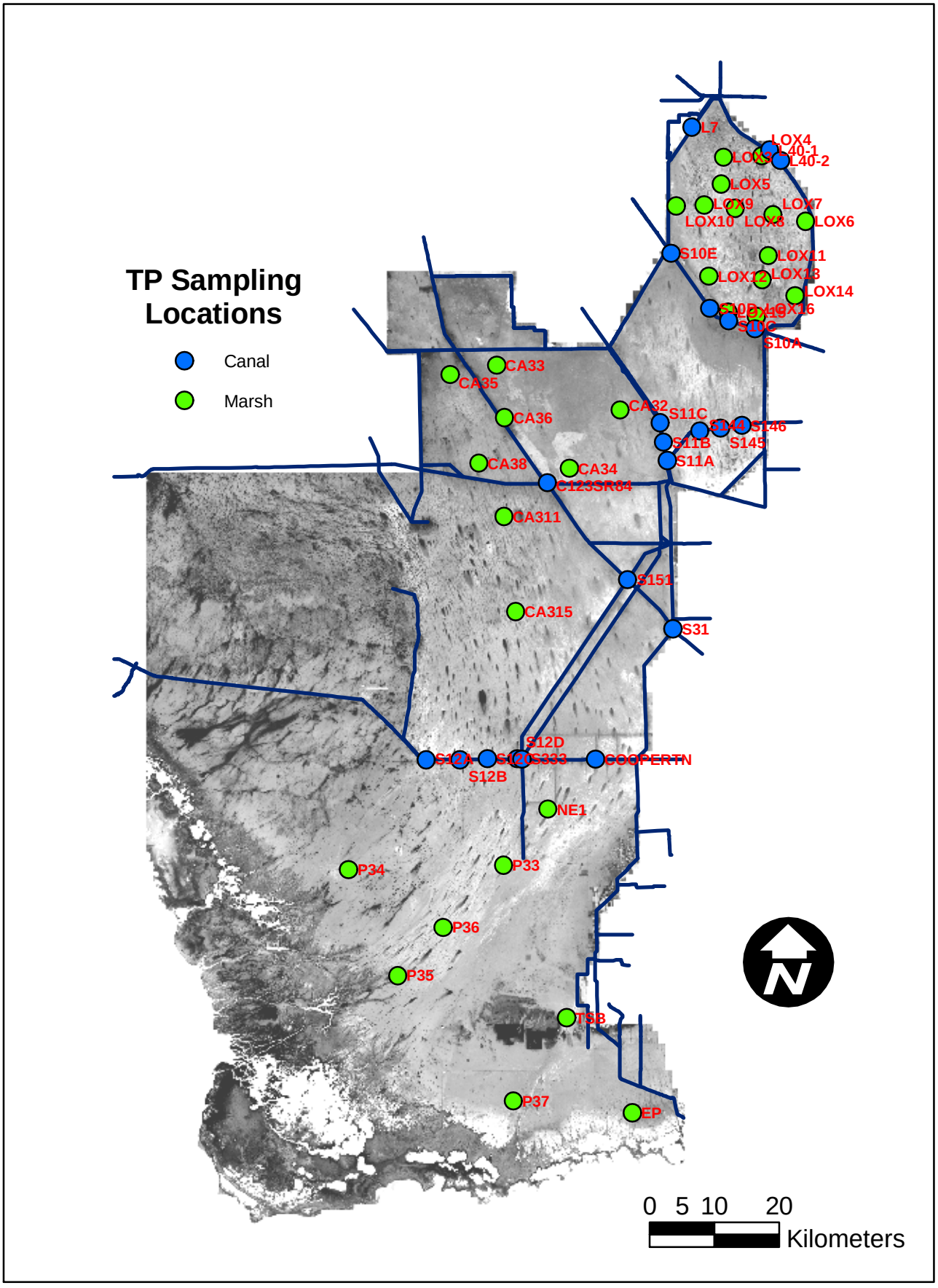


Figure 6.6 Map of most TP and CL monitoring site locations (see also Figure 6.6b).

Figure 6.6b. Map of water quality monitoring locations in WCA-1 and WCA-2A.
Note: the grid that is shown is 1 km^2, which is four times coarser than the 0.25 km^2 scale of the current model.

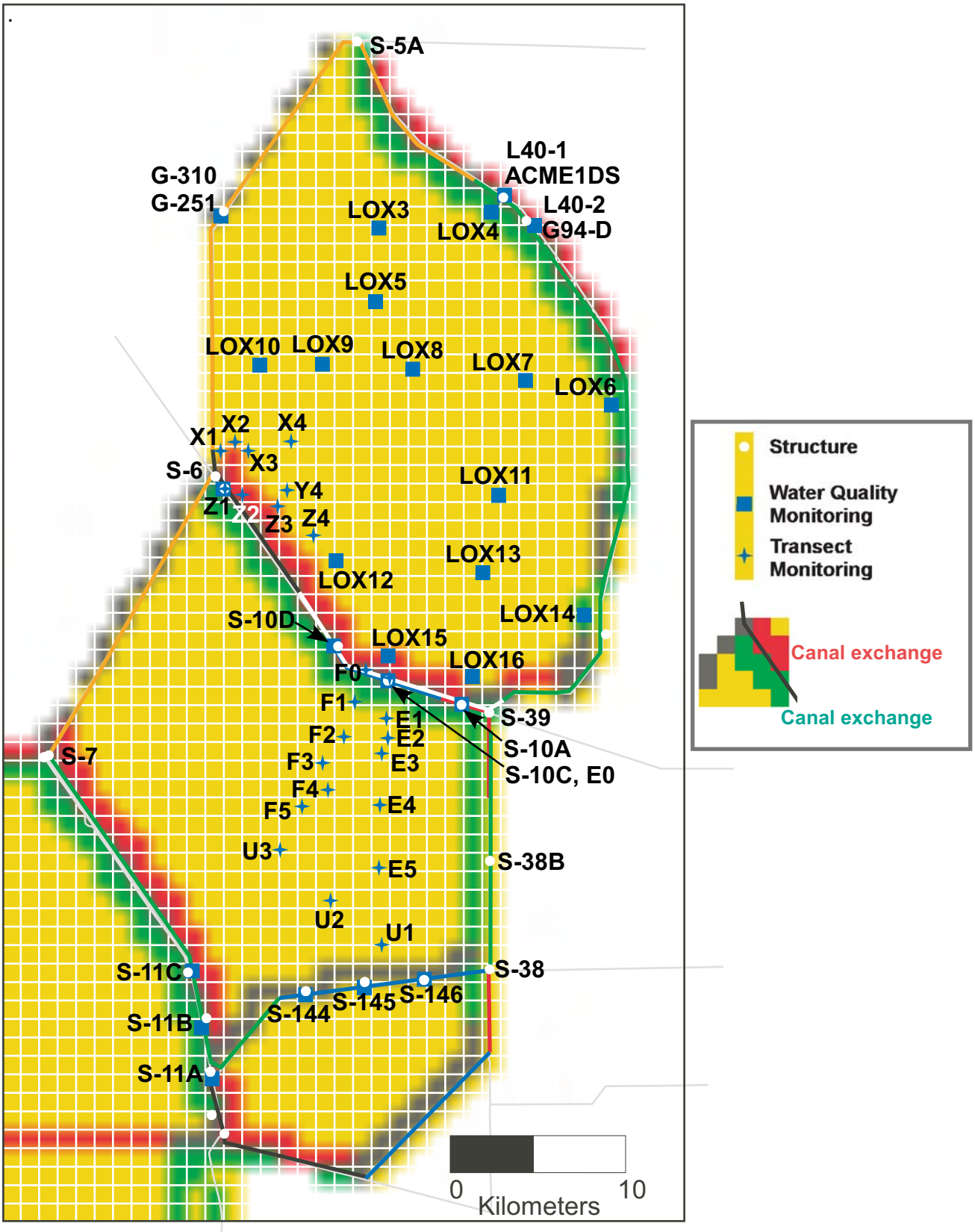


Figure 6.7 Map of statistical seasonal bias in model predictions of observed chloride (Cl) concentrations in marsh and canal locations. Background map is the simulated mean monthly Cl concentration during 1981-2000. Statistics are detailed in Table 6.2.

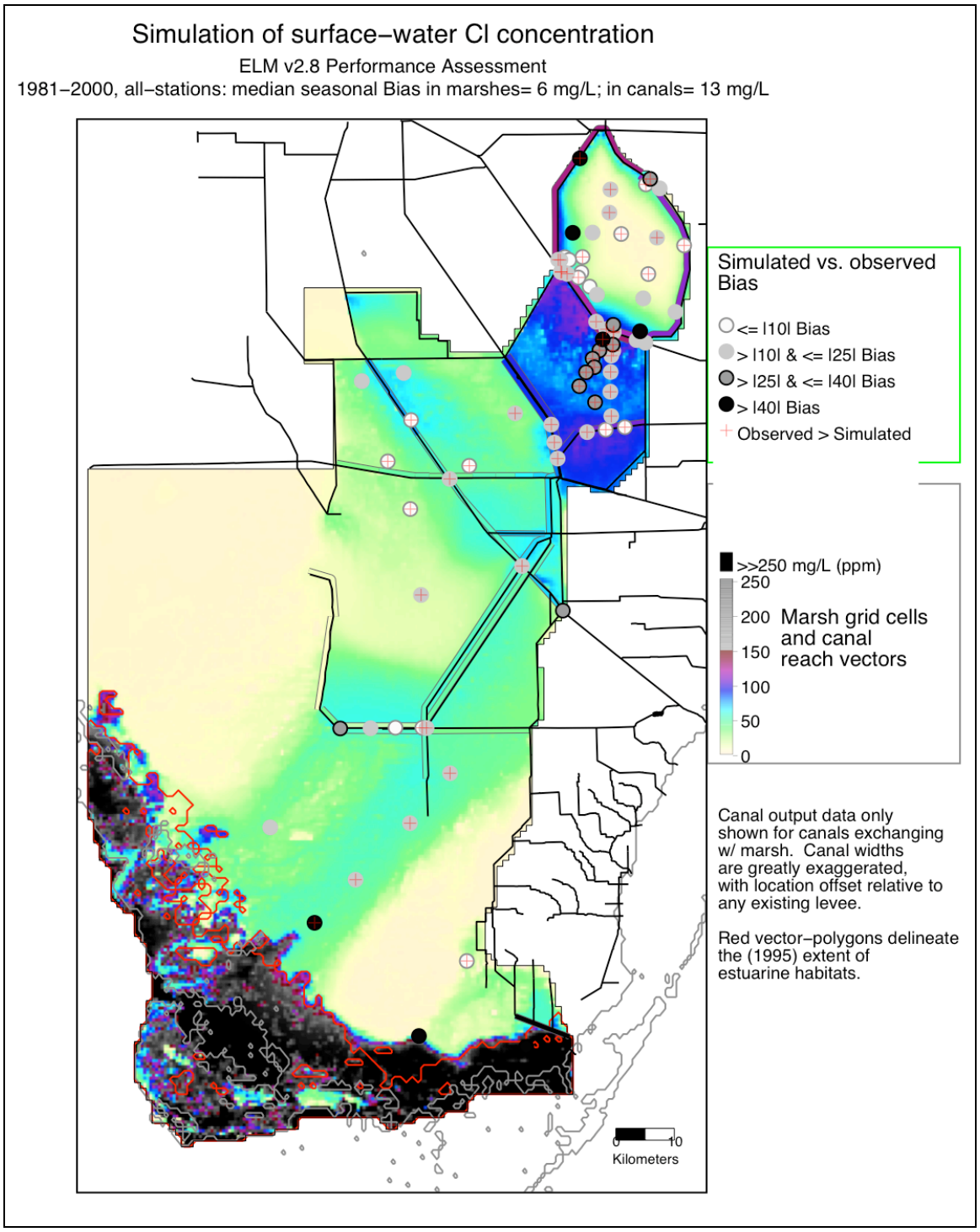


Table 6.2. Statistical evaluation of simulated vs. observed seasonal surface water chloride concentration, 1981 – 2000. Units of Bias (observed minus simulated) and RMSE are mg l^{-1} (ppm).

			1981-2000				
Site	Basin	Site type	N	ObsMean	RelBias	Bias	RMSE
LOX4	WCA1	Marsh	6	68	0.05	3	36
LOX3	WCA1	Marsh	6	37	0.42	15	20
LOX5	WCA1	Marsh	5	18	0.61	11	12
LOX10	WCA1	Marsh	6	28	-1.80	-51	58
LOX9	WCA1	Marsh	6	13	-1.02	-13	21
LOX8	WCA1	Marsh	6	15	0.58	8	9
LOX7	WCA1	Marsh	6	28	0.63	18	18
LOX6	WCA1	Marsh	6	41	0.06	2	20
X2	WCA1	Mar. Trans.	10	102	0.00	0	18
X4	WCA1	Mar. Trans.	10	51	0.02	1	20
X1	WCA1	Mar. Trans.	10	122	0.10	12	15
X3	WCA1	Mar. Trans.	10	87	-0.11	-10	29
Z1	WCA1	Mar. Trans.	10	125	0.20	26	28
Z2	WCA1	Mar. Trans.	10	108	0.10	10	19
Y4	WCA1	Mar. Trans.	10	51	-0.05	-3	26
LOX11	WCA1	Marsh	6	12	0.32	4	7
Z3	WCA1	Mar. Trans.	10	67	0.05	3	30
Z4	WCA1	Mar. Trans.	10	36	-0.16	-6	17
LOX12	WCA1	Marsh	6	28	-0.67	-19	22
LOX13	WCA1	Marsh	6	12	-0.98	-11	17
LOX14	WCA1	Marsh	6	21	-1.21	-25	28
LOX15	WCA1	Marsh	6	48	-1.02	-49	56
LOX16	WCA1	Marsh	6	14	-5.00	-71	74
F1	WCA2A	Mar. Trans.	13	156	0.34	53	58
E1	WCA2A	Mar. Trans.	14	149	0.28	42	50
F2	WCA2A	Mar. Trans.	14	150	0.27	41	44
E2	WCA2A	Mar. Trans.	14	125	0.22	27	35
E3	WCA2A	Mar. Trans.	14	124	0.20	25	36
F3	WCA2A	Mar. Trans.	14	143	0.25	36	38
F4	WCA2A	Mar. Trans.	14	137	0.25	35	39
CA33	WCA3A	Marsh	6	57	-0.43	-25	27
F5	WCA2A	Mar. Trans.	14	144	0.26	37	40
E4	WCA2A	Mar. Trans.	14	122	0.16	20	26
CA35	WCA3A	Marsh	6	33	-0.36	-12	18
U3	WCA2A	Mar. Trans.	14	133	0.28	37	40
E5	WCA2A	Mar. Trans.	14	113	0.17	19	25
U2	WCA2A	Mar. Trans.	14	129	0.30	38	40
CA32	WCA3A	Marsh	6	50	0.24	12	29
U1	WCA2A	Mar. Trans.	14	102	0.10	10	18

continued next page …

Table 6.2 continued. Statistical evaluation of simulated vs. observed seasonal surface water chloride concentration, 1981 – 2000. Units of Bias (observed minus simulated) and RMSE are mg l^{-1} (ppm).

			1981-2000				
Site	Basin	Site type	N	ObsMean	RelBias	Bias	RMSE
CA36	WCA3A	Marsh	6	71	0.06	4	8
CA38	WCA3A	Marsh	6	31	0.03	1	10
CA34	WCA3A	Marsh	6	55	0.03	2	10
CA311	WCA3A	Marsh	6	30	0.11	3	6
CA315	WCA3A	Marsh	6	34	0.33	11	12
NE1	ENP	Marsh	21	78	0.30	23	28
P33	ENP	Marsh	22	71	0.15	11	21
P34	ENP	Marsh	18	22	-1.12	-25	31
P36	ENP	Marsh	22	72	0.20	14	27
P35	ENP	Marsh	21	130	0.63	81	144
TSB	ENP	Marsh	22	39	0.08	3	18
P37	ENP	Marsh	20	30	-2.81	-85	192
EP	ENP	Marsh	19	180	-55.46	-9961	11978
L7	WCA1	Canal	10	228	0.24	55	106
L40-1	WCA1	Canal	18	132	0.25	33	45
L40-2	WCA1	Canal	18	80	-0.27	-22	43
S10A	WCA1	Canal	19	95	-0.23	-22	32
S10C	WCA1	Canal	21	131	0.11	14	30
S10D	WCA1	Canal	33	145	0.19	27	39
S39	WCA1	Canal	34	106	-0.14	-15	35
S10E	WCA1	Canal	15	141	0.13	19	31
X0	WCA1	Can. Trans.	10	131	0.13	17	22
Z0	WCA1	Can. Trans.	10	133	0.14	19	23
E0	WCA2A	Can. Trans.	14	128	0.11	14	21
F0	WCA2A	Can. Trans.	14	132	0.14	18	24
S144	WCA2A	Canal	23	127	0.09	11	25
S145	WCA2A	Canal	29	121	0.06	8	21
S146	WCA2A	Canal	24	117	0.02	3	21
S11A	WCA2A	Canal	26	118	0.12	14	27
S11B	WCA2A	Canal	27	122	0.14	16	29
S11C	WCA2A	Canal	33	117	0.10	11	22
C123SR84	WCA3A	Canal	18	75	0.18	13	17
S151	WCA3A	Canal	34	98	0.20	19	29
S12A	WCA3A	Canal	33	29	-1.13	-33	36
S12B	WCA3A	Canal	33	39	-0.61	-24	29
S12C	WCA3A	Canal	34	54	-0.17	-9	23
S12D	WCA3A	Canal	34	69	0.08	6	23
S333	WCA3A	Canal	33	77	0.16	12	25
S31	WCA3B	Canal	18	89	-0.38	-33	73
		Median All:	14	80	0.11	11	27
		Median Canal:	24	118	0.11	13	28
		Median Marsh:	10	62	0.10	6	27

6.6.2 Ecological performance

Evaluations of the ecological performance of ELM v2.8 for this project were not part of the current objectives, which focused on hydrologic performance. However, to demonstrate consistency with ELM v2.5, we briefly summarize some of the summary statistics for water quality performance – which showed improvements relative to those of ELM v2.5.

6.6.2.1 Surface water P concentration: statistical metrics

The marsh and canal TP concentration monitoring locations used in evaluating the model performance were the same as those used to evaluate ELM v2.5. The median Bias of all predicted TP concentrations in the marsh for the 1981-2000 period of record was 0 ug l^{-1} (ppb), and 0 ug l^{-1} in canal predictions.

6.7 Discussion

6.7.1 Model performance summary

Multiple methods were used to evaluate the performance characteristics of this model of greater Everglades hydro-ecology. The following summarizes those performance evaluations:

6.7.1.1 Model Objectives - Hydrology

- Water stage: median bias in predicting stage elevations was 0 cm for 82 marsh locations in the greater Everglades, whose hydroperiod ranged from continuously flooded to rarely flooded; these and other statistical metrics were comparable to the SFWMM
- Water flows: basin-wide flow budgets were in general concordance with those of the SFWMM;
- Water flows: distribution of chloride (Cl) concentrations throughout the freshwater Everglades showed patterns of long-term flow regimes that were consistent with our understanding of major flow paths, with a median relative error of 10% in marshes.

6.7.1.2 Model Consistency – Other Ecological Dynamics

- P concentration: median bias in predicting surface water TP concentrations was 0 ug l^{-1} for 78 marsh and canal locations in the greater Everglades, whose mean concentrations ranged from less than 10 to more than 100 ug l^{-1}

We note here that we have not evaluated the model performance within the mangrove-dominated region (that is delineated in the results map of the Cl tracer regional). Thus, application of these ELM Performance Measures within that specific region have an undocumented level of accuracy.

6.7.2 Performance refinements

The overall statistical summaries presented in this current Chapter 6 were influenced by a small number of locations where stage or water quality performance is significantly lower than other, even adjacent, locations. In this version, we did not take the time to correct isolated performance "problems" at a handful of locations.

There are limits to model performance that are supported by input data that drive the model, as discussed in the Uncertainty Chapter 7 of the ELM v2.5 Documentation Report. However, we also acknowledge that the current version can (and will) be improved within this boundary of expectations.

In the Model Refinement Chapter 9 of the ELM v2.5 Documentation Report, the near-term and long-term steps in model refinement were presented. Several of those refinements were made for ELM v2.8; there remain some relatively straightforward steps that can and will be taken to improve the model performance in the near term:

- Mangrove region in south and southwest: Tidal boundary conditions are extremely aggregated in both space and time. Spatial distributions of tidal amplitude are not accounted for in our implementation, nor does the monthly-mean tide, repeating every year, accommodate the observed fluctuations at both fine temporal scales, nor among years.

Importantly, we have not completed our efforts to improve upon the parameter estimates used in the model (see the Uncertainty Chapter 7 of the ELM v2.5 Documentation Report, which included an evaluation of model sensitivity to parameter modifications). Nevertheless, the existing code and data support sufficient model performance to enable users to have reasonable confidence in applying model results to long term planning under new management alternatives.

6.7.3 Conclusions

The ELM performance was rigorously quantified in the greater Everglades system for a multi-decadal period of record (1981 through 2000). For the current ELM v2.8, the primary goal was to provide fine scale (500 m grid resolution) hydrologic output for use in driving other ecological models (see the Executive Summary of this documentation report). For this specific project under the auspices of Joint Ecosystem Modeling partnership, the ecological performance measures were of secondary importance.

The model "skill" in predicting stage and flow was improved over earlier ELM versions, and continued to be consistent with the SFWMM output. Of particular interest with respect to "driving" fine scale ecological models, this scale of ELM hydrologic output exhibited detailed spatial patterns of flow, with improved connectivity among and within habitats (such as sloughs) relative to the 4x (ELM v2.5) or ~40x (SFWMM v5.4) coarser-scale resolution hydrologic models previously available for the greater Everglades region. Making use of a model with a "native" (original) resolution of 0.25 km^2 may likely lead to improved realism in modeling animal communities which respond to hydrologic patterns at these relatively fine scales.

6.8 Appendix A: Computational methods for statistics

Although numerous methods exist for analyzing and summarizing model performance, there is no consensus in the modeling community on a standard analytical suite for hydrology and ecological (incl. water quality) models. It appears most useful to use a variety of methods to evaluate model performance, as no single statistic can fully capture all of the important characteristics of a comparison between the simulated and observed data. We employed the below methods to estimate Bias, RMSE, R^2, and NS Efficiency in assessing some aspects of the model performance relative to observed data.

Bias:

$$\text{Bias} = \frac{\sum (x - y)}{n}$$

Where x is the field-observation values, y is the model-prediction values, and n is the number of observations.

Bias is calculated as the mean differences between paired modeled and observed values. It is a measure of how biased the overall values simulated by the model from the observed values. The bias should be as close to zero as possible.

Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{\sum (y - x)^2}{n}}$$

Where x is the field-observation values and y is the model-prediction values.

RMSE is the square root of the average values of the prediction errors squared. RMSE measures the discrepancy between modeled and observed values on an individual level to indicate accuracy of model predictions. Because of the quadratic term, RMSE gives greater weight to larger discrepancies than smaller ones. The RMSE should be as close to zero as possible.

Pearson product-moment correlation coefficient (R^2):

$$R^2 = \left(\frac{\sum (y - y_m)(x - x_m)}{\sqrt{\sum (y - y_m)^2 \sum (x - x_m)^2}} \right)^2$$

Where x_m is the observed mean of x (calculated as $\Sigma x/n$), and y_m is the model-predicted mean of observed y (calculated as $\Sigma y/n$).

The R^2 measure the degree of linear association between x and y (i.e., field observation and model predictions). It represents the amount of variability of one variable that is explained by correlating it with another variable. Depending on the strength of the linear relationships, the R^2 varies from 0.0 to 1.0, with 1.0 indicating a perfect fit.

Nash-Sutcliffe Efficiency (Eff):

$$\text{Eff} = 1 - \frac{\sum (y - x)^2}{\sum (x - x_m)^2},$$

Where x_m is the mean of the observed x, and y is the model prediction.

Like correlation coefficient, model efficiency is another overall indication of goodness of fit (Mayer and Butler 1993, Janssen and Heuberger 1995). Efficiency is equal to one minus the sum of squared prediction errors divided by the sum of squared deviation of observed values from the mean. It represents the amount of variability of one variable that is explained by modeled values. A model efficiency of 1.0 indicates a perfect fit between modeled and observed values, and a efficiency of 0.0 indicates the fit to $y = x$ is no better than $x = x_m$.

6.9 Appendix B: Time series & CFDs: stage

Figures B.1 – B.80. Plots of stage hydrographs and their associated Cumulative Frequency Distributions (CFD) for the period of record 1981-2000 at each monitoring location. The sequence of the figures is based on geographic location, starting in the northwest, moving towards the southeast. A map of all sites is provided in the Model Performance Chapter.

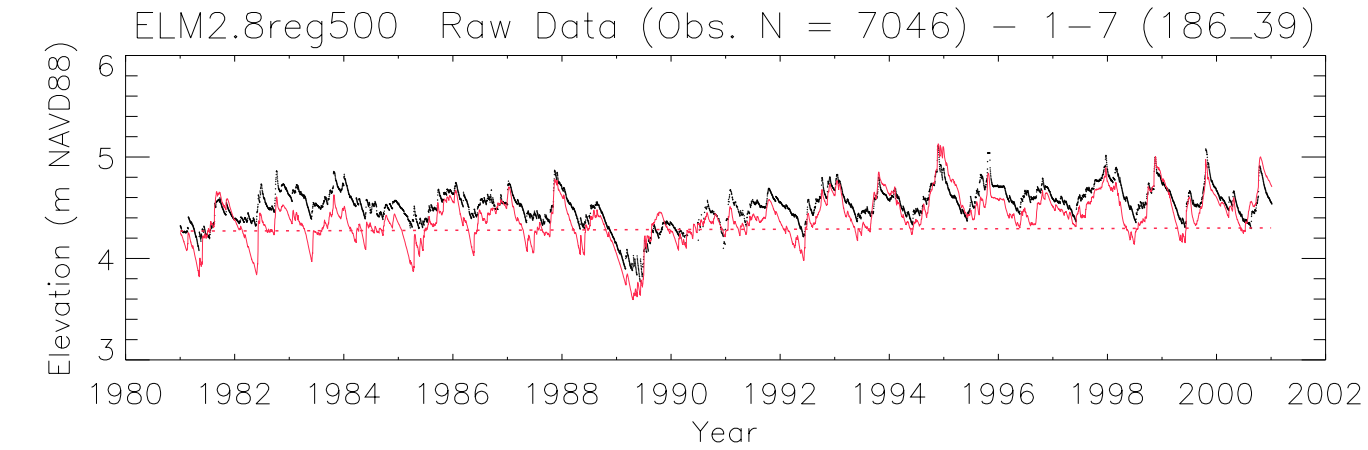
The red dashed line in the stage hydrographs is the model grid cell's land surface elevation, which is a time-varying output variable of the model. The model grid cell column and row locations are shown in parentheses (col_row) of each plot's title.

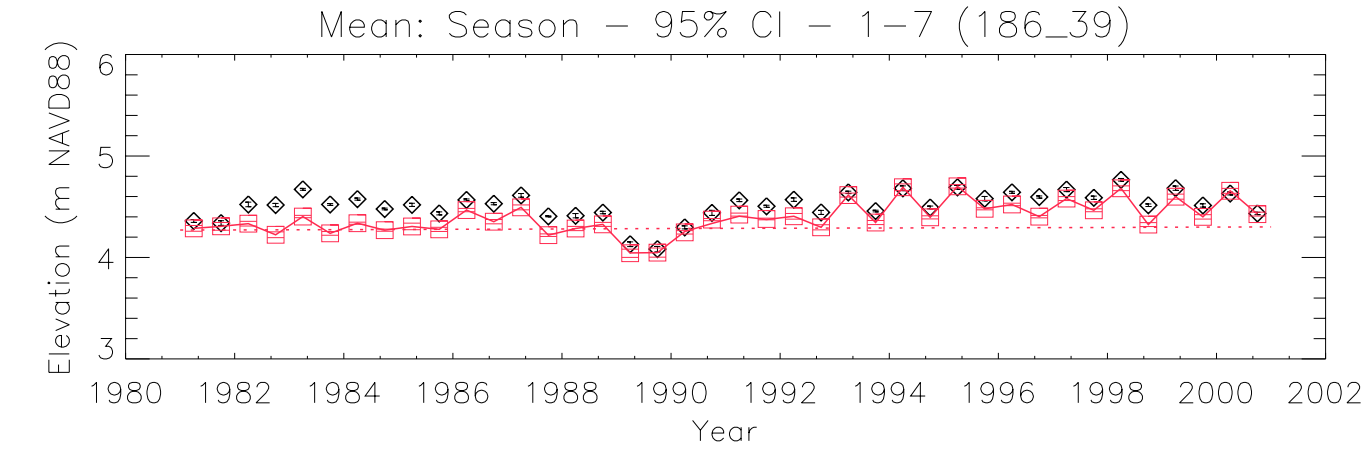
a) All data, with no temporal aggregation, of daily observations (black dots) and model results (red line).

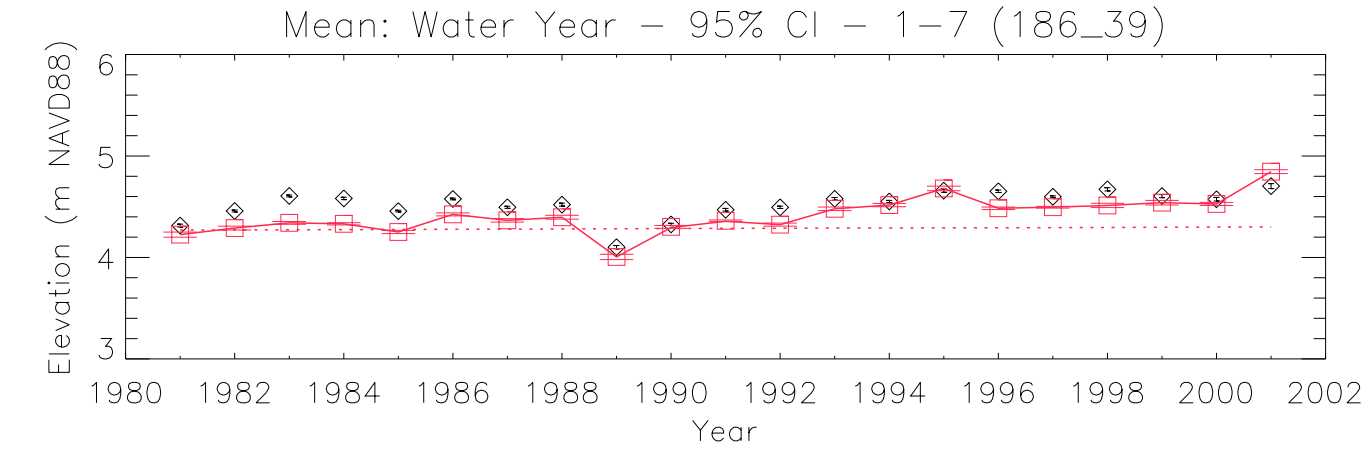
b) All data were aggregated into arithmetic mean values by wet and dry seasons within water years; the continuous lines pass through mean of all daily data points for each season; the mean of paired simulated & observed values are shown in red boxes and black diamonds, respectively; the 95% Confidence Interval (CI) of the paired means are shown by the "__" symbols in the red for the model and black for the observed data.

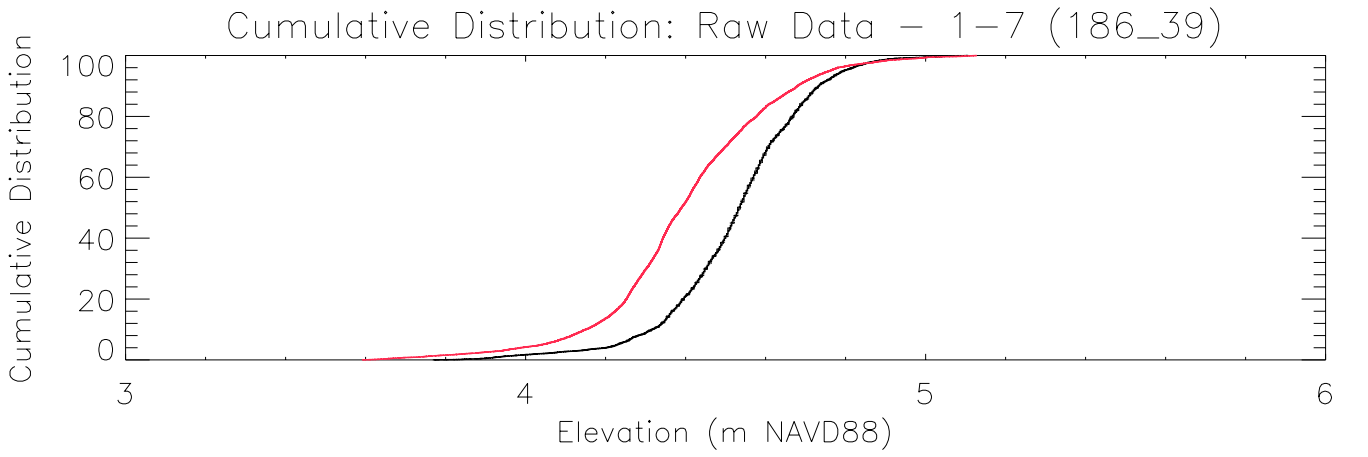
c) All data aggregated into arithmetic mean values by water year, with the same treatment as in plot b).

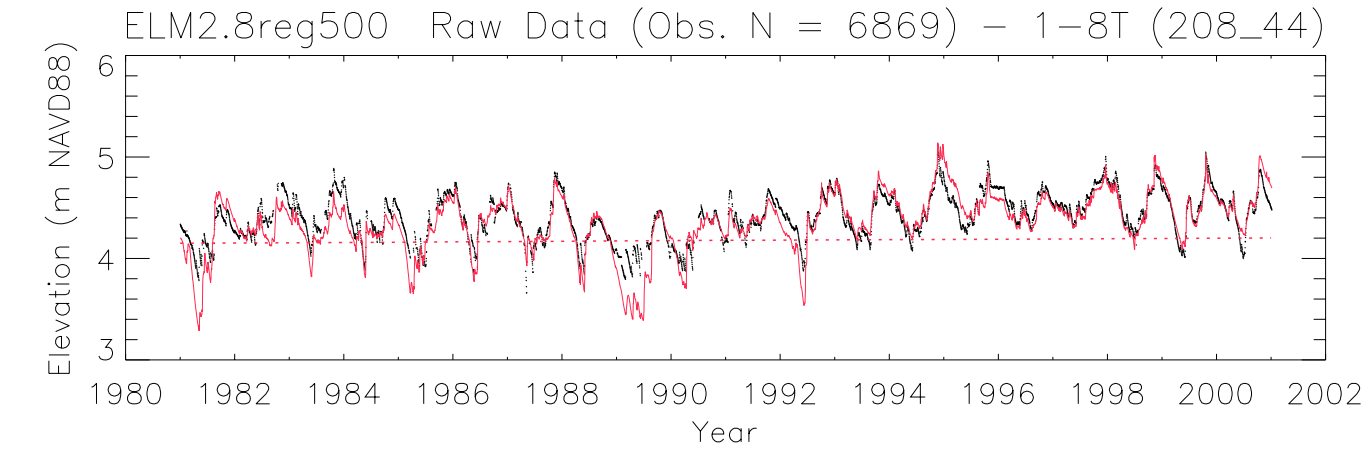
d) The cumulative frequency distributions of the simulated and observed (raw, un-aggregated) data; the 95% confidence interval for observed data is shown in the dashed black lines. Note that only paired simulated and observed data points are used.

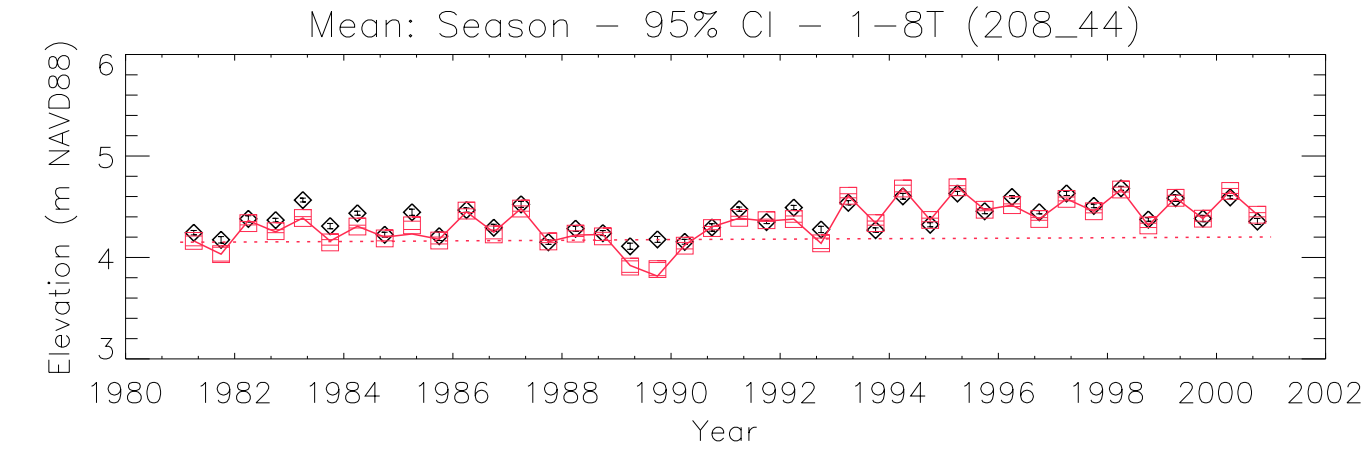

ELM2.8reg500 Raw Data (Obs. N = 7046) – 1–7 (186_39)
Elevation (m NAVD88)
6
5
4
3
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

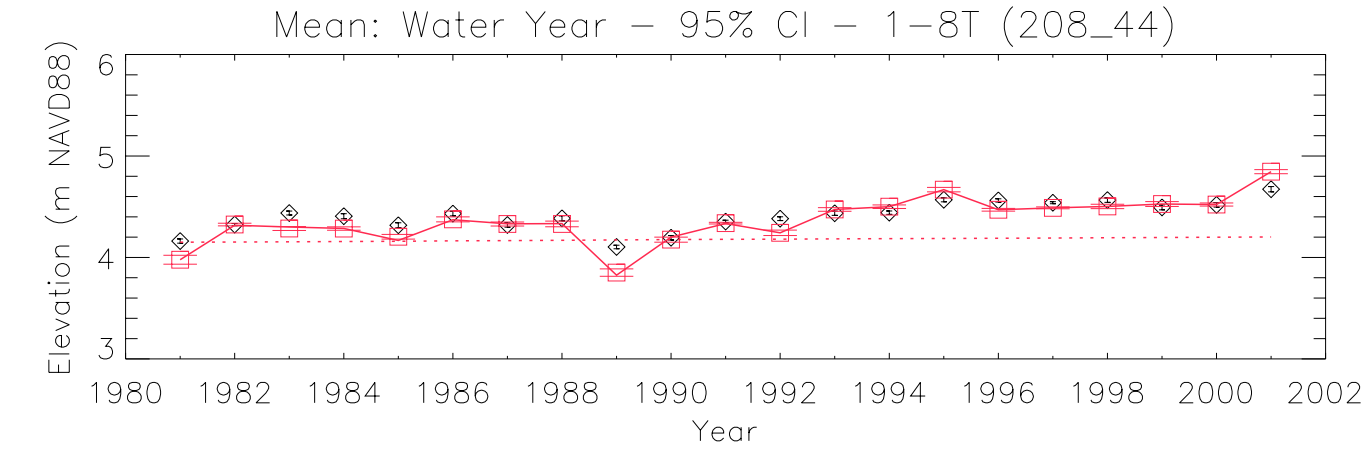

Mean: Season – 95% CI – 1–7 (186_39)
Elevation (m NAVD88)
6
5
4
3
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

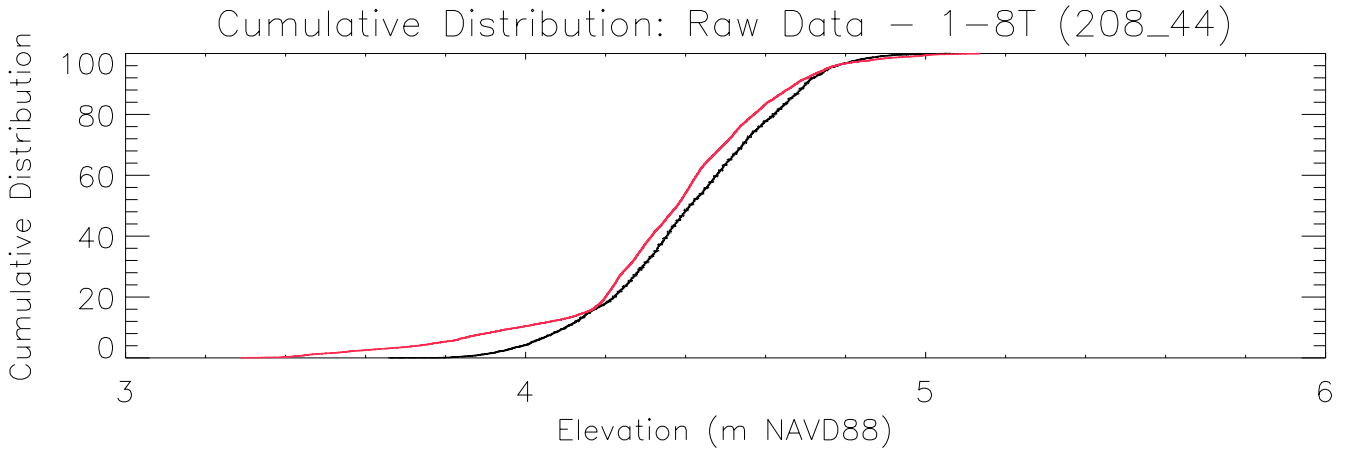

Mean: Water Year – 95% CI – 1–7 (186_39)
Elevation (m NAVD88)
6
5
4
3
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

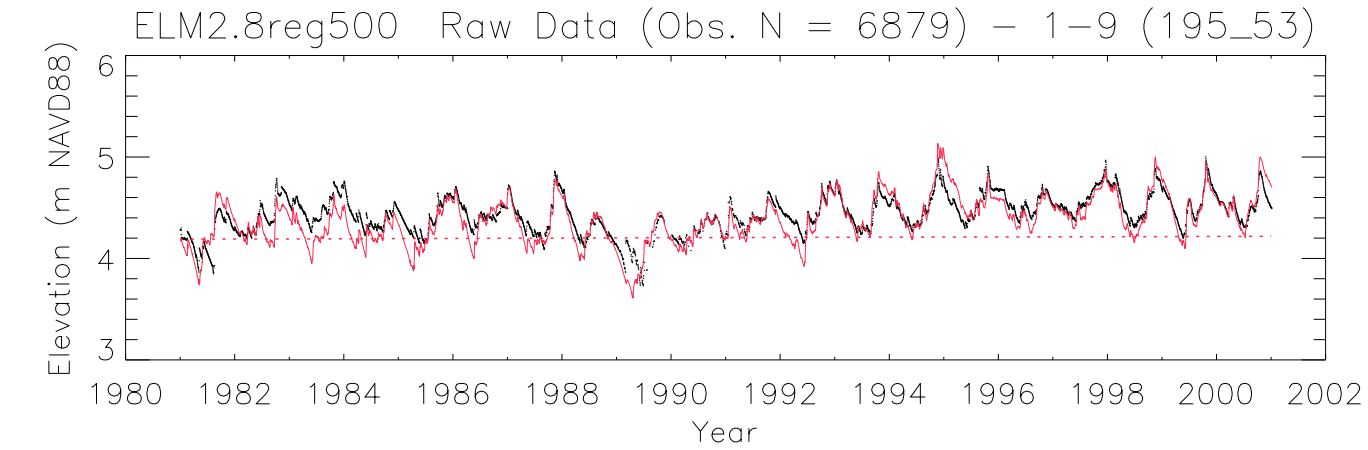

Cumulative Distribution: Raw Data – 1–7 (186_39)
Cumulative Distribution
100
80
60
40
20
0
3
4
5
6
Elevation (m NAVD88)

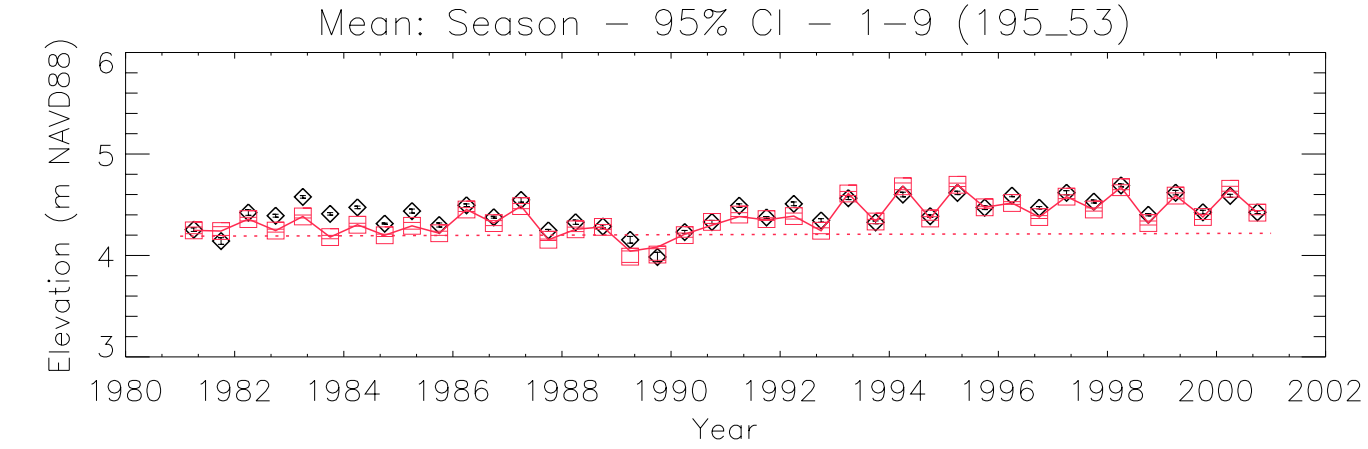

ELM2.8reg500 Raw Data (Obs. N = 6869) – 1–8T (208_44)
Elevation (m NAVD88)
6
5
4
3
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

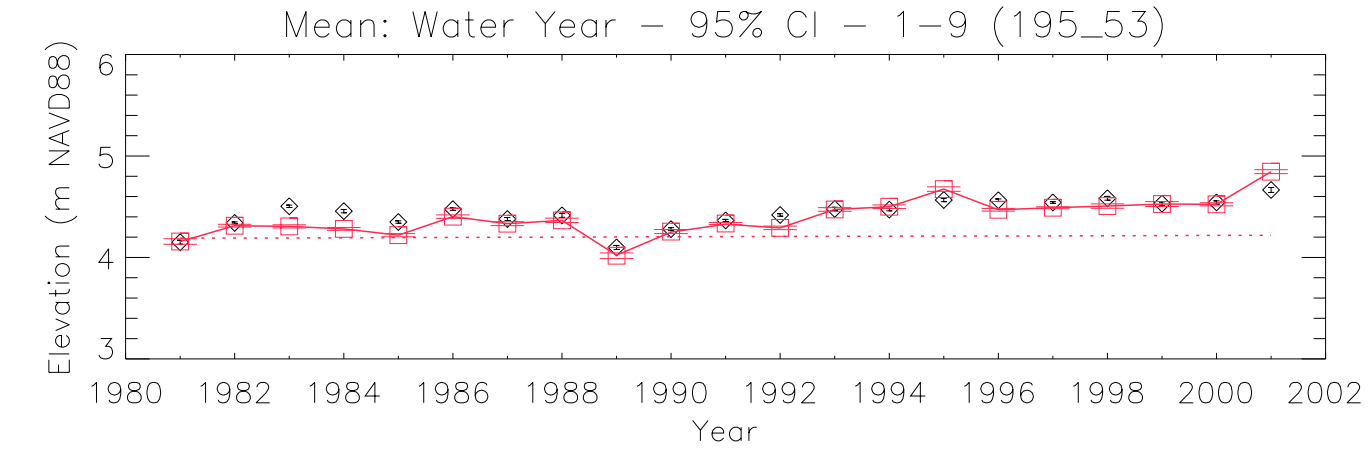

Mean: Season – 95% CI – 1–8T (208_44)
Elevation (m NAVD88)
6
5
4
3
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

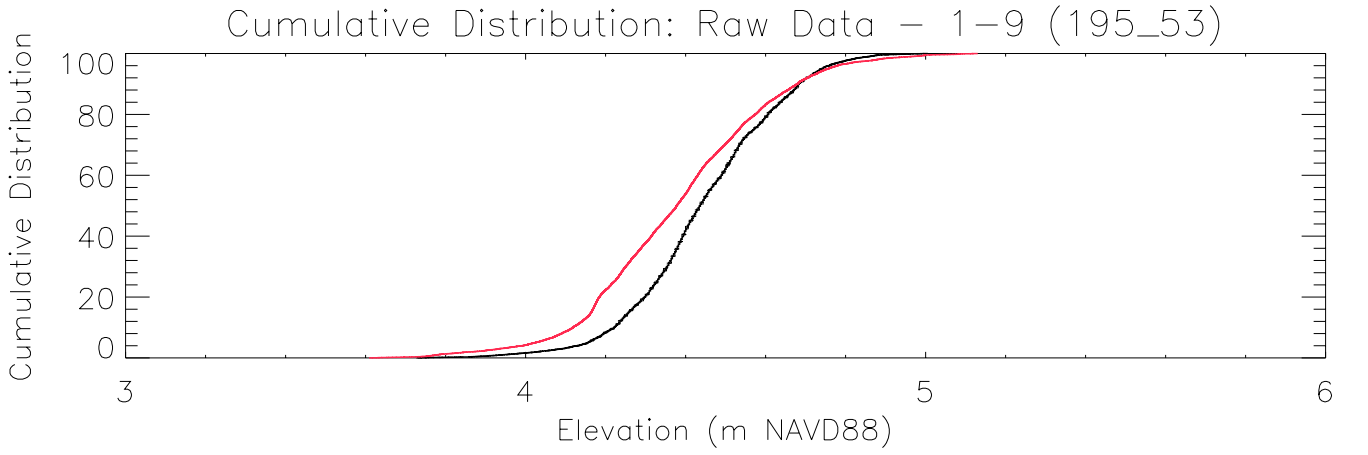

Mean: Water Year – 95% CI – 1–8T (208_44)
Elevation (m NAVD88)
6
5
4
3
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

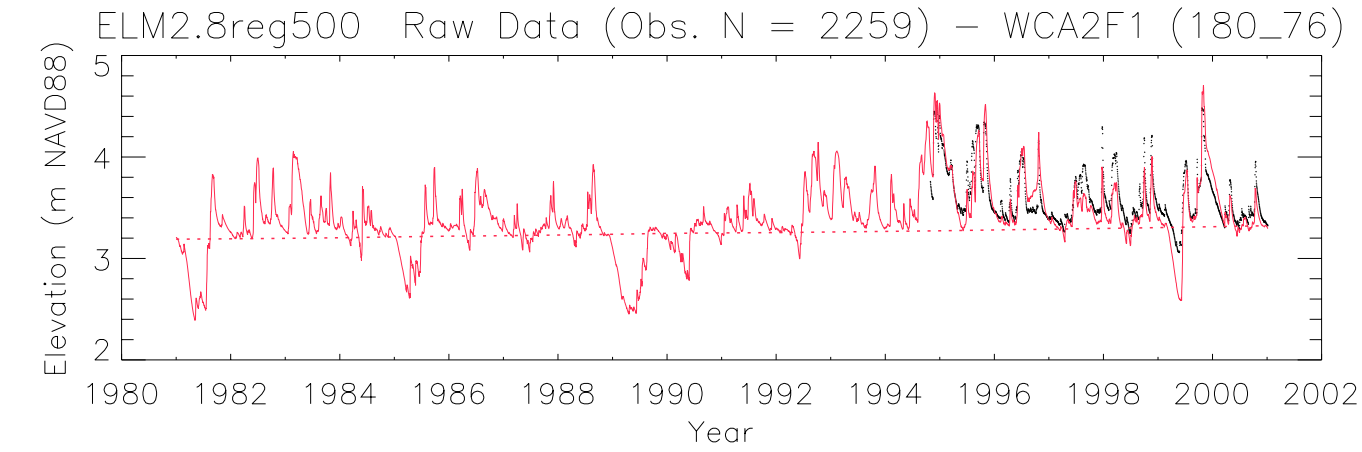

Cumulative Distribution: Raw Data – 1–8T (208_44)
Cumulative Distribution
100
80
60
40
20
0
3
4
5
6
Elevation (m NAVD88)

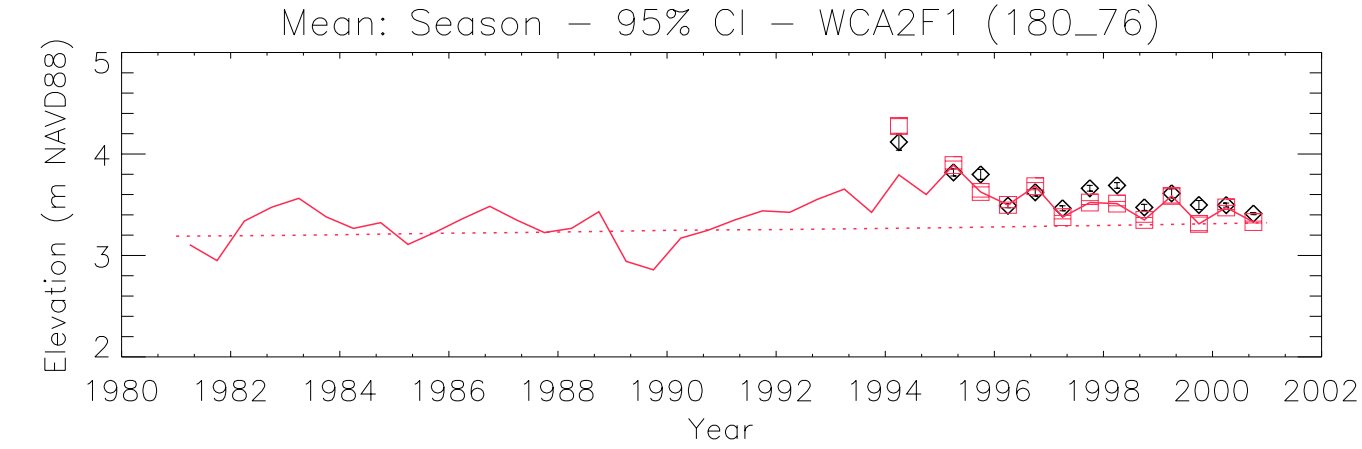

ELM2.8reg500 Raw Data (Obs. N = 6879) – 1–9 (195_53)
Elevation (m NAVD88)
6
5
4
3
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

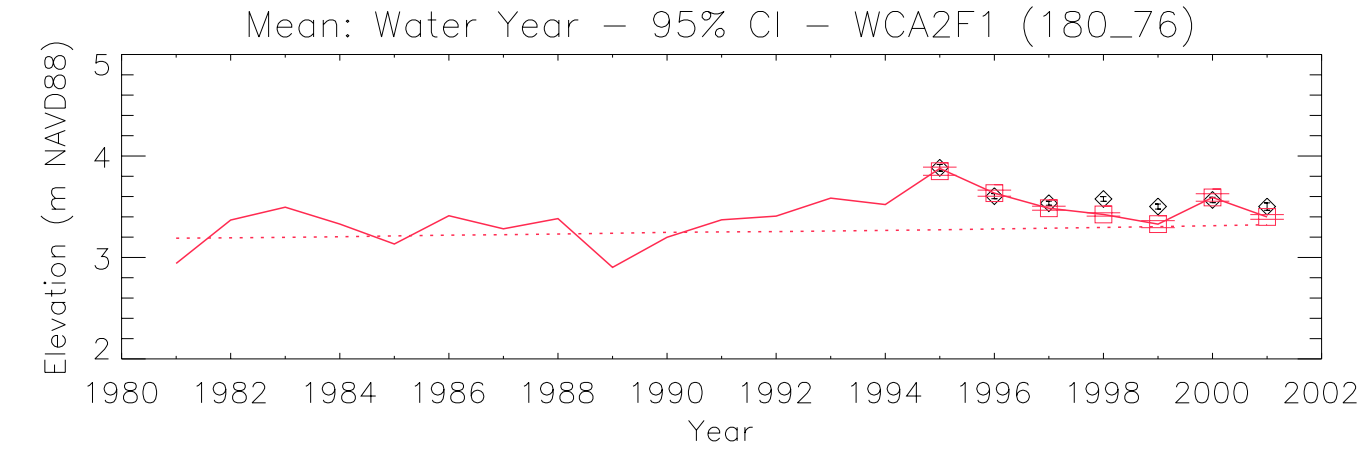

Mean: Season – 95% CI – 1–9 (195_53)
Elevation (m NAVD88)
6
5
4
3
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

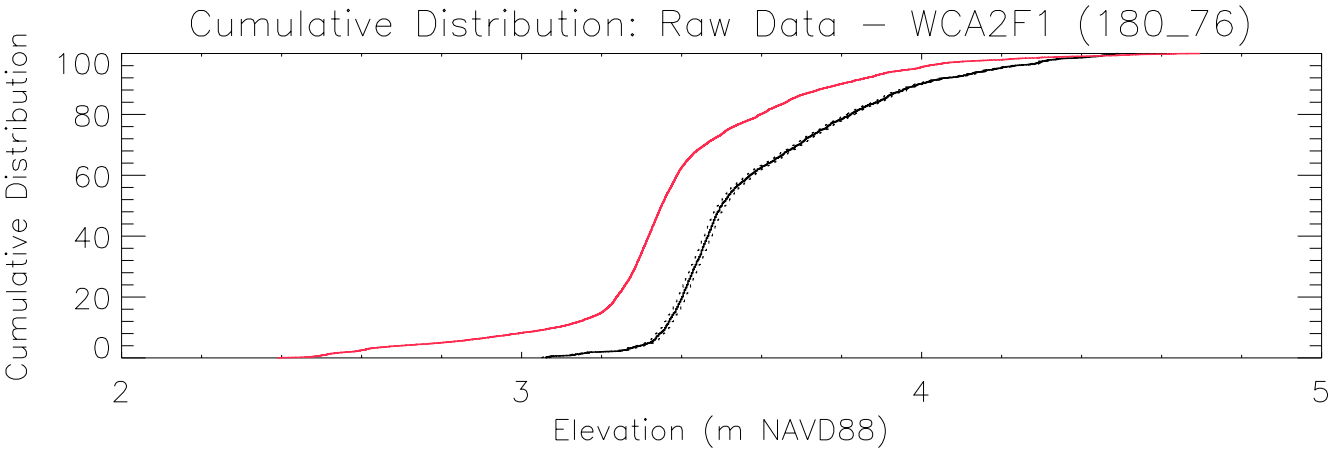

Mean: Water Year – 95% CI – 1–9 (195_53)
Elevation (m NAVD88)
6
5
4
3
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

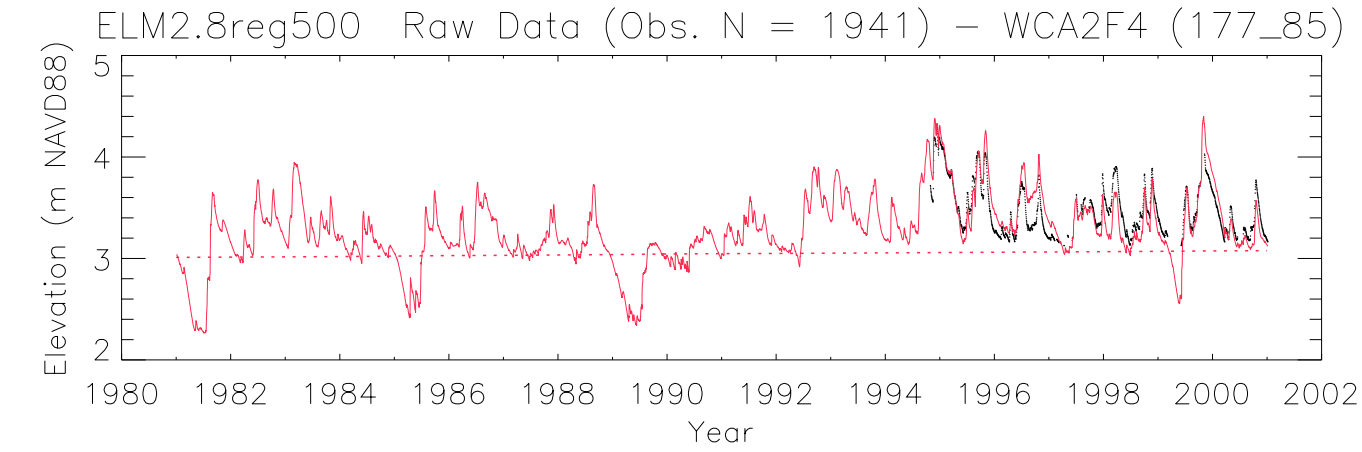

Cumulative Distribution: Raw Data – 1–9 (195_53)
Cumulative Distribution
100
80
60
40
20
0
3
4
5
6
Elevation (m NAVD88)

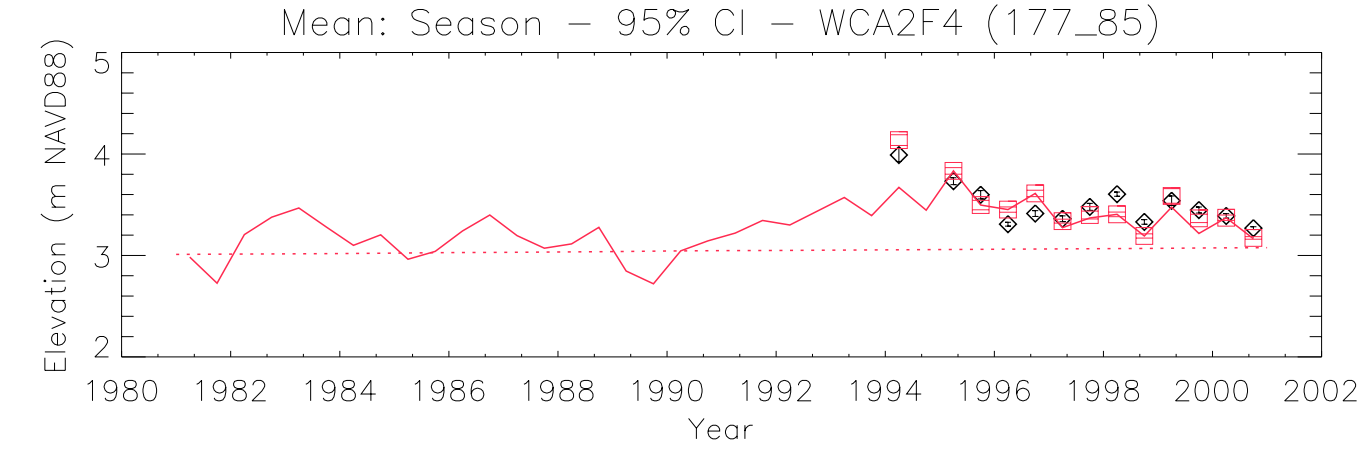

ELM2.8reg500 Raw Data (Obs. N = 2259) – WCA2F1 (180_76)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

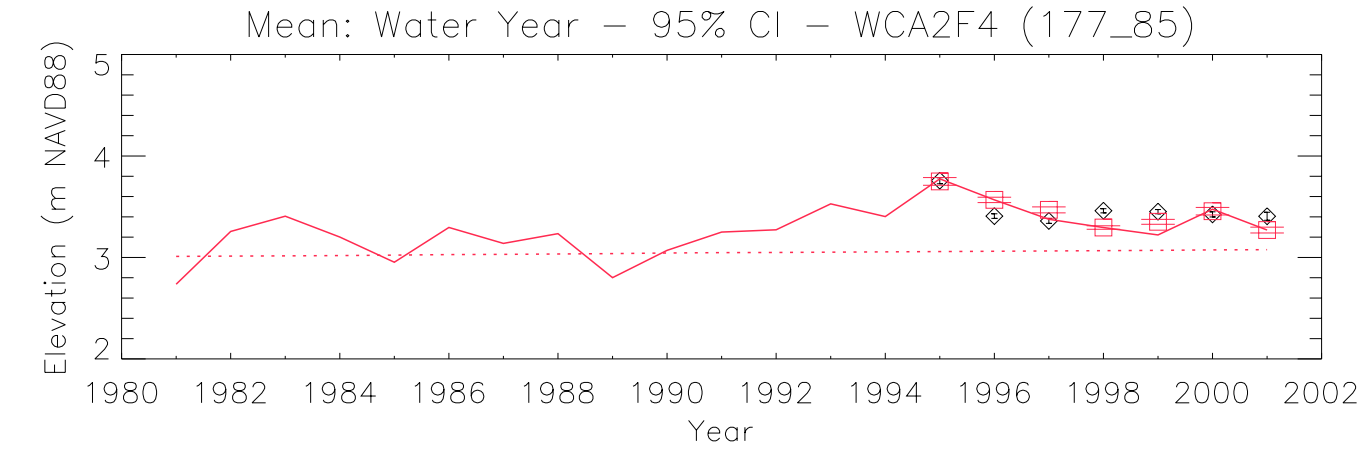

Mean: Season – 95% CI – WCA2F1 (180_76)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

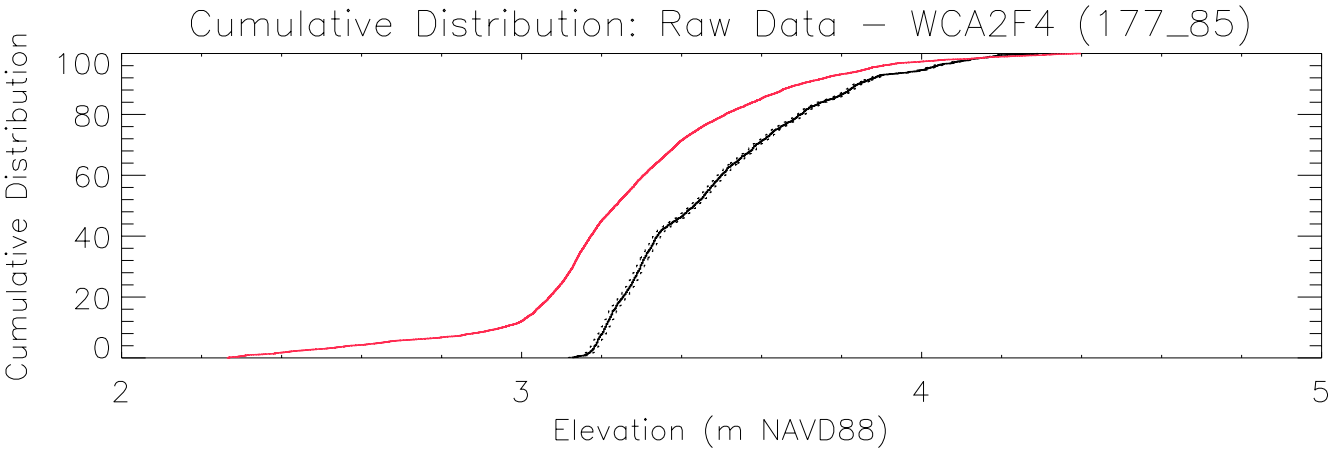

Mean: Water Year – 95% CI – WCA2F1 (180_76)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

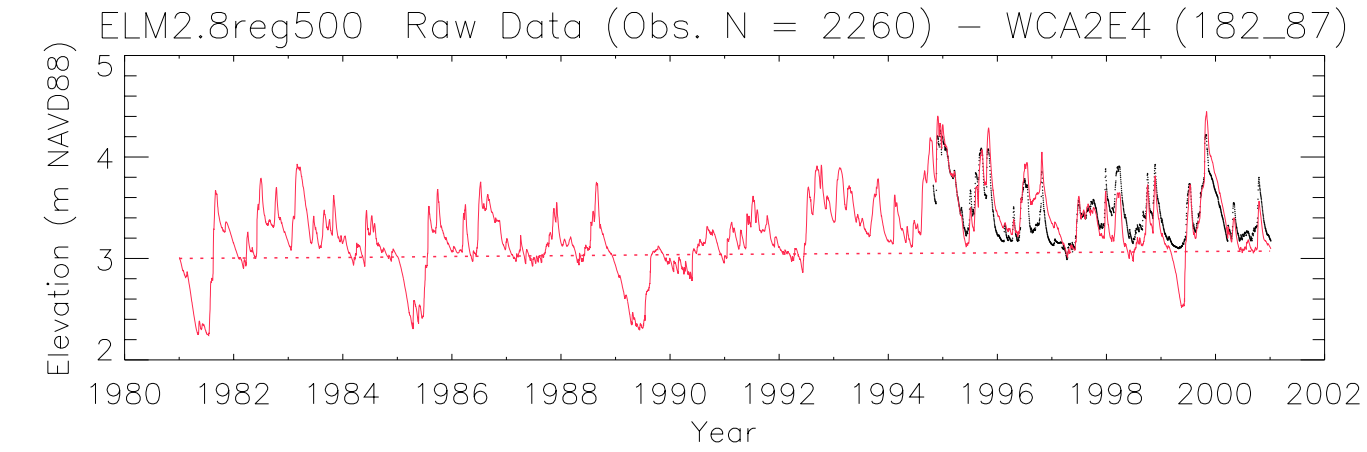

Cumulative Distribution: Raw Data – WCA2F1 (180_76)
Cumulative Distribution
100
80
60
40
20
0
2 3 4 5
Elevation (m NAVD88)

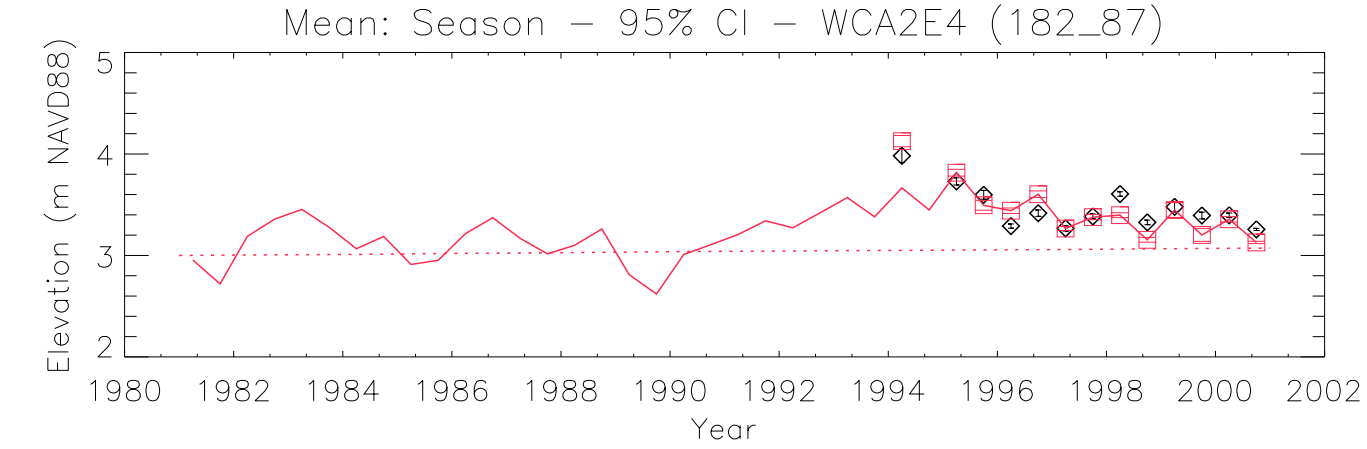

ELM2.8reg500 Raw Data (Obs. N = 1941) – WCA2F4 (177_85)
Elevation (m NAVD88)
5
4
3
2
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

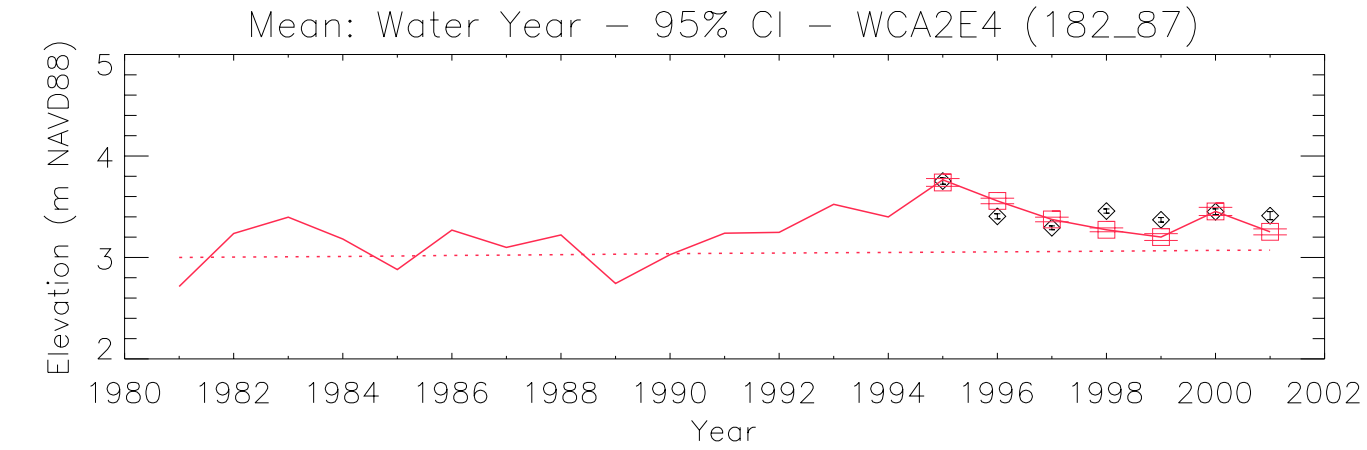

Mean: Season – 95% CI – WCA2F4 (177_85)
Elevation (m NAVD88)
5
4
3
2
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

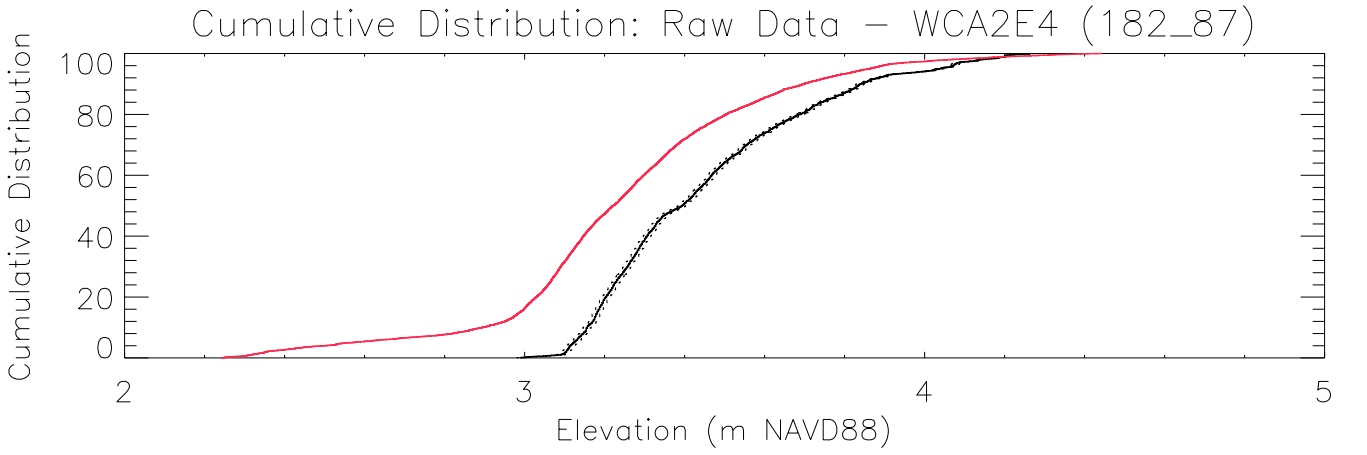

Mean: Water Year – 95% CI – WCA2F4 (177_85)
Elevation (m NAVD88)
5
4
3
2
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

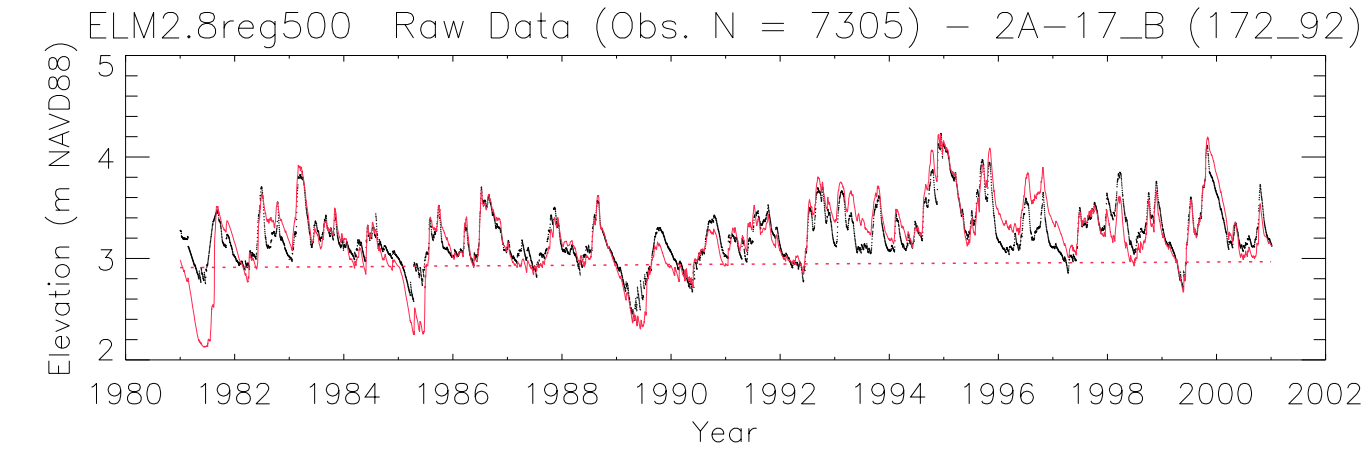

Cumulative Distribution: Raw Data – WCA2F4 (177_85)
Cumulative Distribution
100
80
60
40
20
0
2
3
4
5
Elevation (m NAVD88)

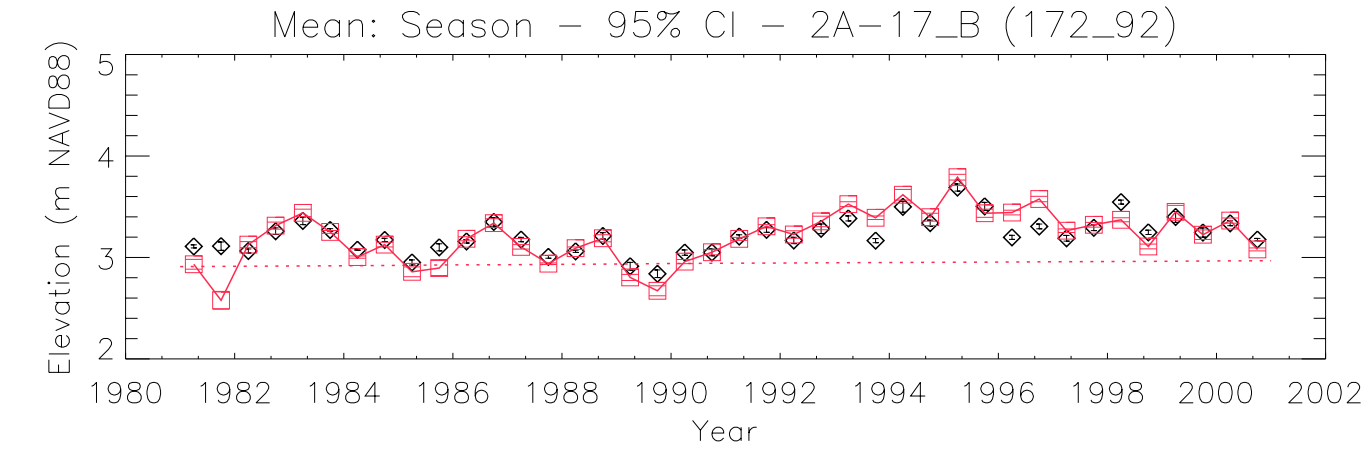

ELM2.8reg500 Raw Data (Obs. N = 2260) – WCA2E4 (182_87)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

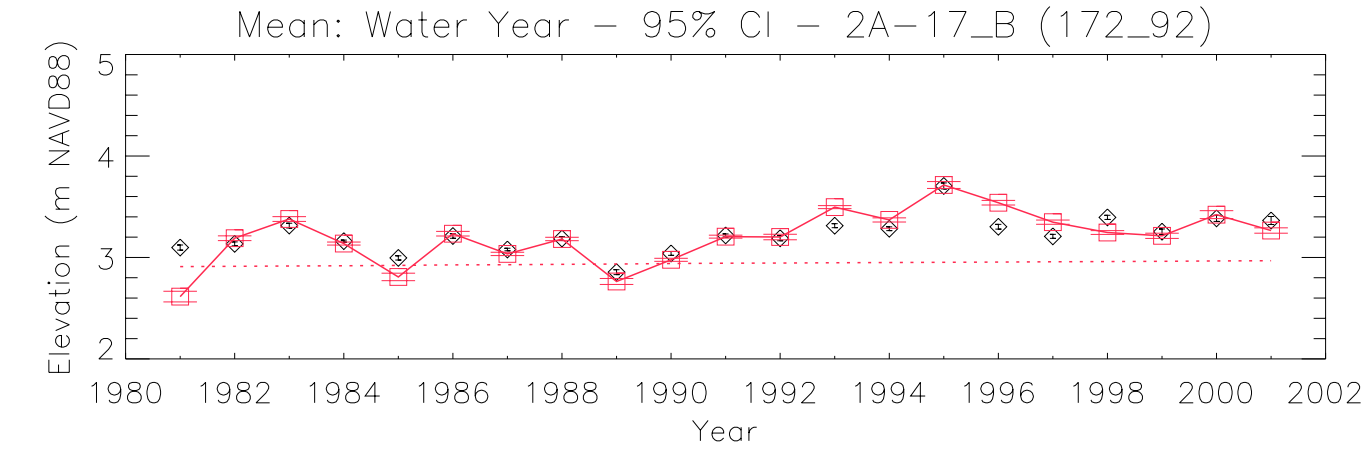

Mean: Season – 95% CI – WCA2E4 (182_87)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

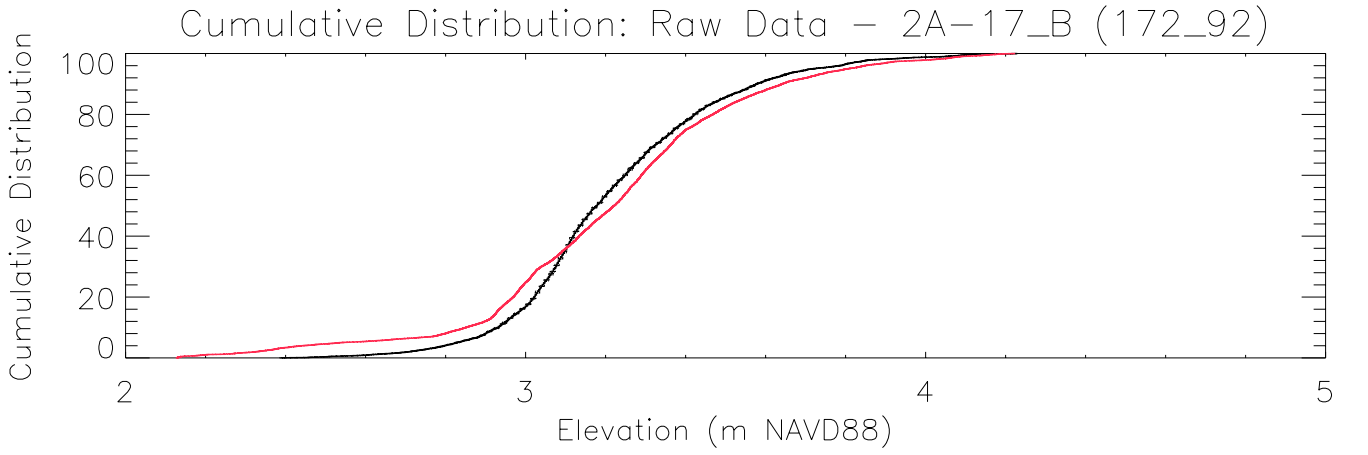

Mean: Water Year – 95% CI – WCA2E4 (182_87)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

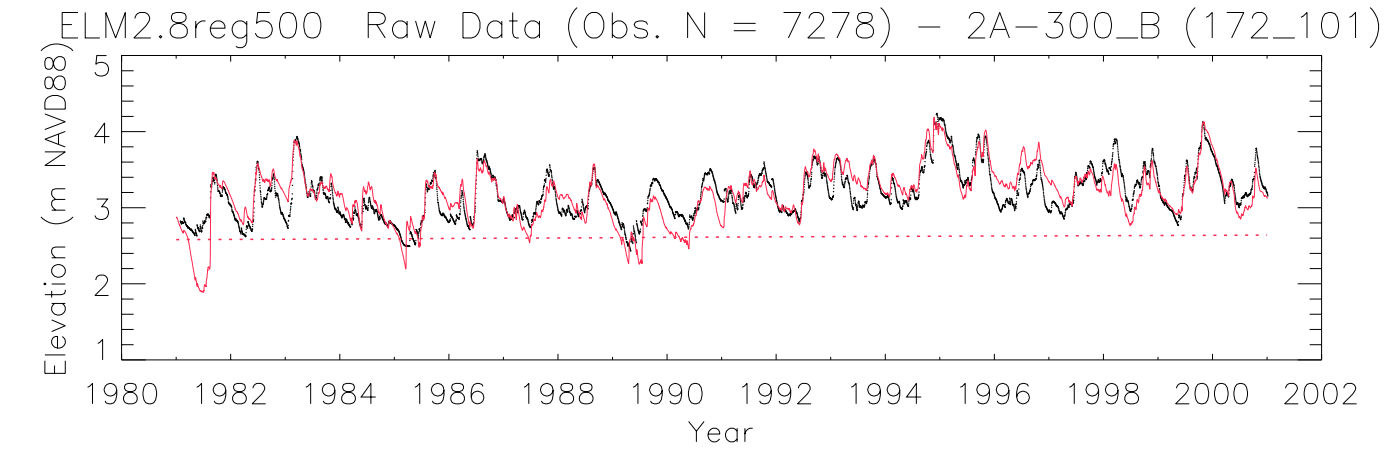

Cumulative Distribution: Raw Data – WCA2E4 (182_87)
Cumulative Distribution
100
80
60
40
20
0
2 3 4 5
Elevation (m NAVD88)

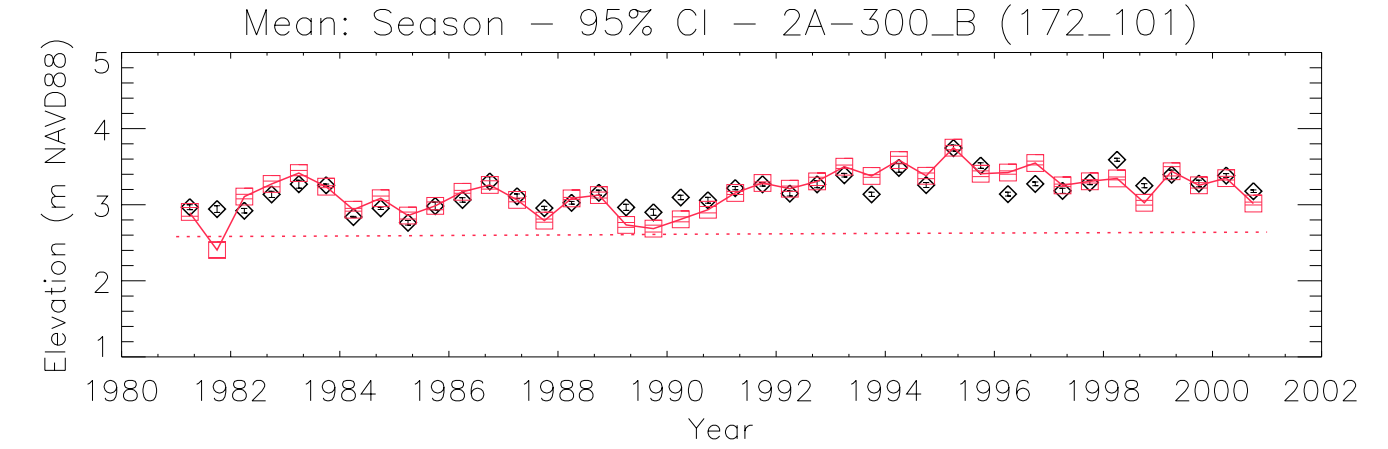

ELM2.8reg500 Raw Data (Obs. N = 7305) – 2A–17_B (172_92)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

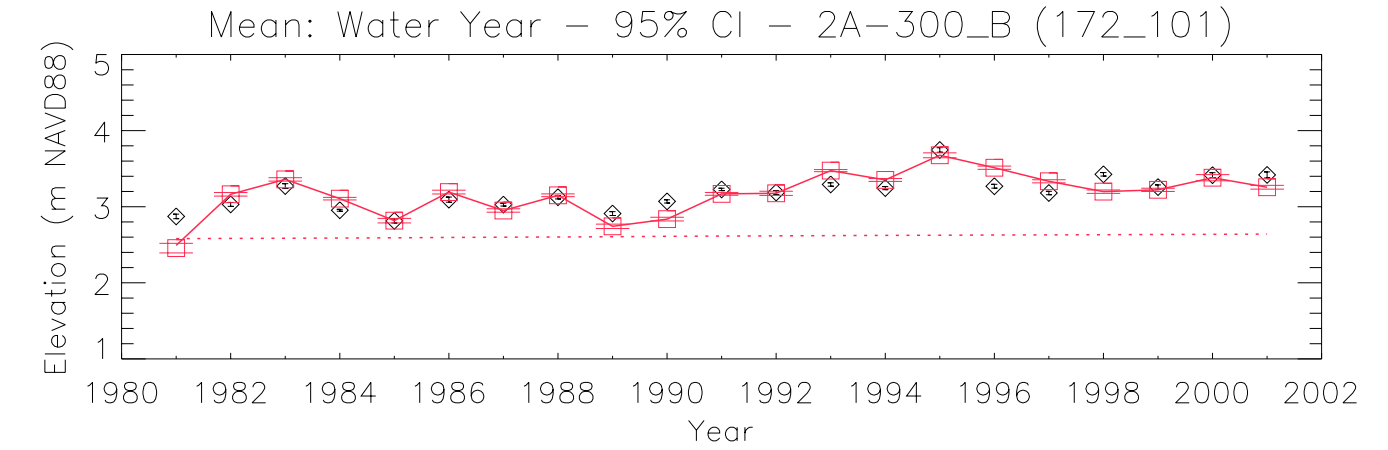

Mean: Season – 95% CI – 2A–17_B (172_92)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

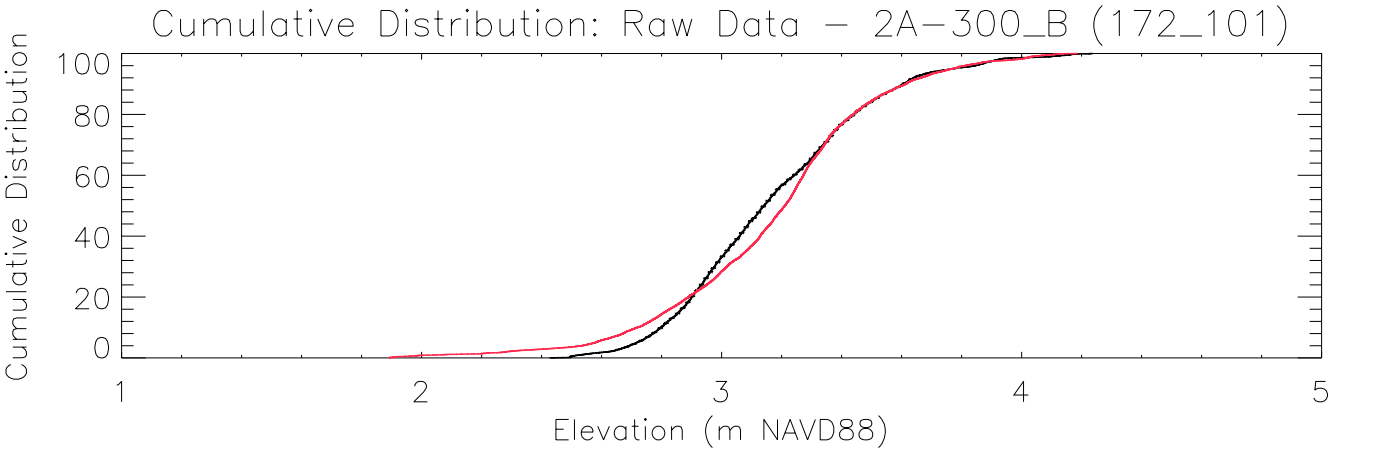

Mean: Water Year – 95% CI – 2A–17_B (172_92)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

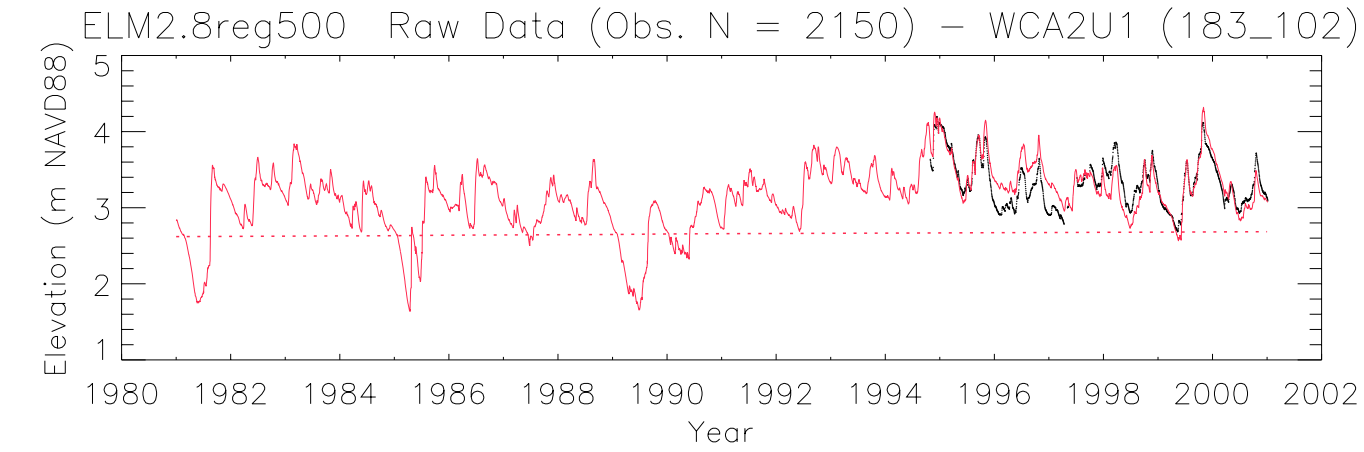

Cumulative Distribution: Raw Data – 2A–17_B (172_92)
Cumulative Distribution
100
80
60
40
20
0
2 3 4 5
Elevation (m NAVD88)

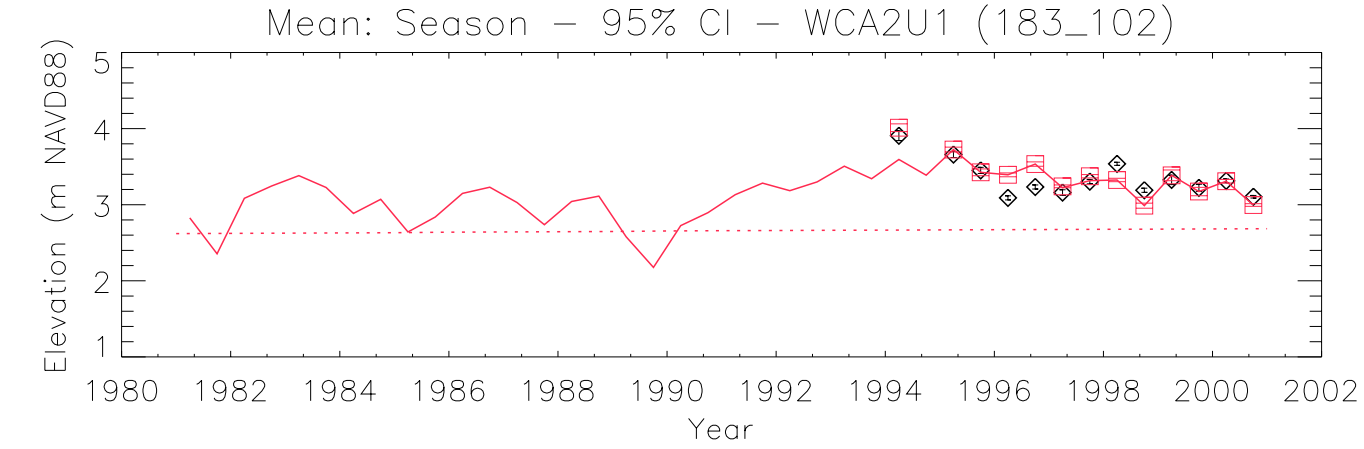

ELM2.8reg500 Raw Data (Obs. N = 7278) – 2A–300_B (172_101)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

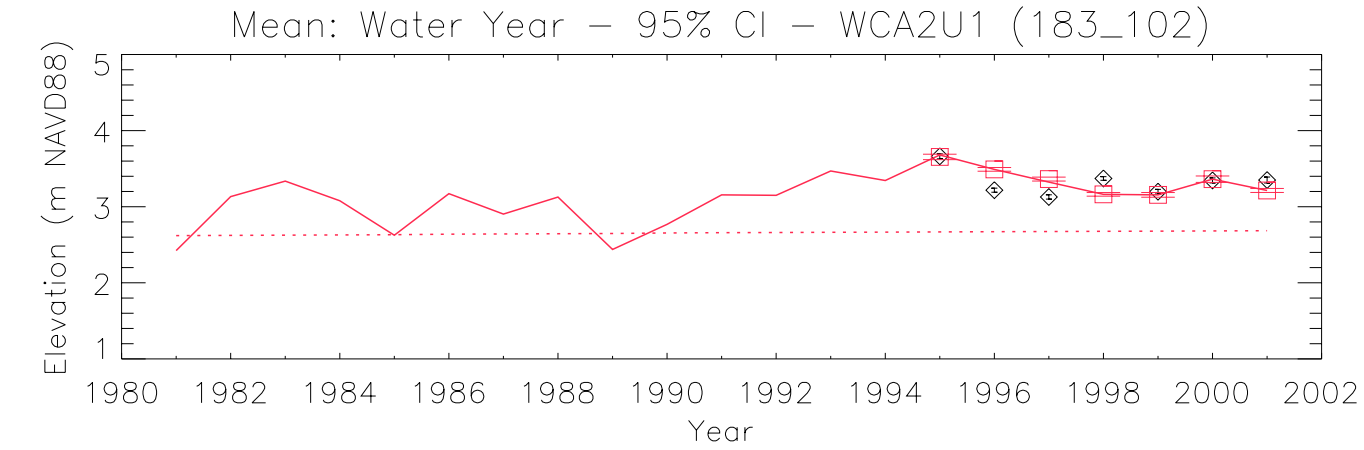

Mean: Season – 95% CI – 2A–300_B (172_101)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

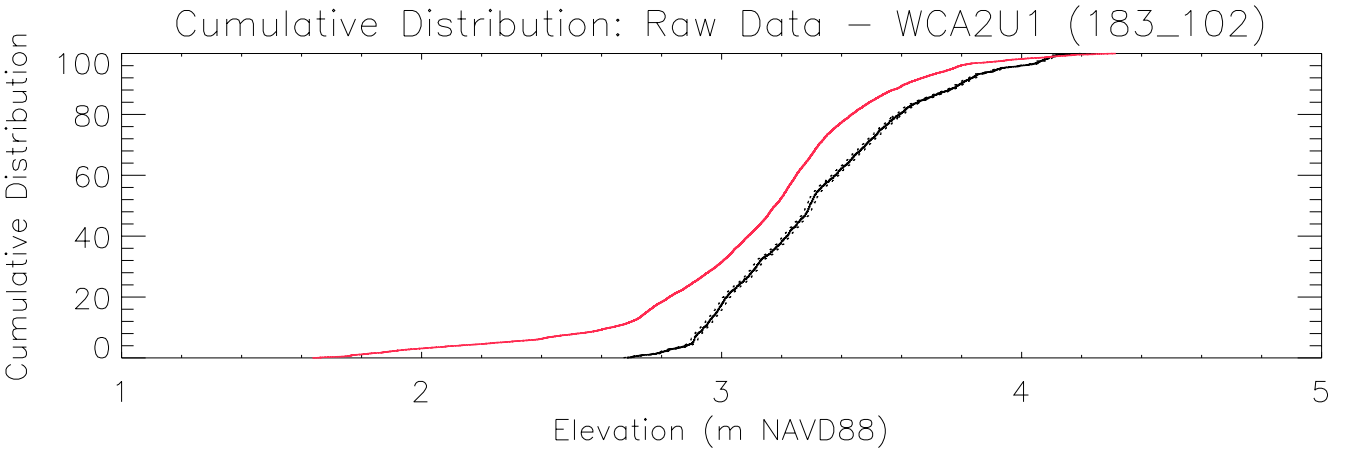

Mean: Water Year – 95% CI – 2A–300_B (172_101)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

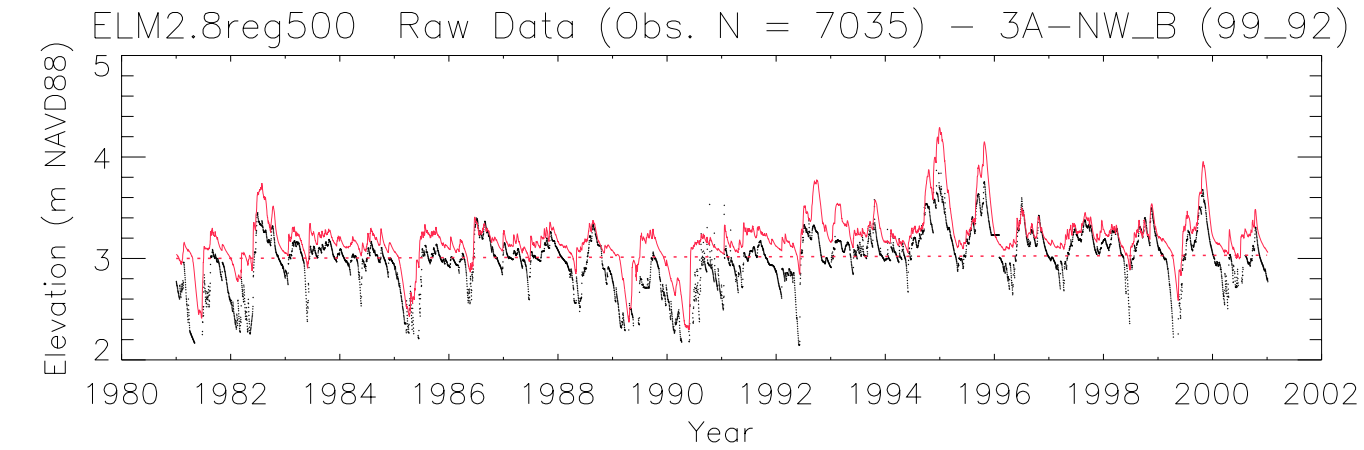

Cumulative Distribution: Raw Data – 2A–300_B (172_101)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
5
Elevation (m NAVD88)

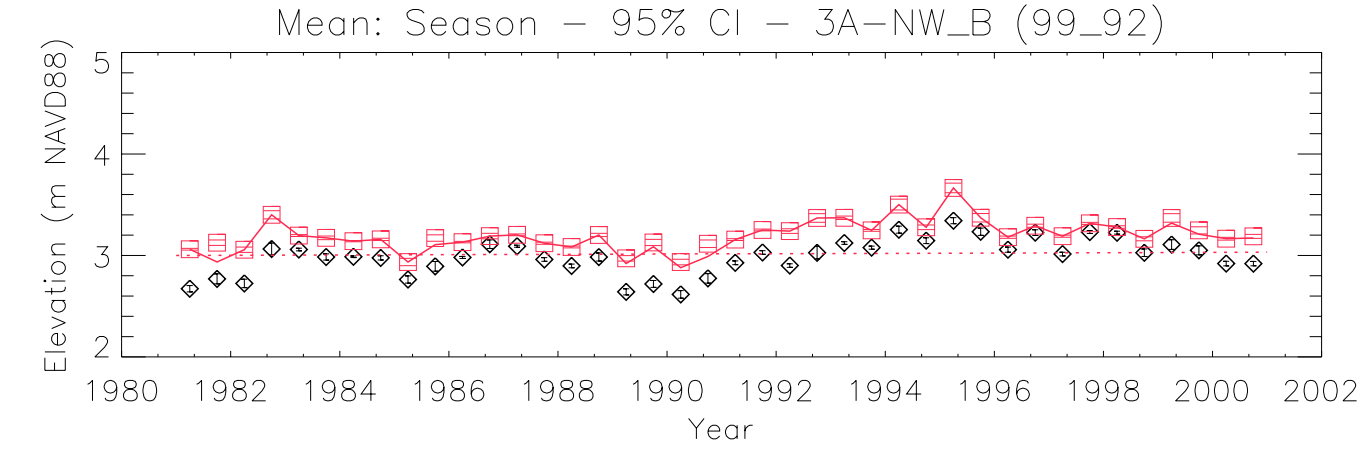

ELM2.8reg500 Raw Data (Obs. N = 2150) – WCA2U1 (183_102)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

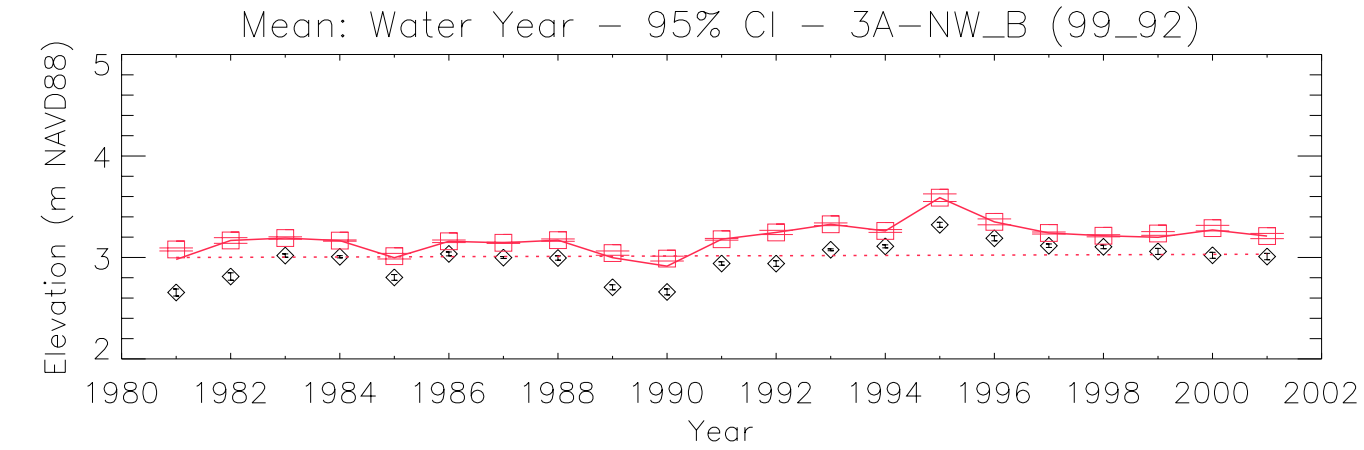

Mean: Season – 95% CI – WCA2U1 (183_102)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

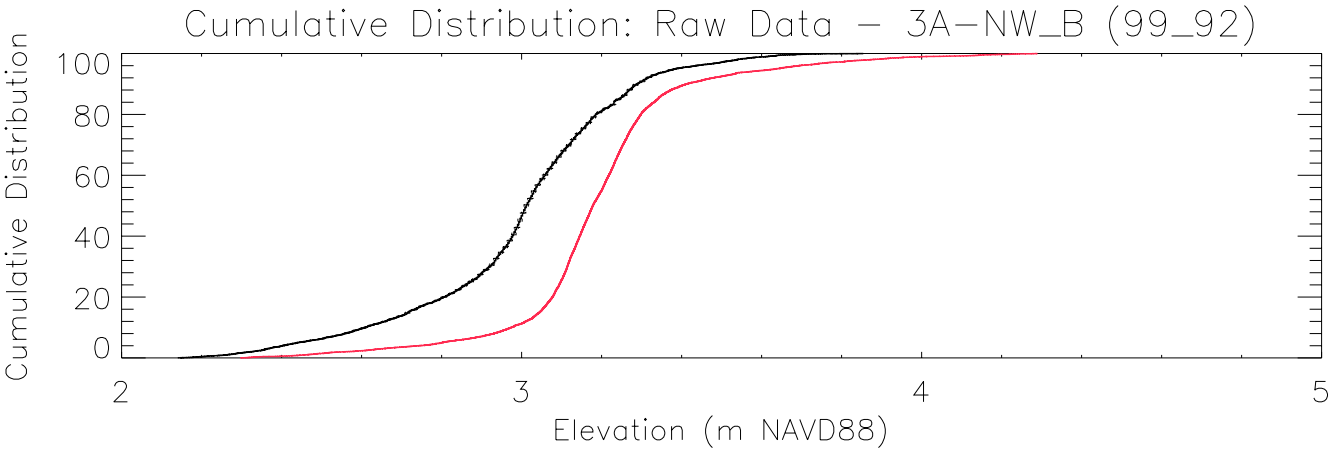

Mean: Water Year – 95% CI – WCA2U1 (183_102)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

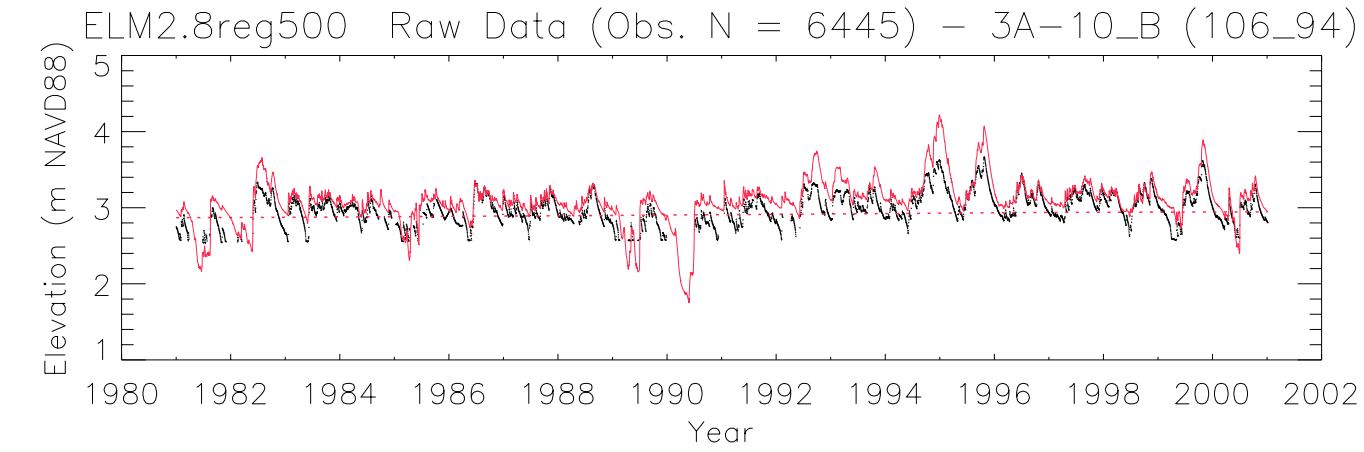

Cumulative Distribution: Raw Data – WCA2U1 (183_102)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
5
Elevation (m NAVD88)

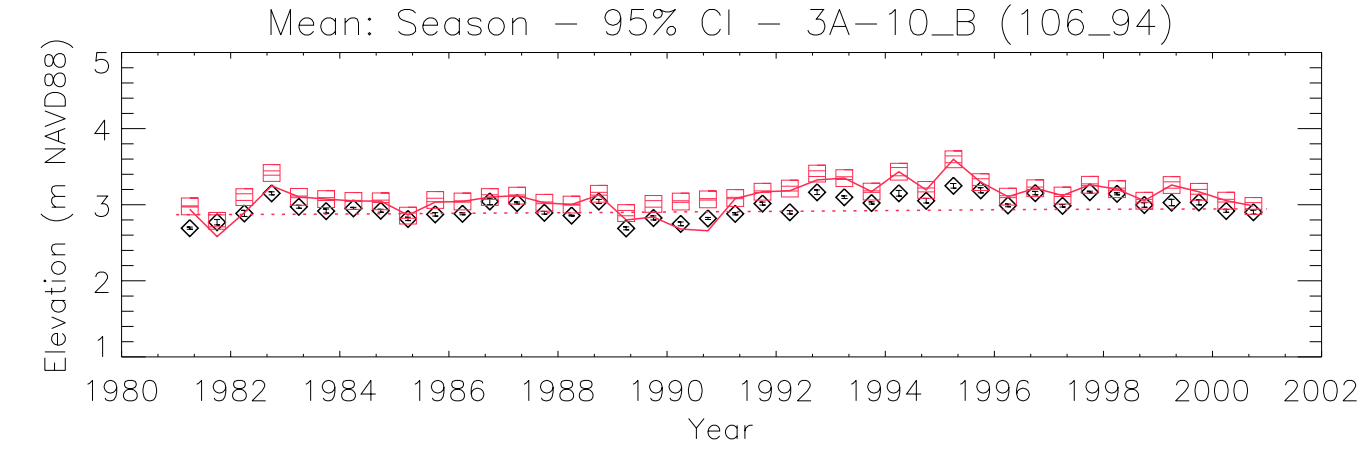

ELM2.8reg500 Raw Data (Obs. N = 7035) – 3A–NW_B (99_92)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

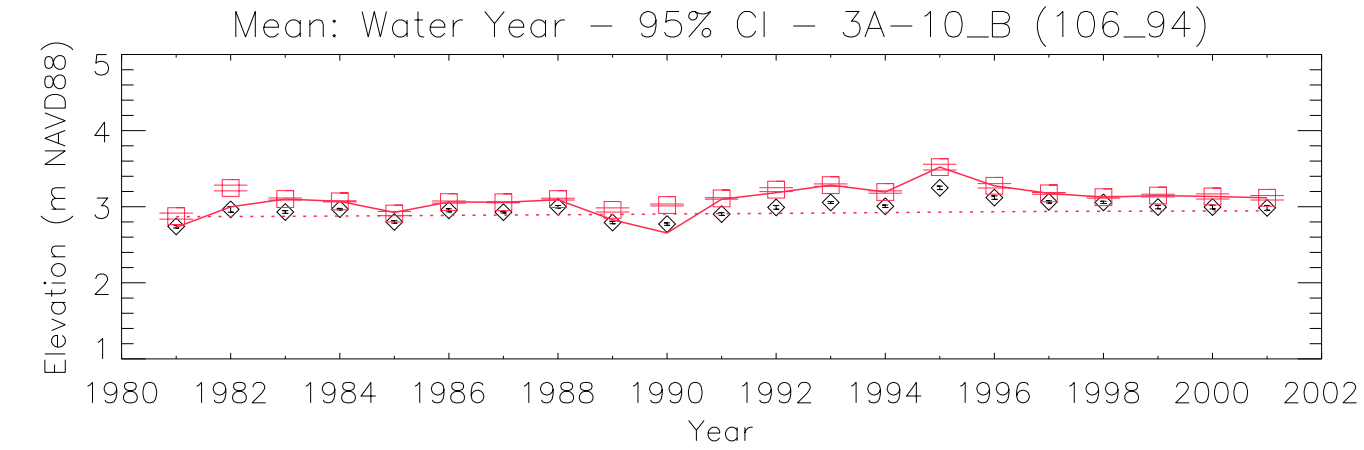

Mean: Season – 95% CI – 3A–NW_B (99_92)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

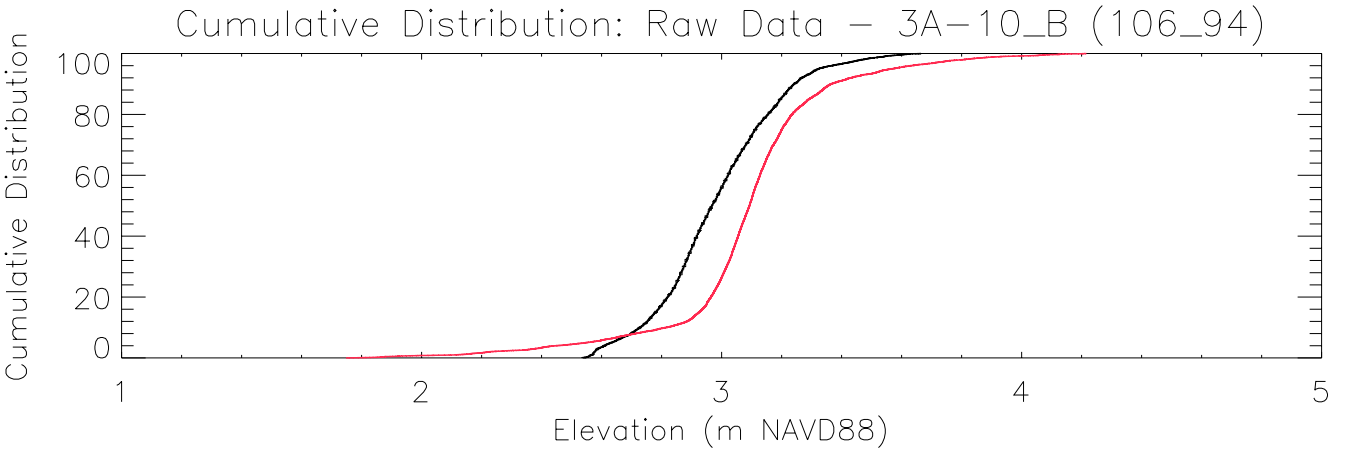

Mean: Water Year – 95% CI – 3A–NW_B (99_92)
Elevation (m NAVD88)
5
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

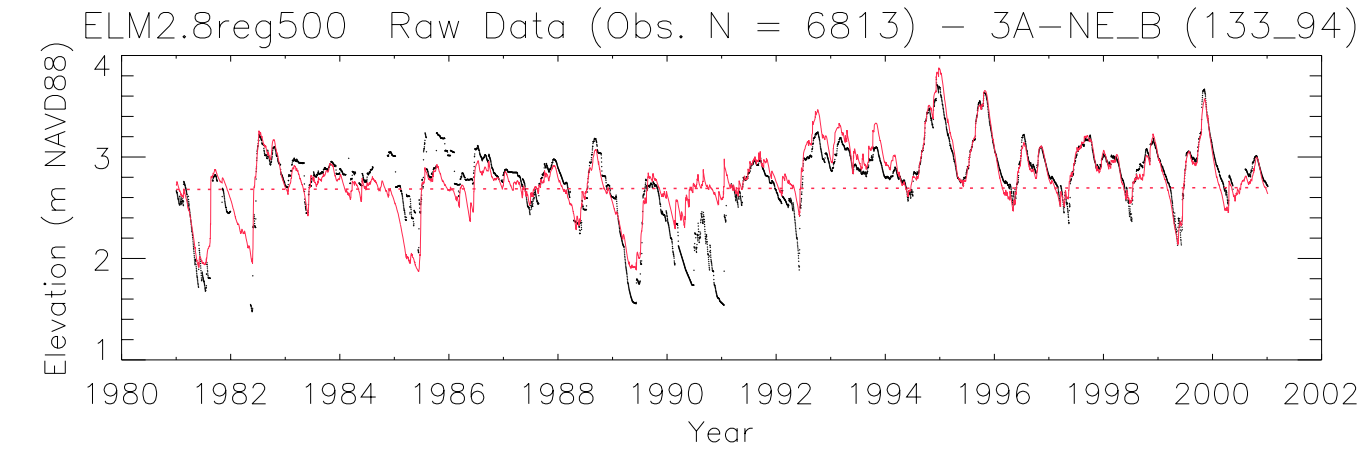

Cumulative Distribution: Raw Data – 3A–NW_B (99_92)
Cumulative Distribution
100
80
60
40
20
0
2
3
4
5
Elevation (m NAVD88)

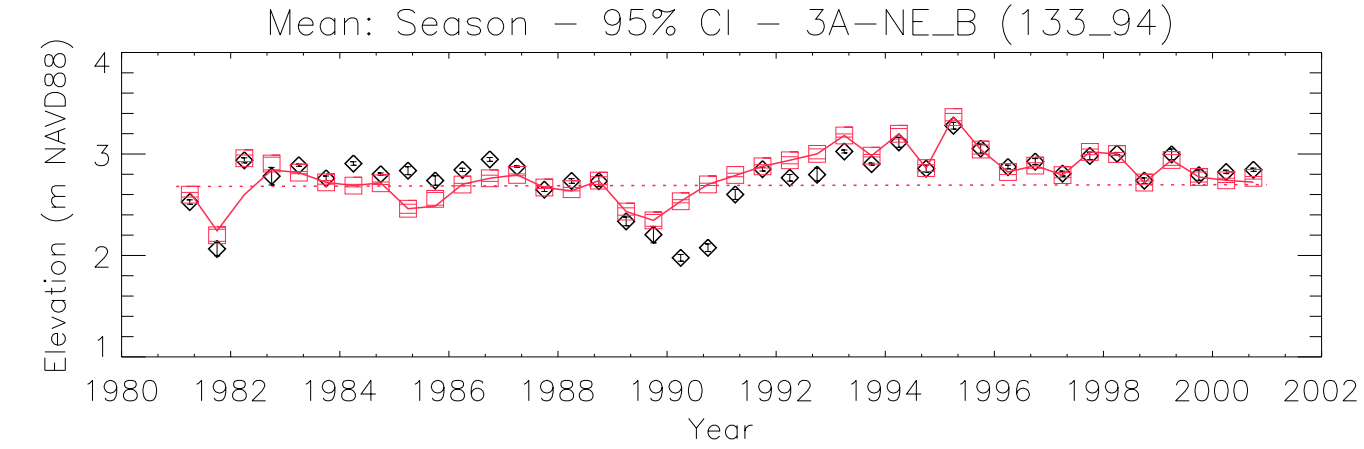

ELM2.8reg500 Raw Data (Obs. N = 6445) – 3A–10_B (106_94)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

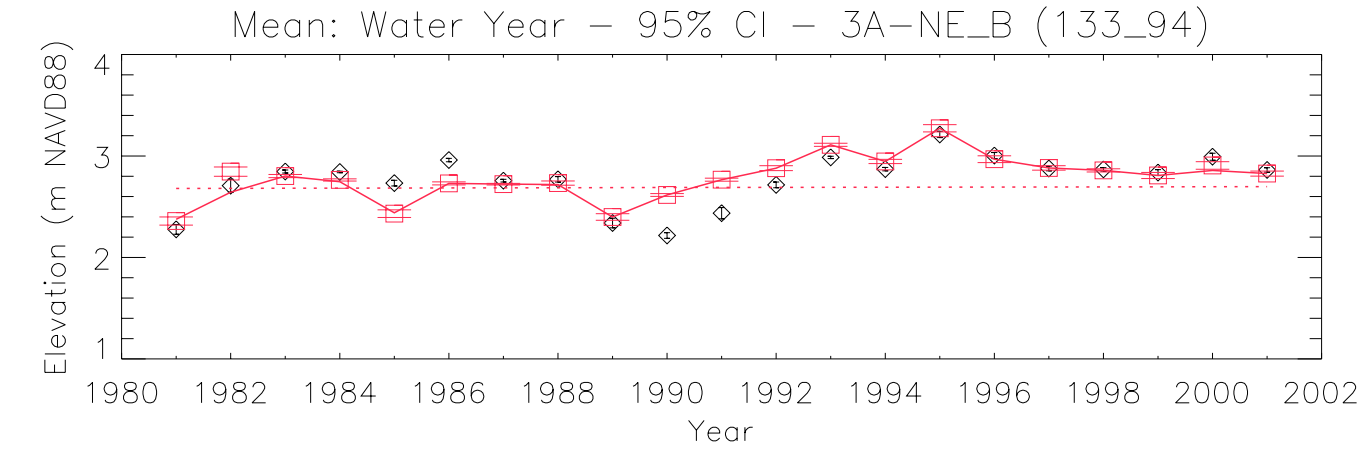

Mean: Season – 95% CI – 3A–10_B (106_94)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

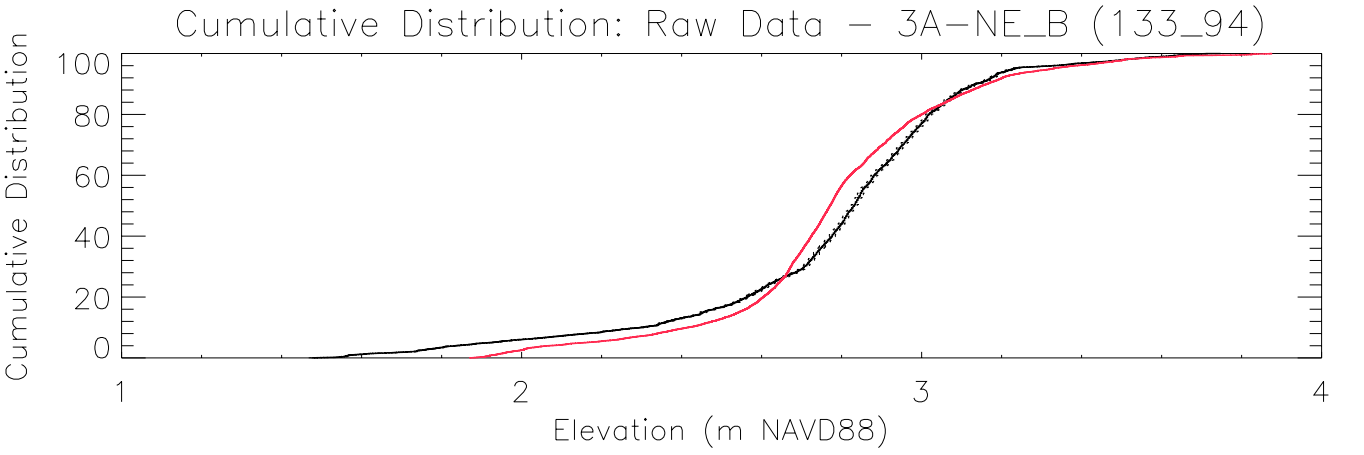

Mean: Water Year – 95% CI – 3A–10_B (106_94)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

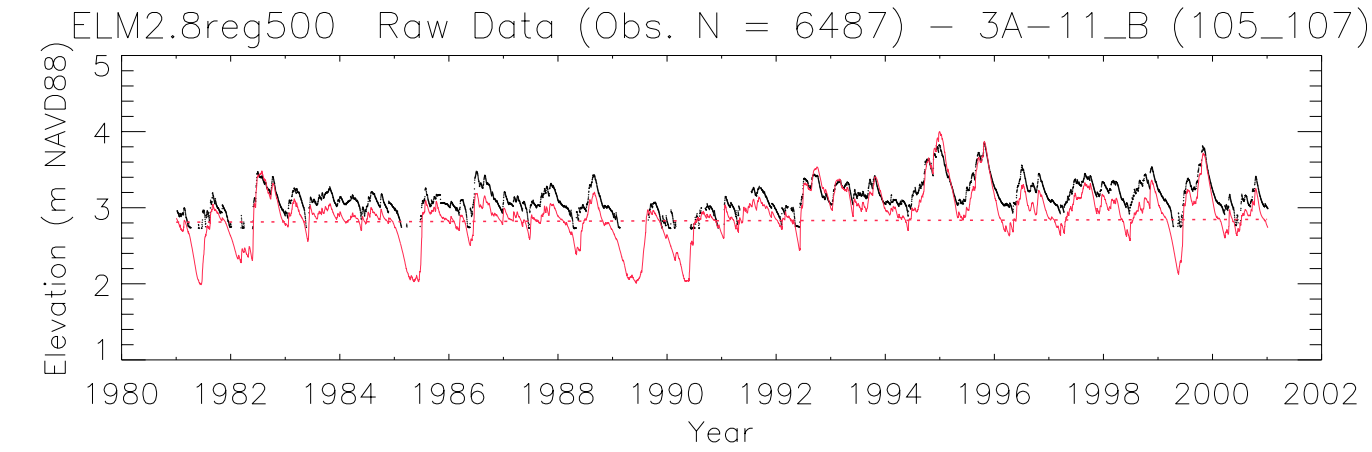

Cumulative Distribution: Raw Data – 3A–10_B (106_94)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
5
Elevation (m NAVD88)

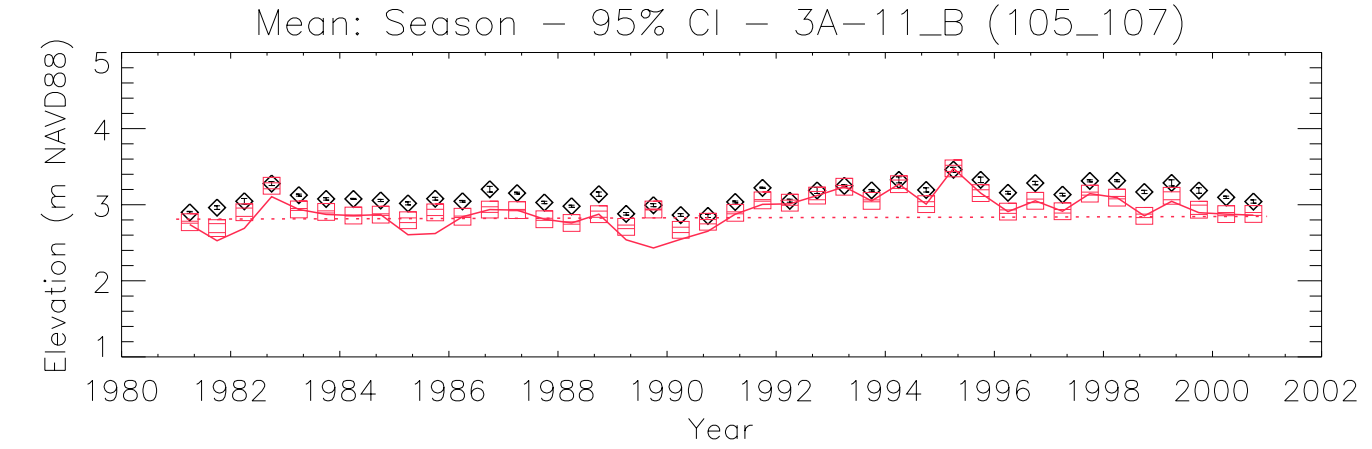

ELM2.8reg500 Raw Data (Obs. N = 6813) – 3A–NE_B (133_94)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

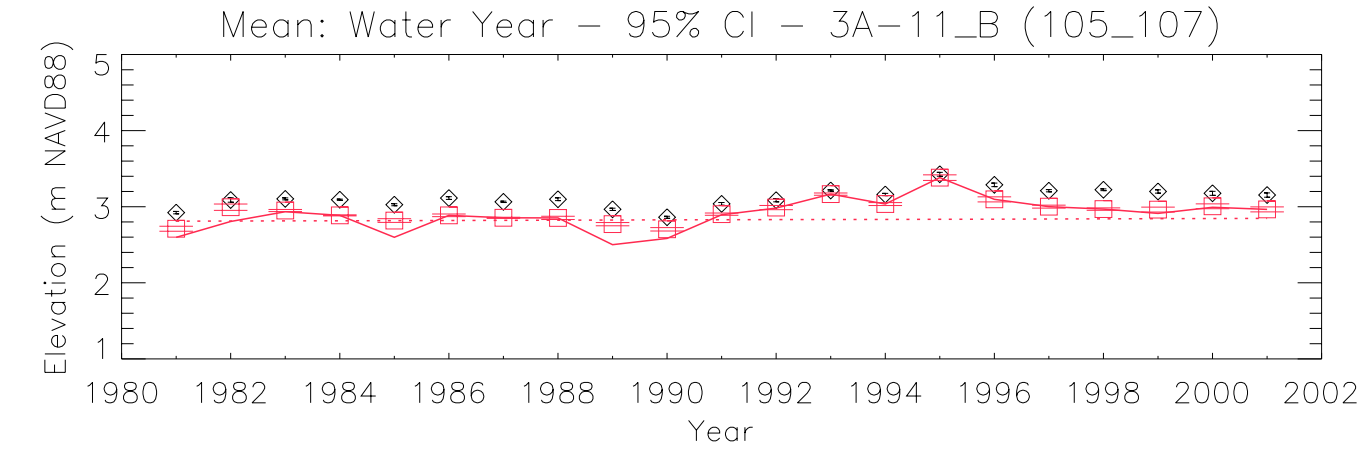

Mean: Season – 95% CI – 3A–NE_B (133_94)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

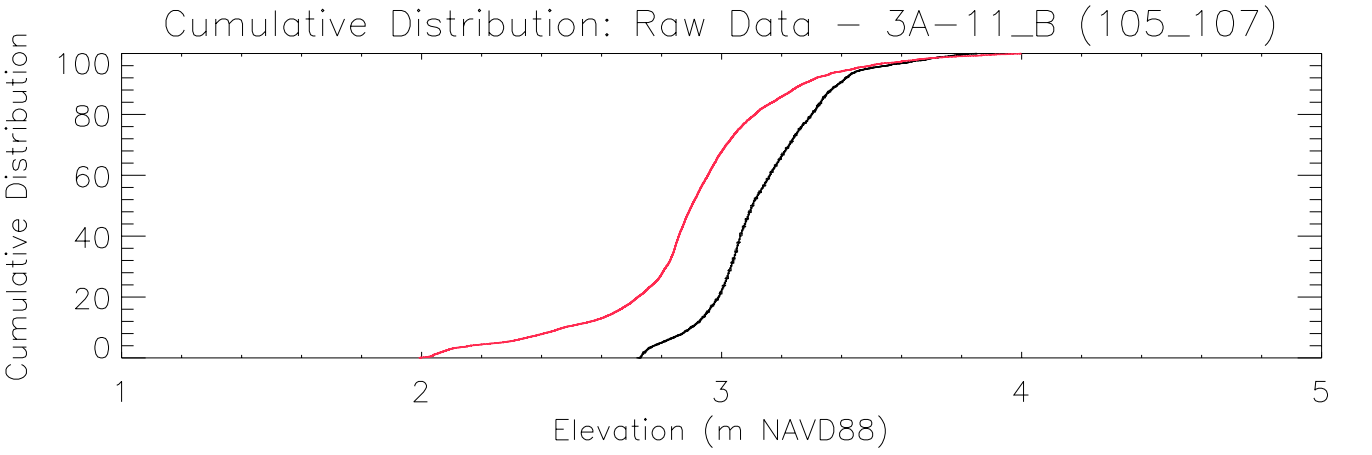

Mean: Water Year – 95% CI – 3A–NE_B (133_94)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

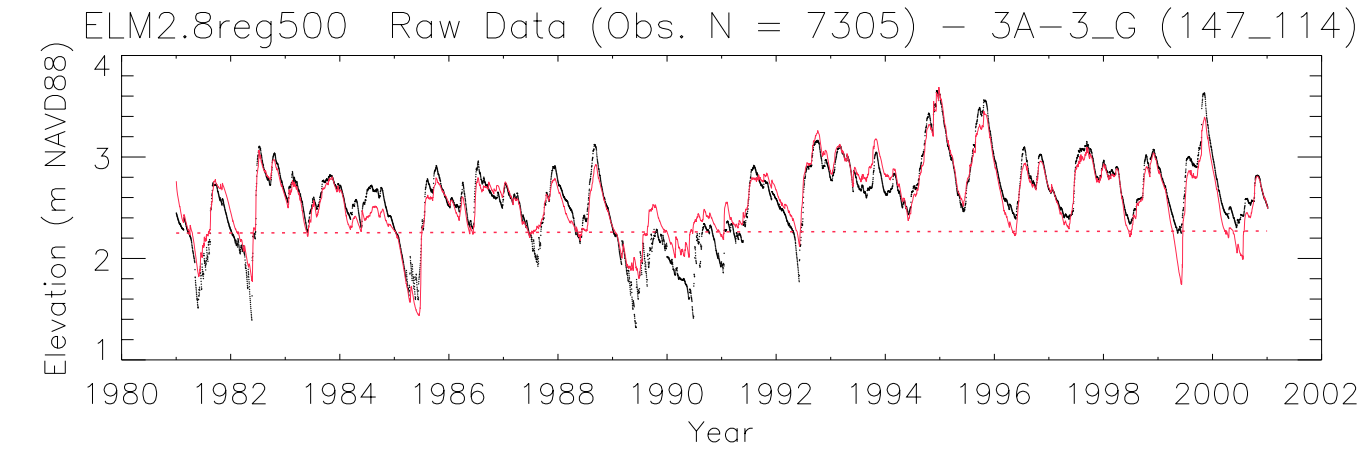

Cumulative Distribution: Raw Data – 3A–NE_B (133_94)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

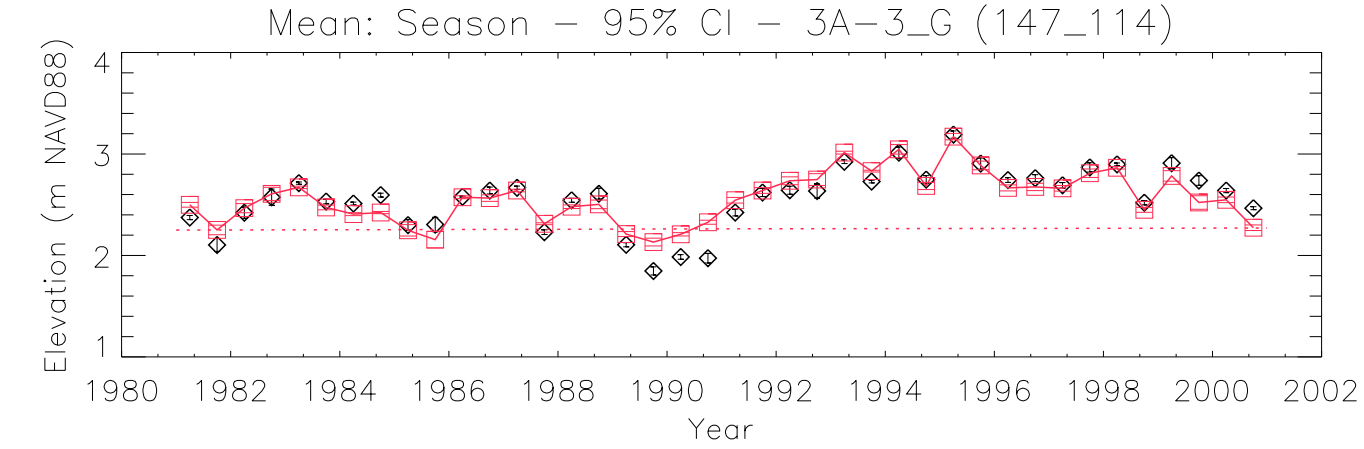

ELM2.8reg500 Raw Data (Obs. N = 6487) – 3A–11_B (105_107)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

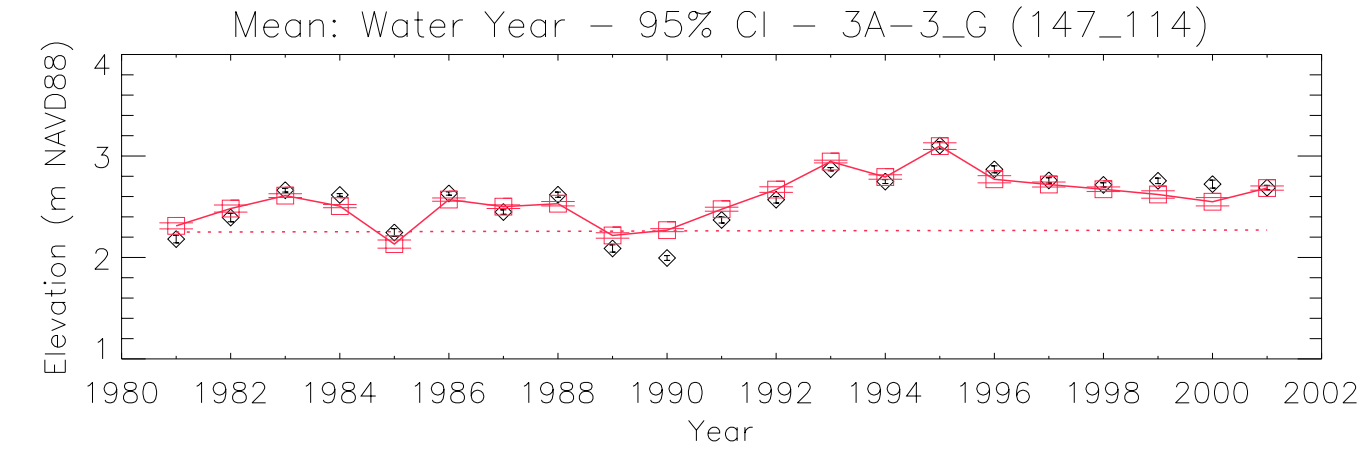

Mean: Season – 95% CI – 3A–11_B (105_107)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

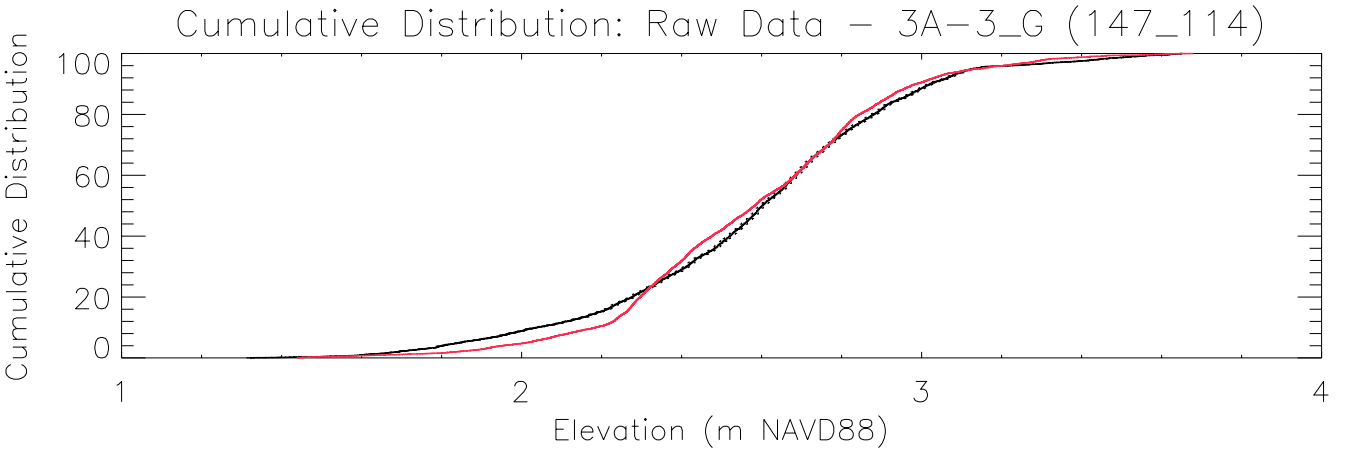

Mean: Water Year – 95% CI – 3A–11_B (105_107)
Elevation (m NAVD88)
1
2
3
4
5
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

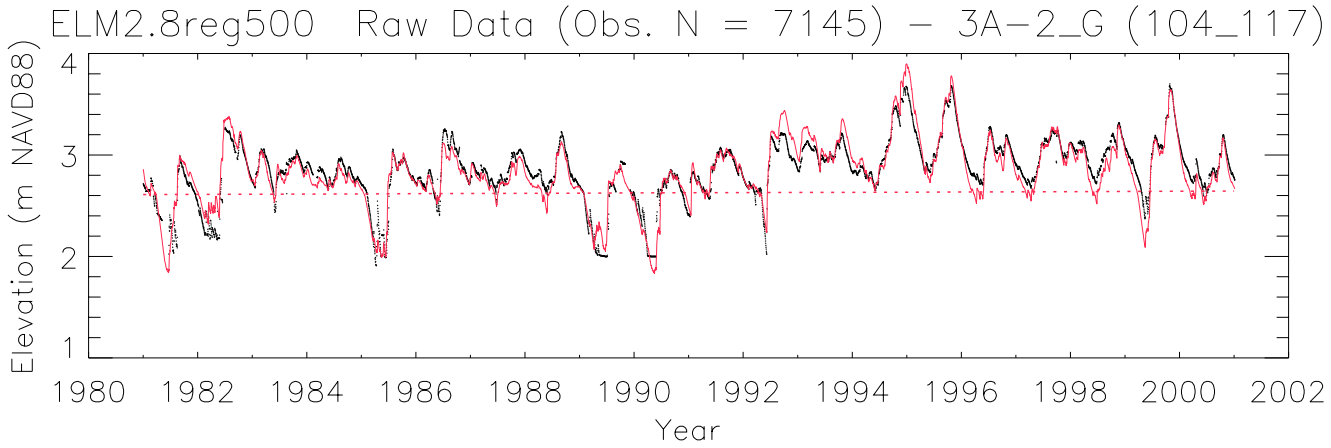

Cumulative Distribution: Raw Data – 3A–11_B (105_107)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
5
Elevation (m NAVD88)

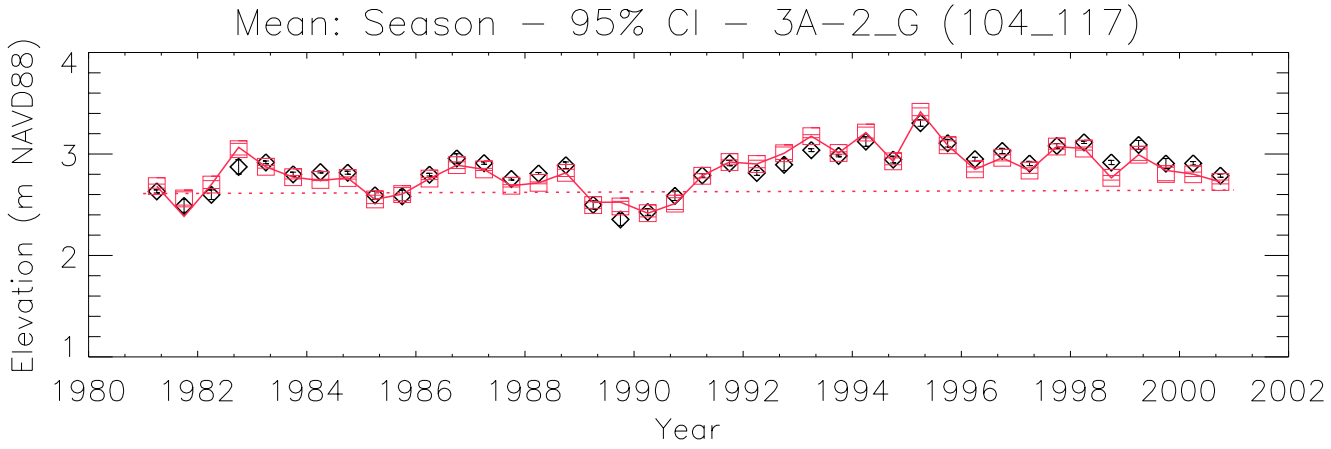

ELM2.8reg500 Raw Data (Obs. N = 7305) – 3A–3_G (147_114)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

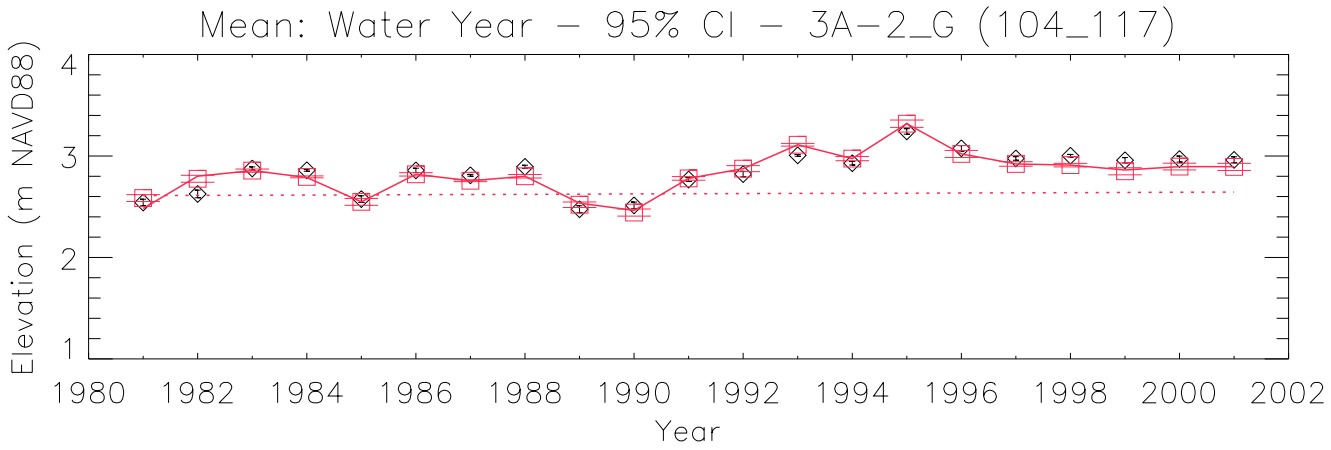

Mean: Season – 95% CI – 3A–3_G (147_114)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

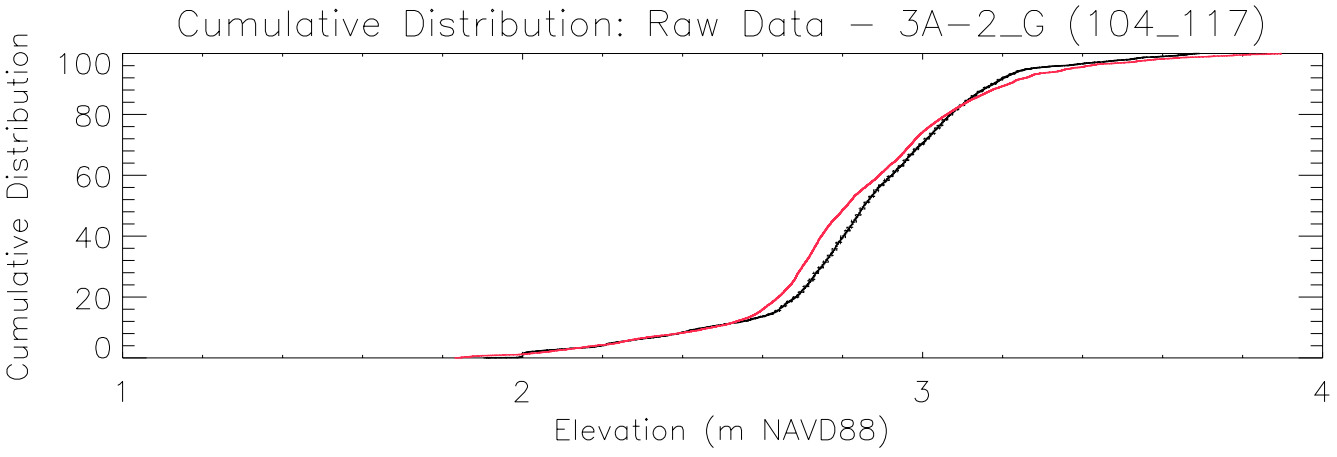

Mean: Water Year – 95% CI – 3A–3_G (147_114)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

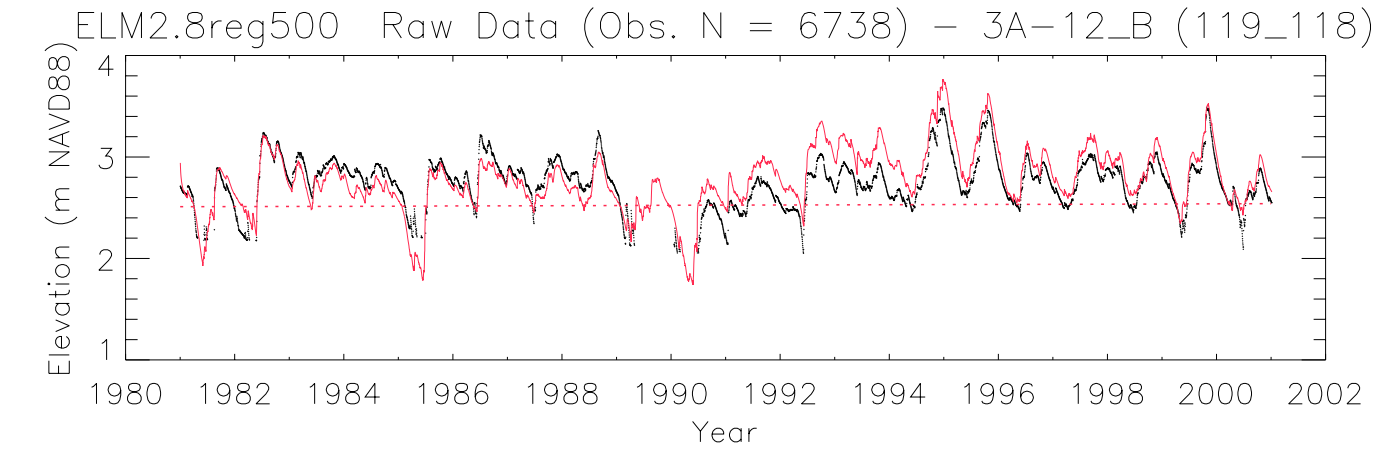

Cumulative Distribution: Raw Data – 3A–3_G (147_114)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

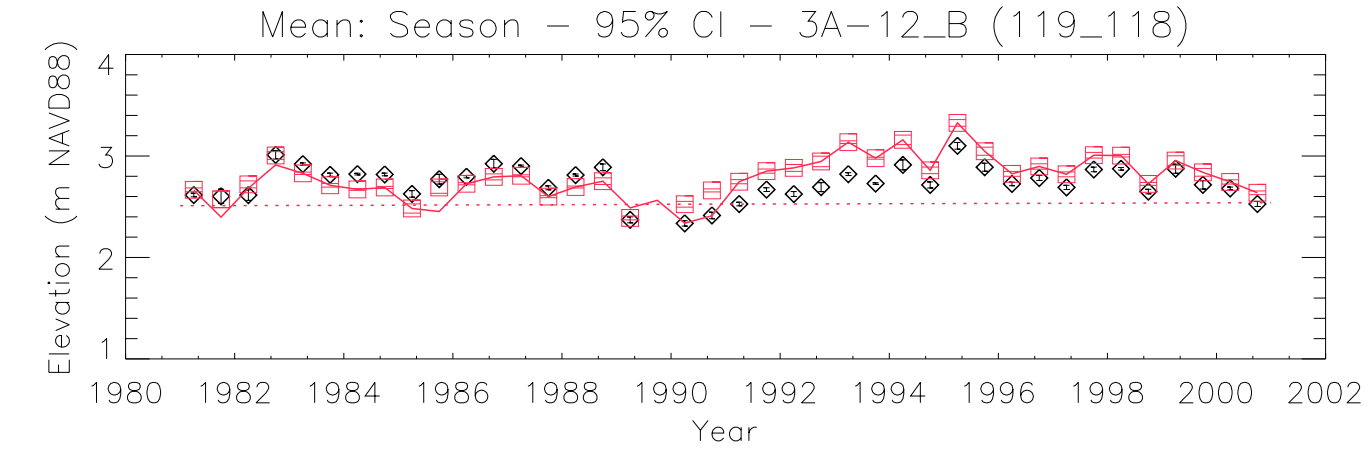

ELM2.8reg500 Raw Data (Obs. N = 7145) – 3A–2_G (104_117)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

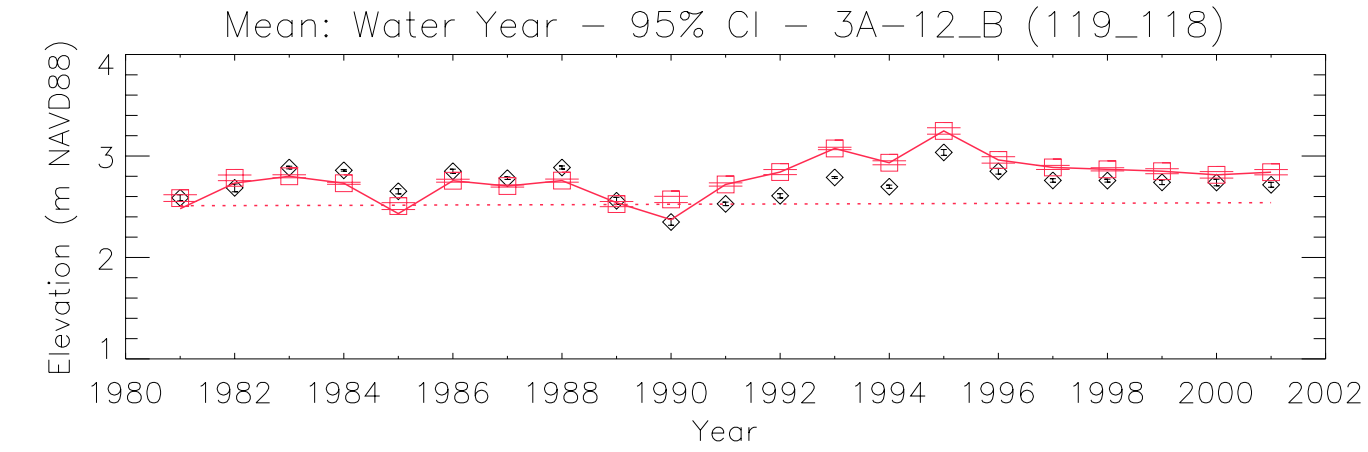

Mean: Season – 95% CI – 3A–2_G (104_117)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

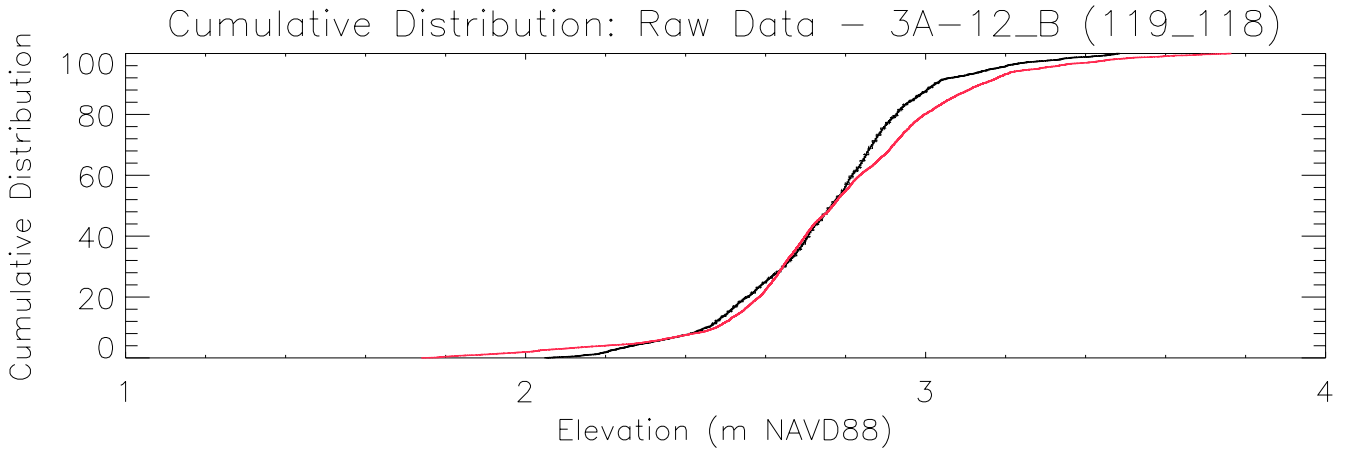

Mean: Water Year – 95% CI – 3A–2_G (104_117)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

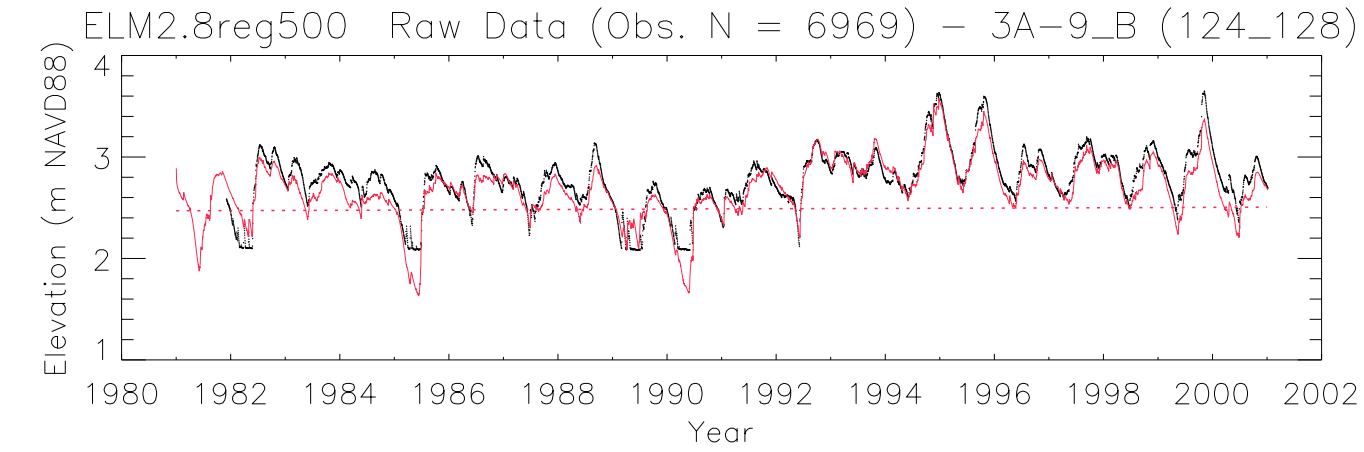

Cumulative Distribution: Raw Data – 3A–2_G (104_117)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

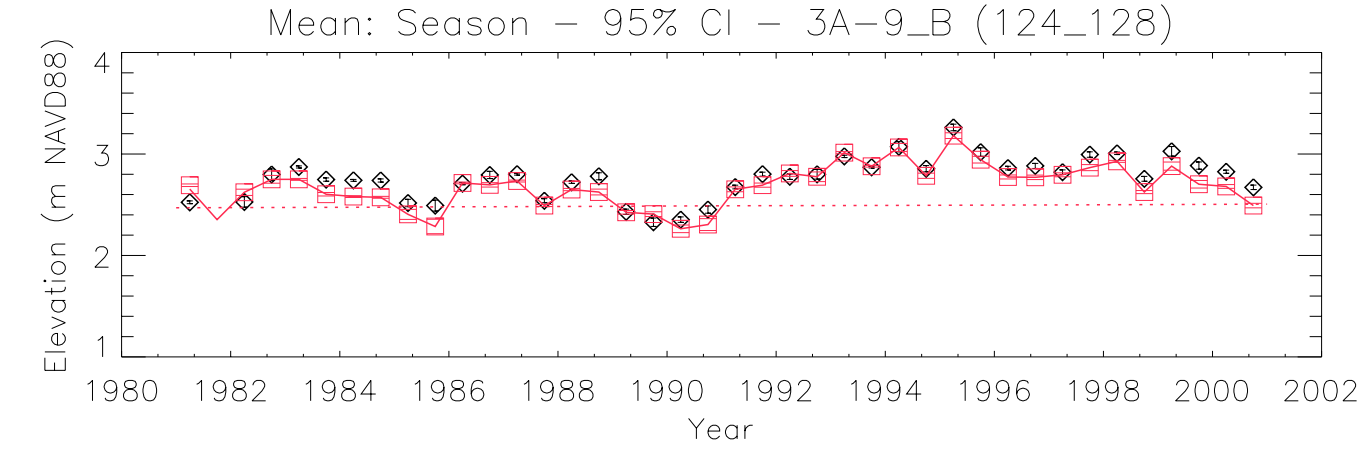

ELM2.8reg500 Raw Data (Obs. N = 6738) – 3A–12_B (119_118)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

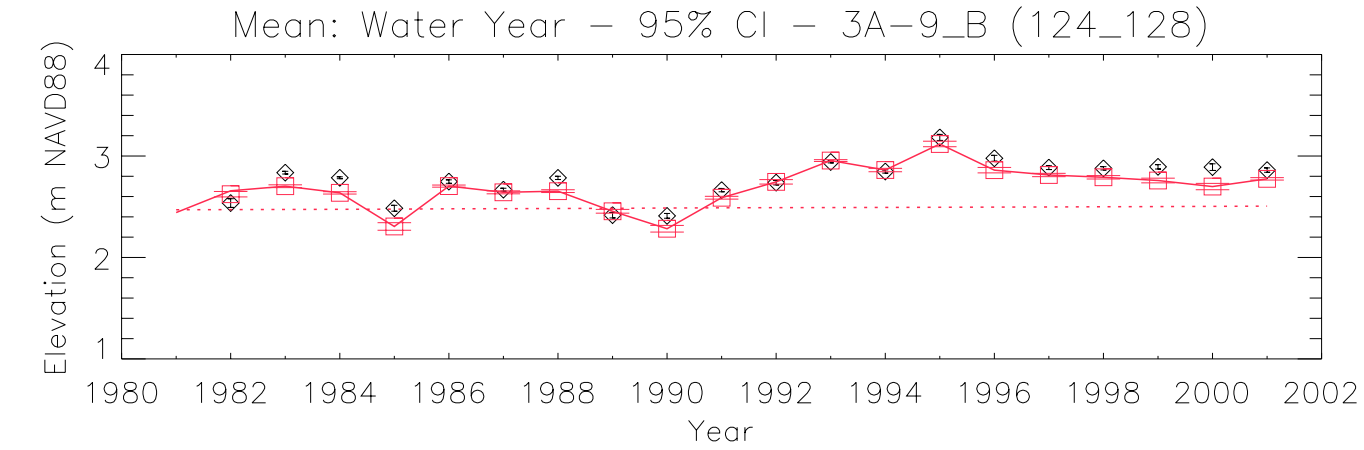

Mean: Season – 95% CI – 3A–12_B (119_118)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

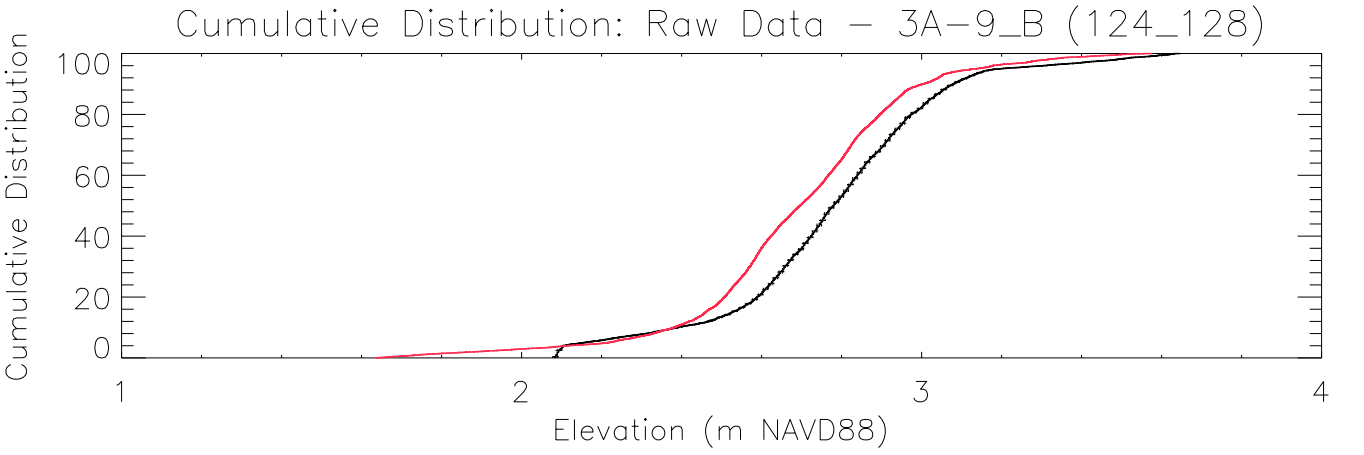

Mean: Water Year – 95% CI – 3A–12_B (119_118)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

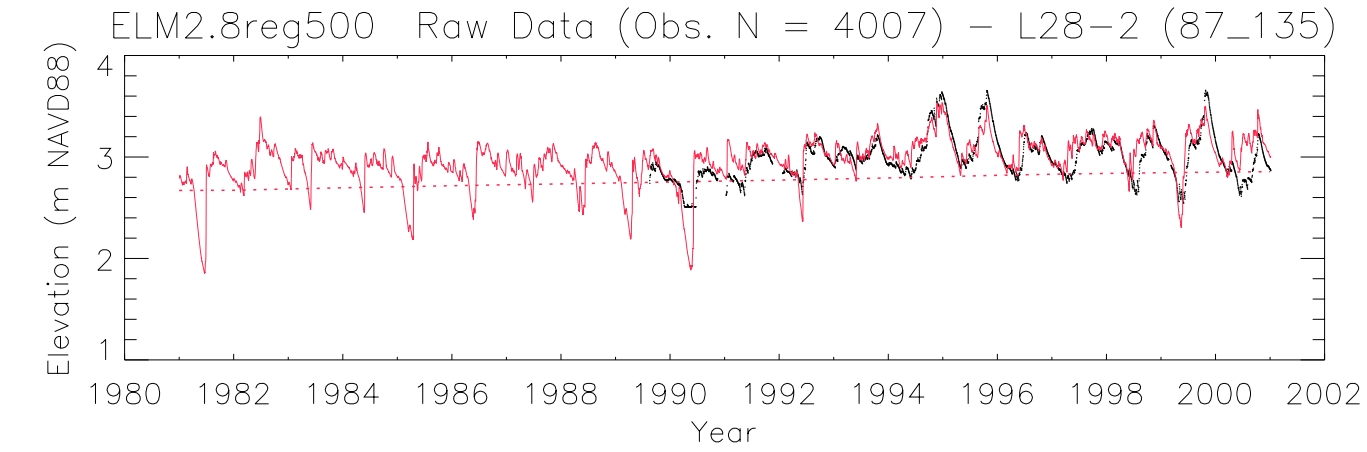

Cumulative Distribution: Raw Data – 3A–12_B (119_118)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

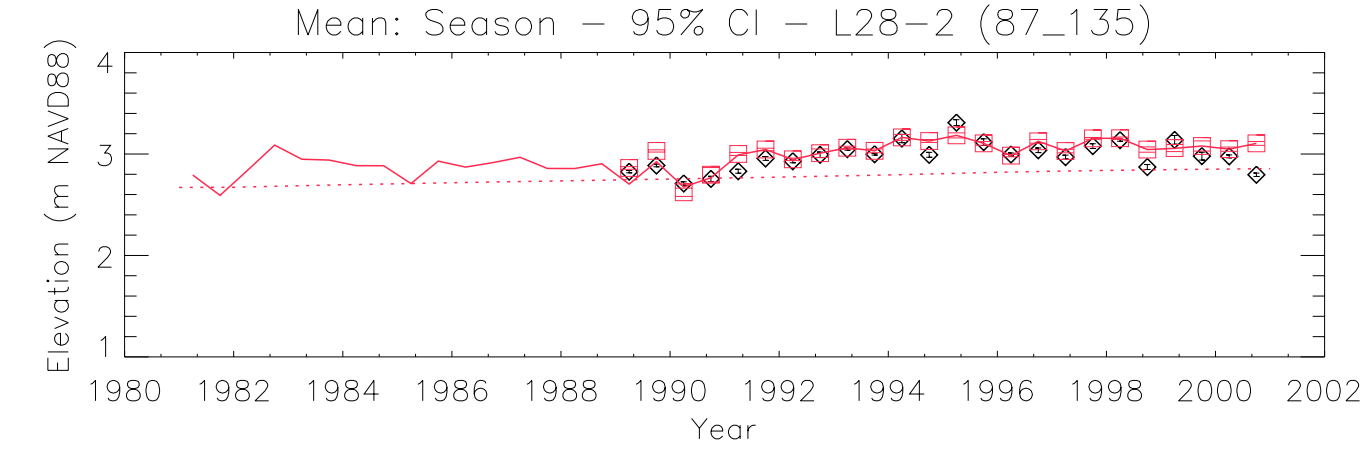

ELM2.8reg500 Raw Data (Obs. N = 6969) – 3A–9_B (124_128)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

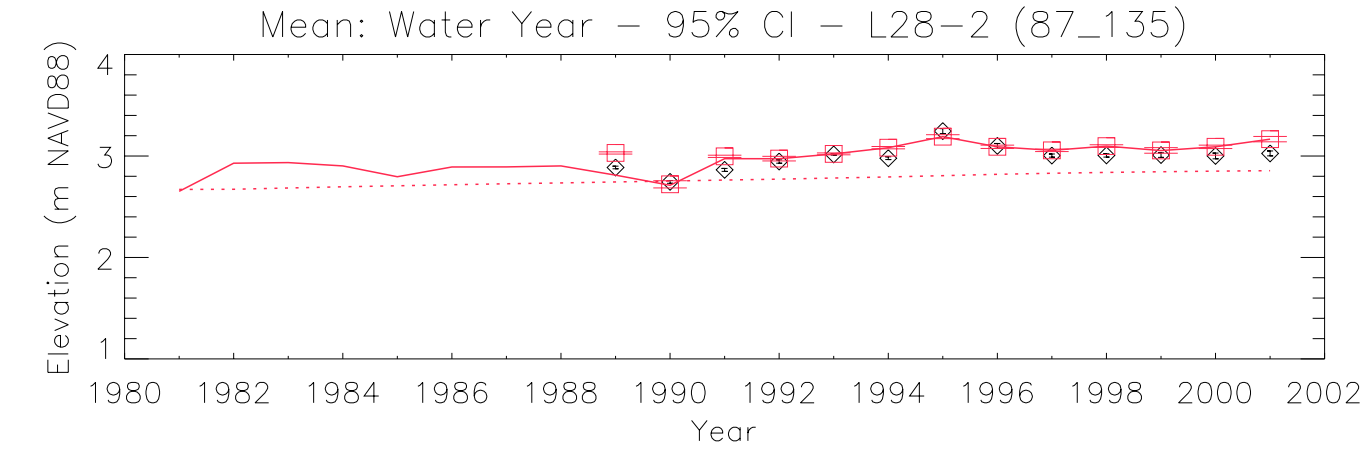

Mean: Season – 95% CI – 3A–9_B (124_128)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

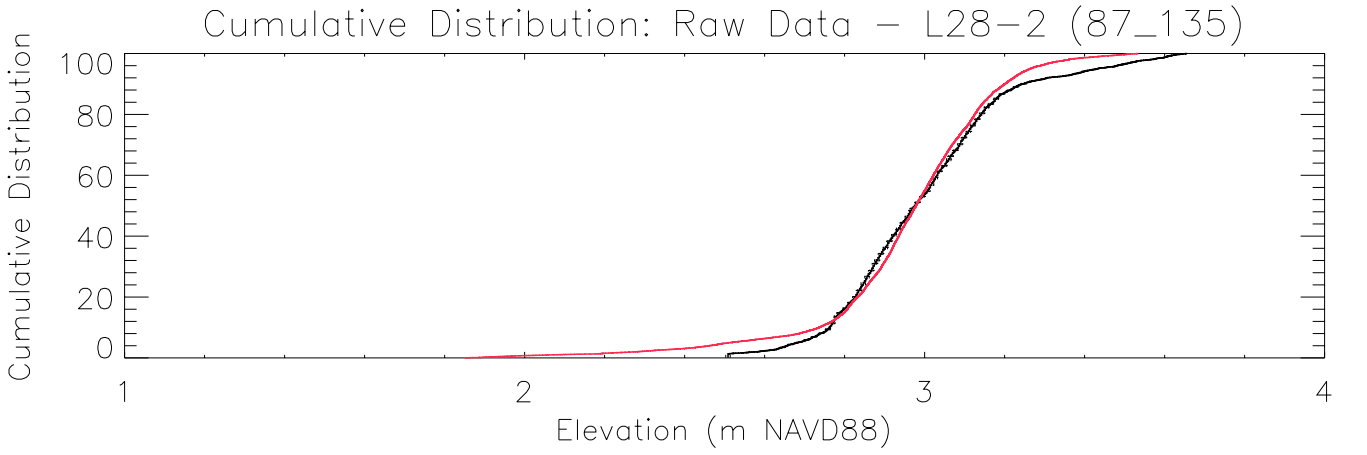

Mean: Water Year – 95% CI – 3A–9_B (124_128)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

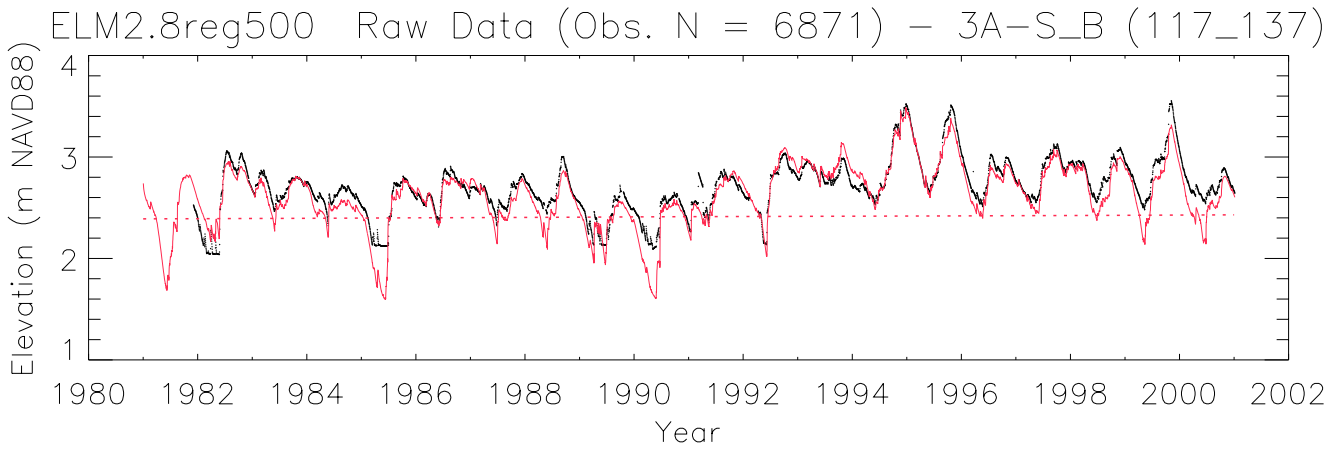

Cumulative Distribution: Raw Data – 3A–9_B (124_128)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

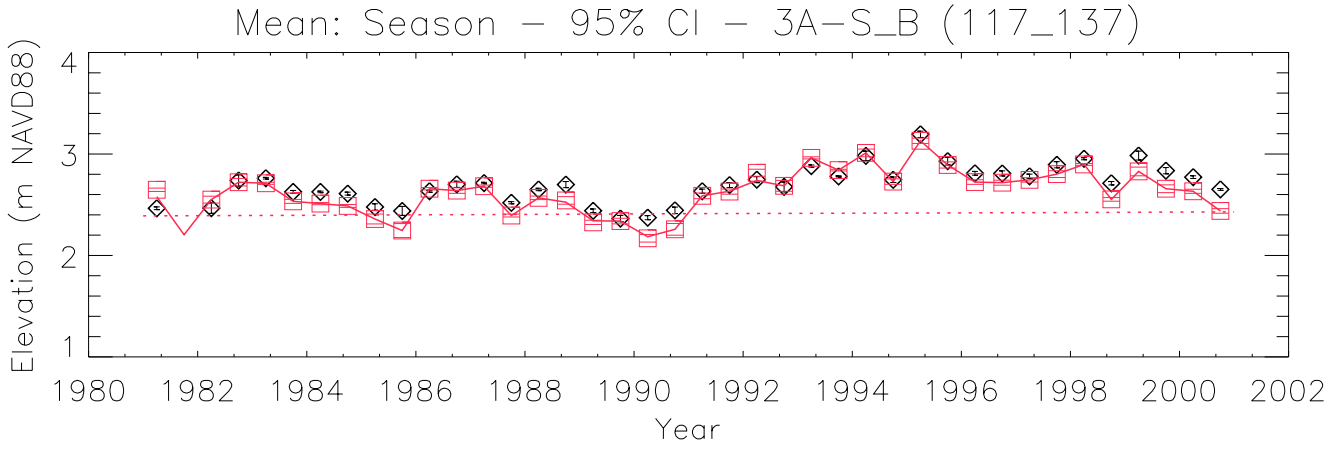

ELM2.8reg500 Raw Data (Obs. N = 4007) – L28–2 (87_135)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

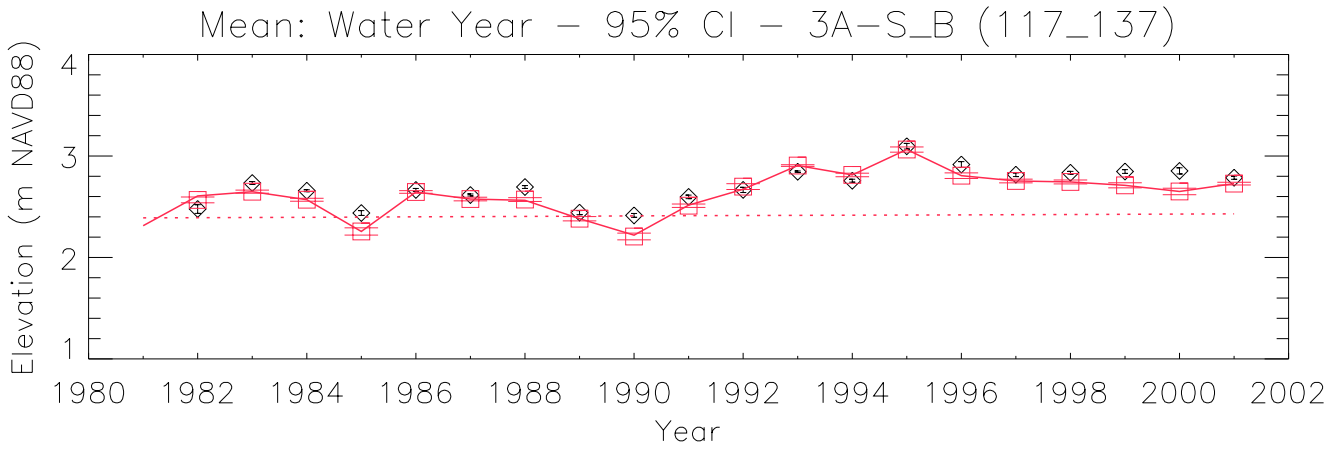

Mean: Season – 95% CI – L28–2 (87_135)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

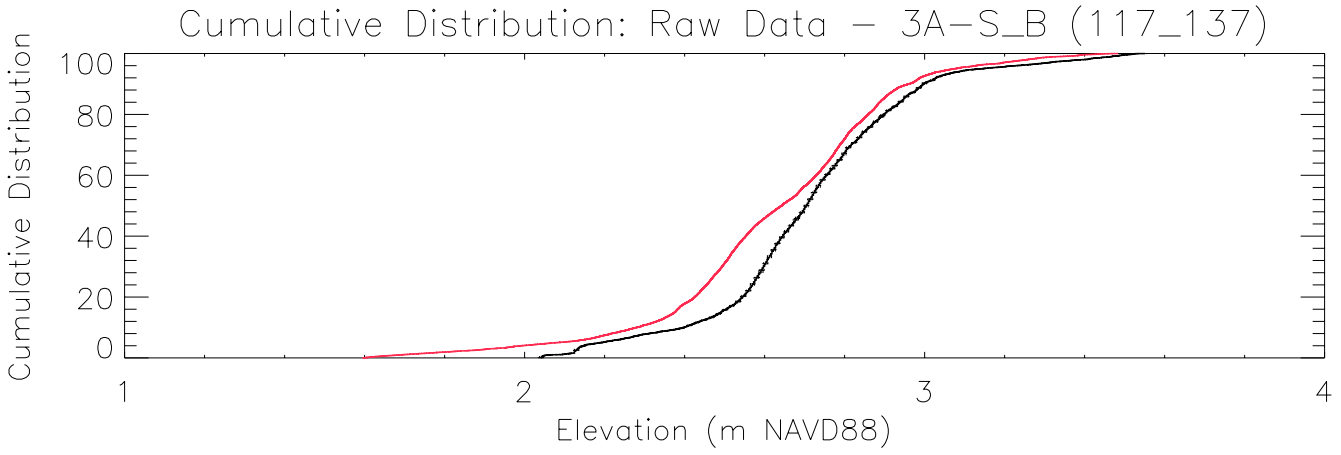

Mean: Water Year – 95% CI – L28–2 (87_135)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

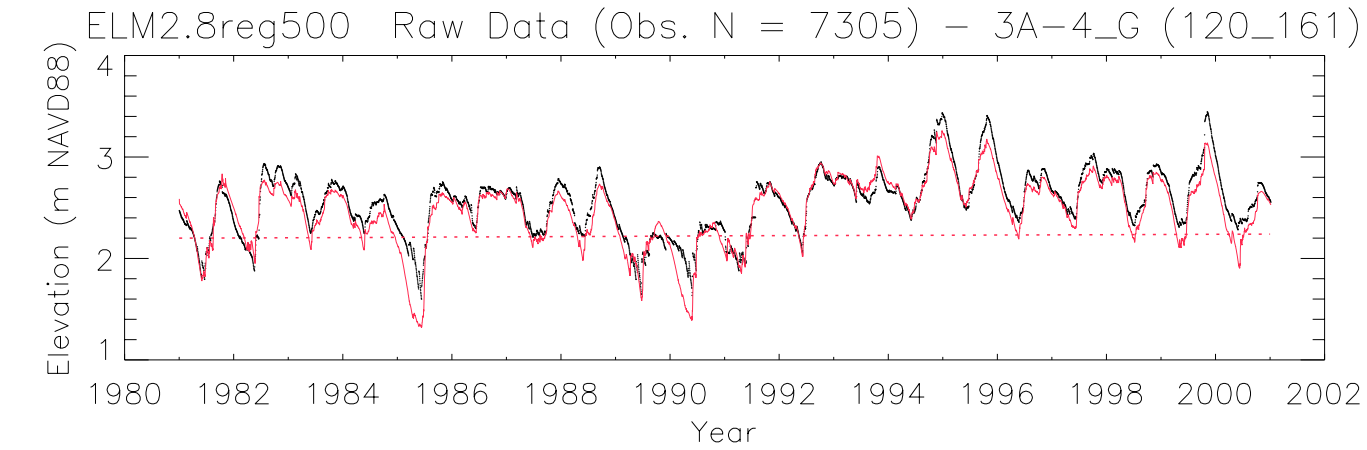

Cumulative Distribution: Raw Data – L28–2 (87_135)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

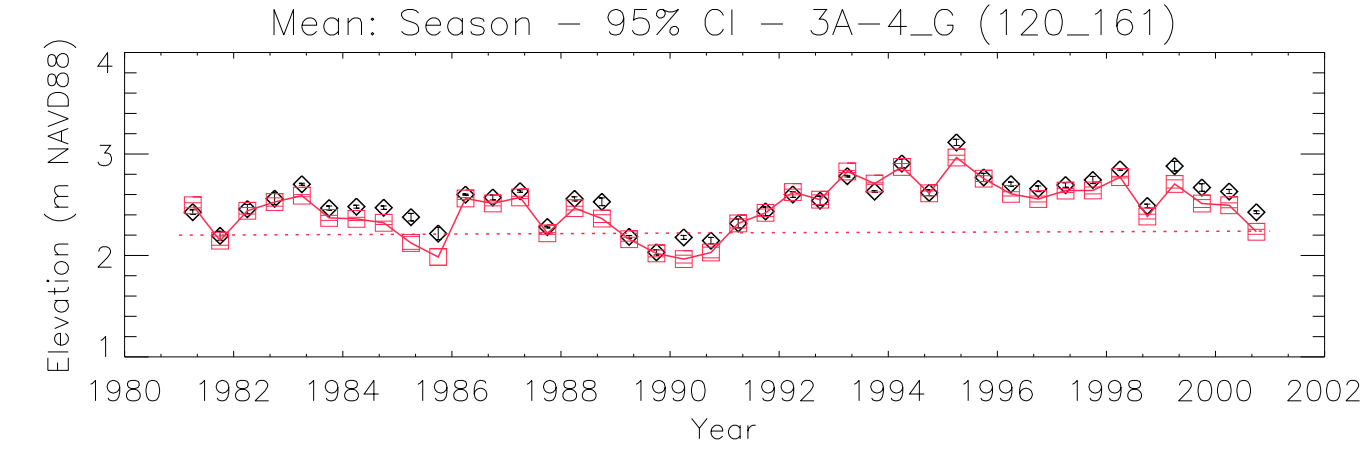

ELM2.8reg500 Raw Data (Obs. N = 6871) – 3A–S_B (117_137)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

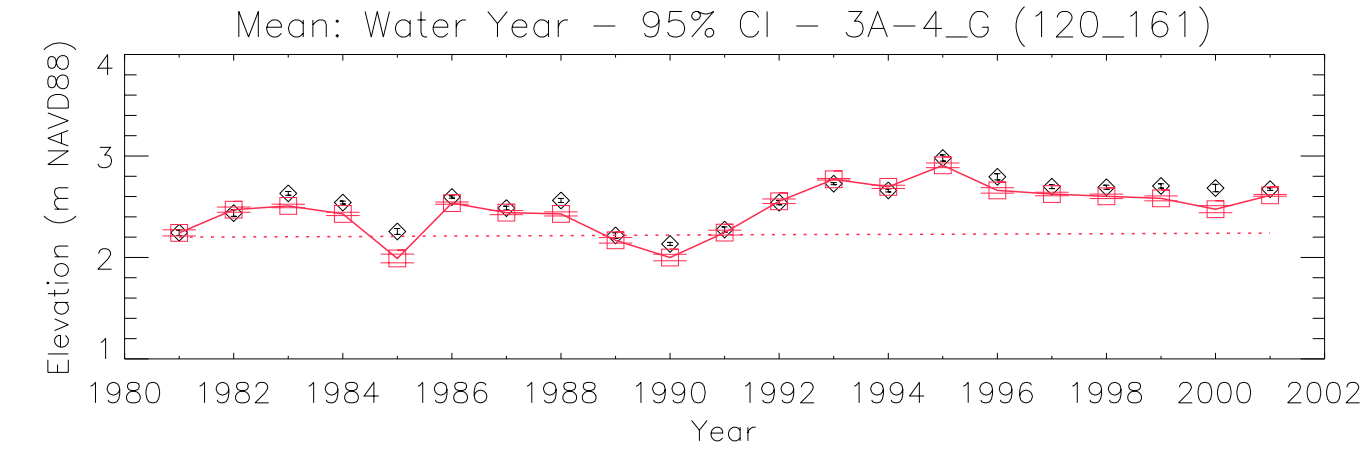

Mean: Season – 95% CI – 3A–S_B (117_137)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

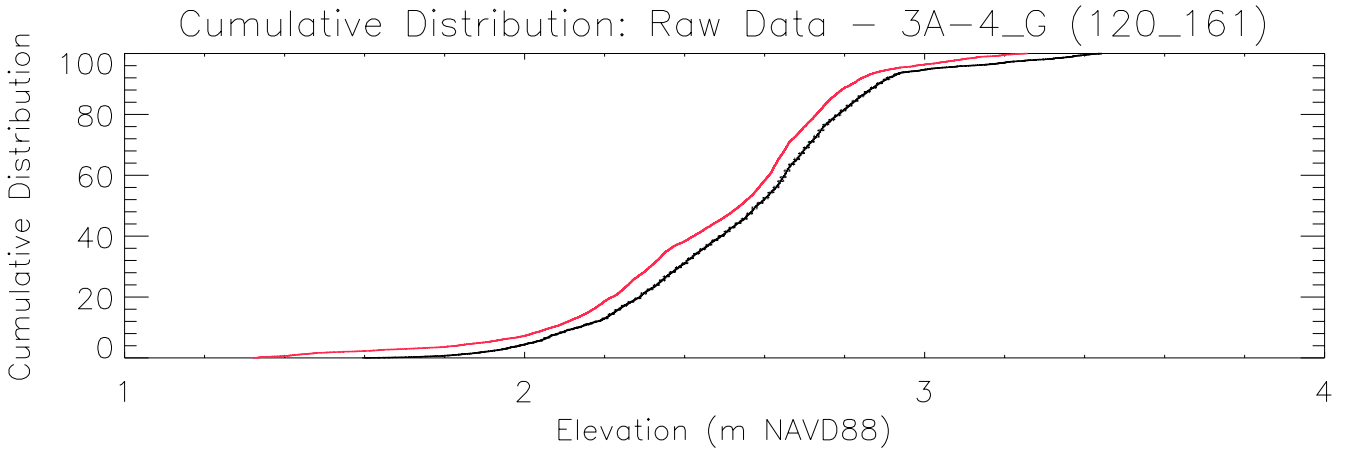

Mean: Water Year – 95% CI – 3A–S_B (117_137)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

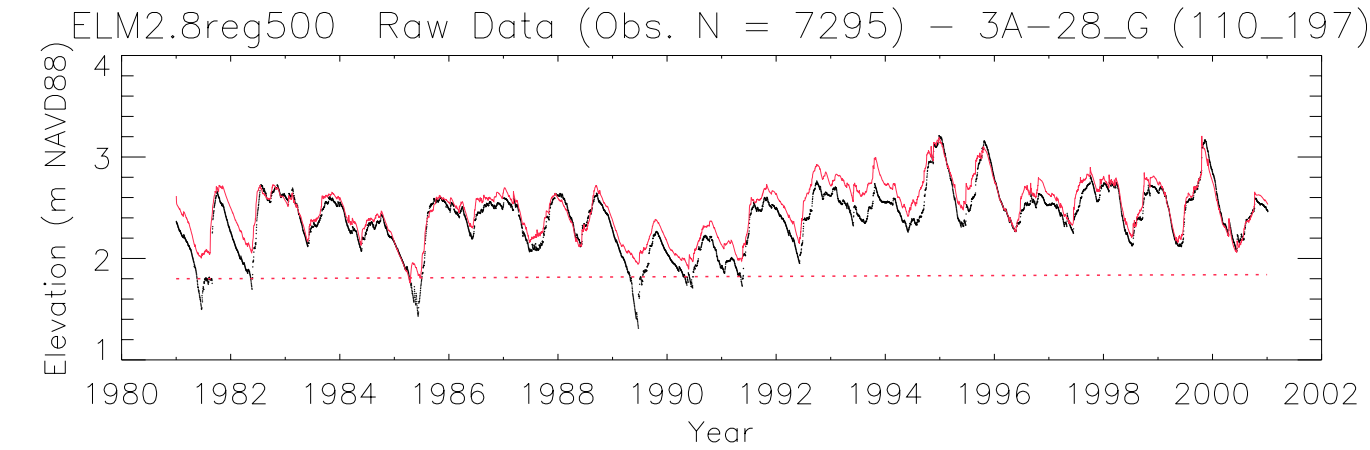

Cumulative Distribution: Raw Data – 3A–S_B (117_137)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

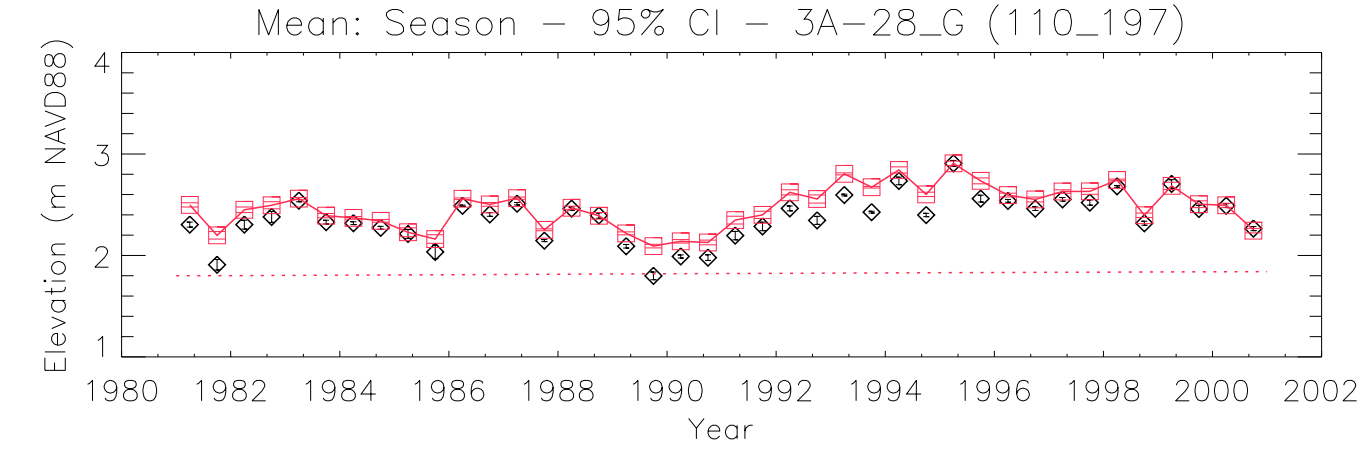

ELM2.8reg500 Raw Data (Obs. N = 7305) – 3A–4_G (120_161)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

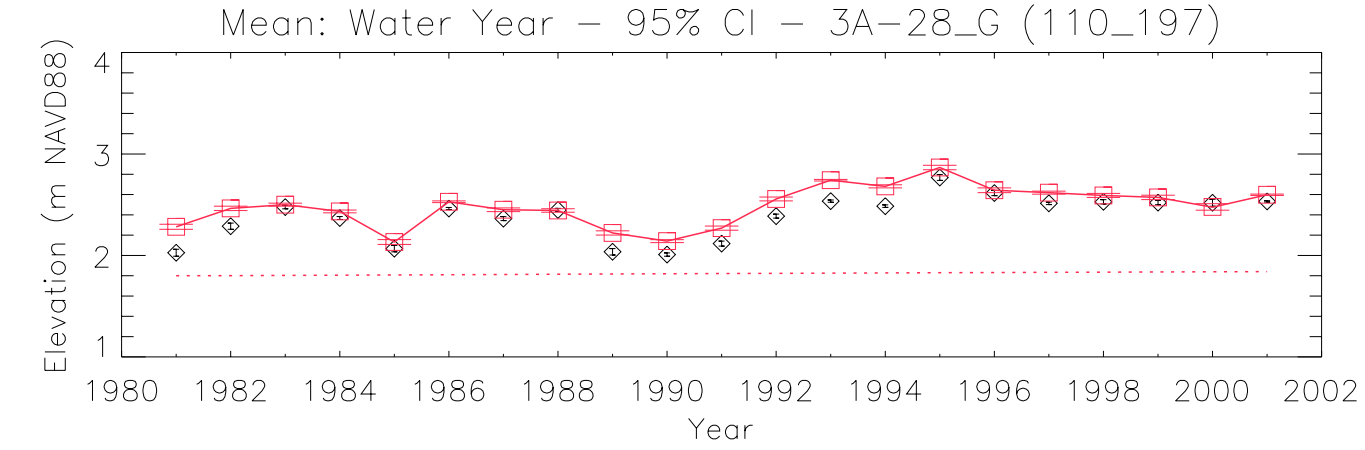

Mean: Season – 95% CI – 3A–4_G (120_161)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

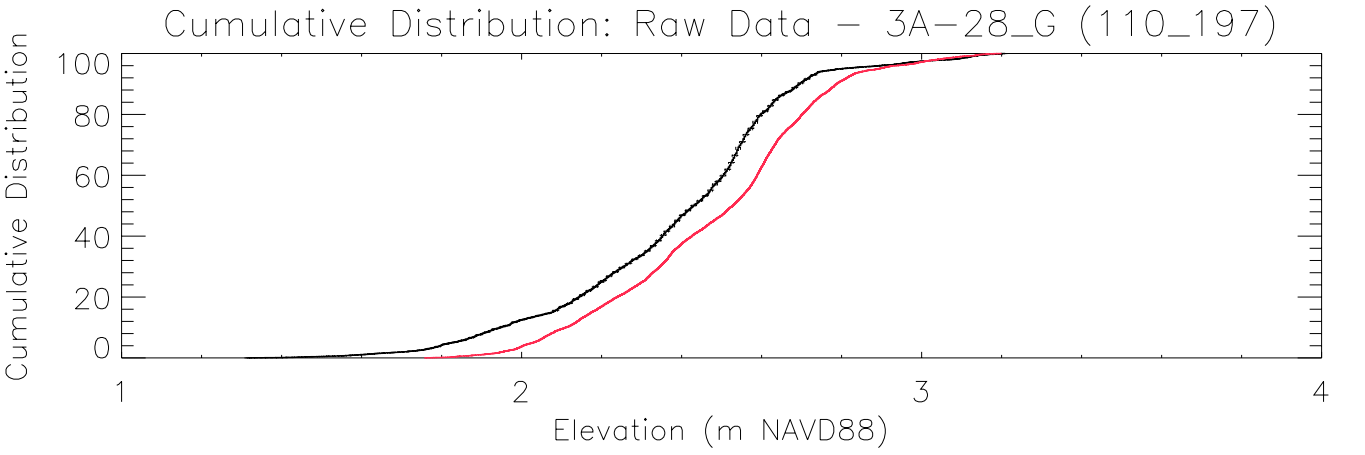

Mean: Water Year – 95% CI – 3A–4_G (120_161)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

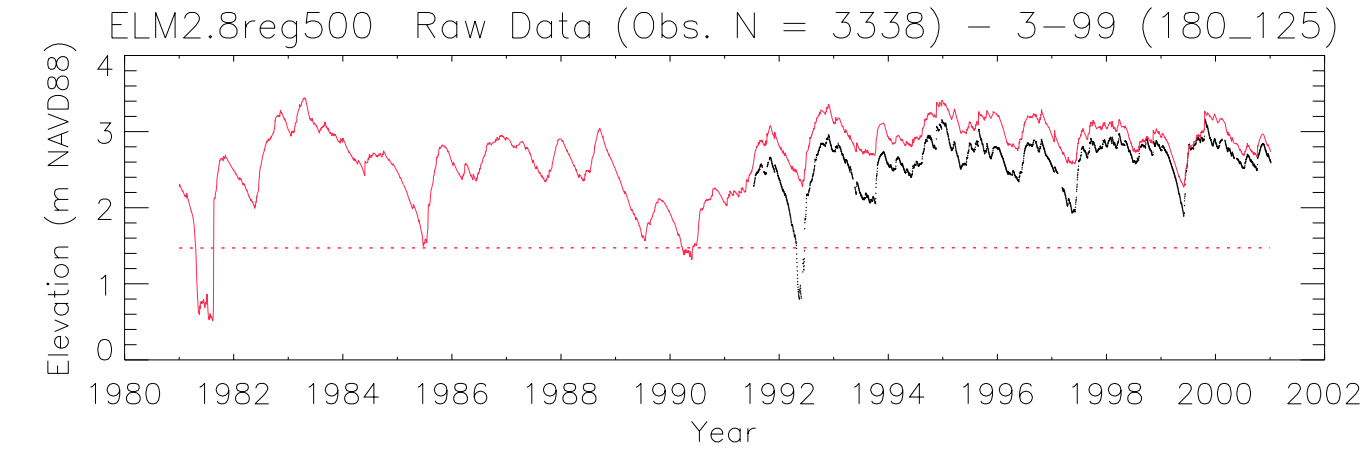

Cumulative Distribution: Raw Data – 3A–4_G (120_161)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

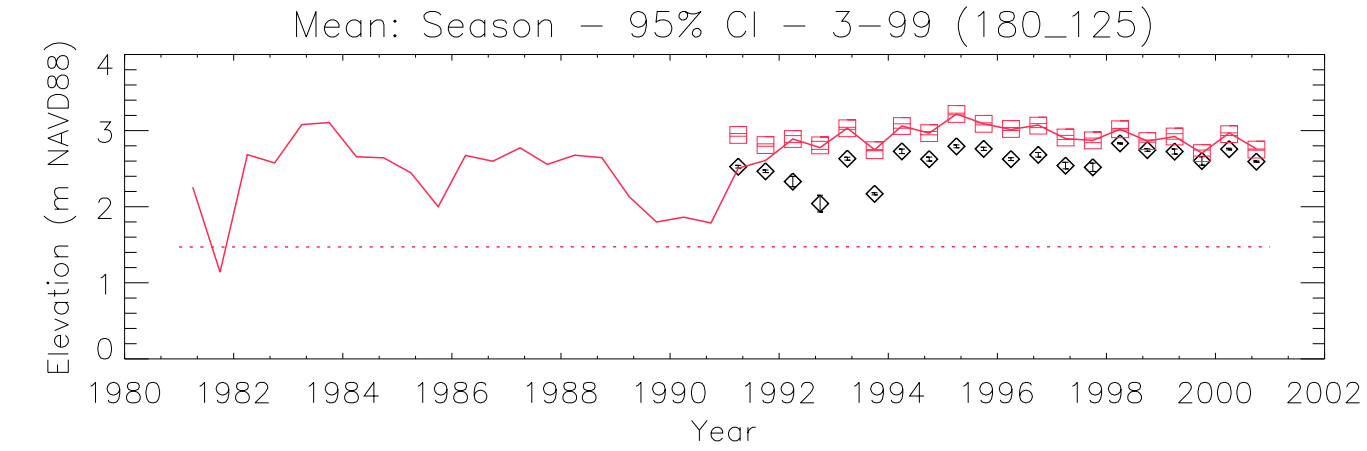

ELM2.8reg500 Raw Data (Obs. N = 7295) – 3A–28_G (110_197)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

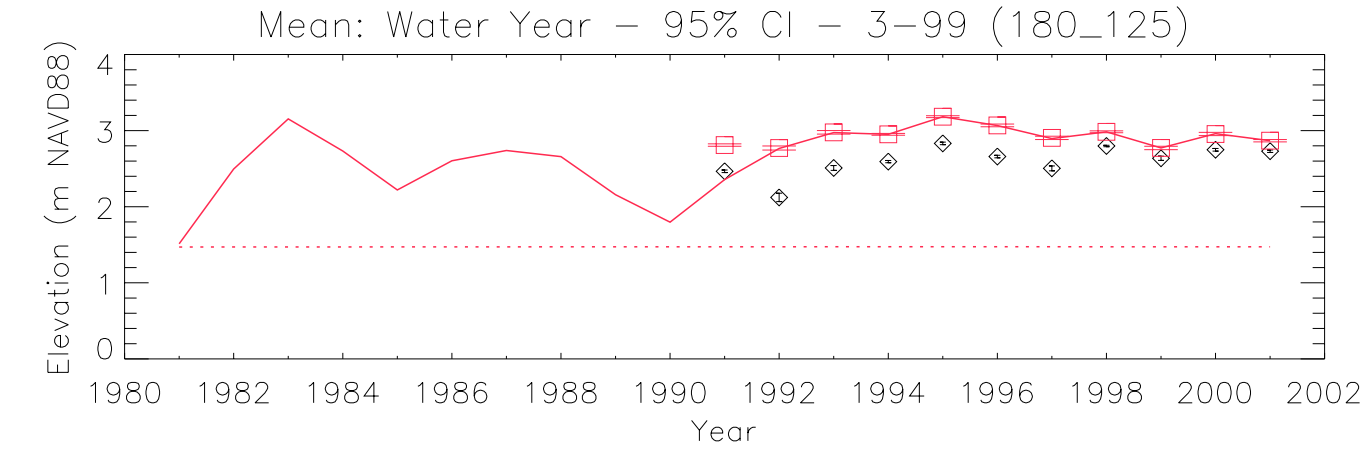

Mean: Season – 95% CI – 3A–28_G (110_197)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

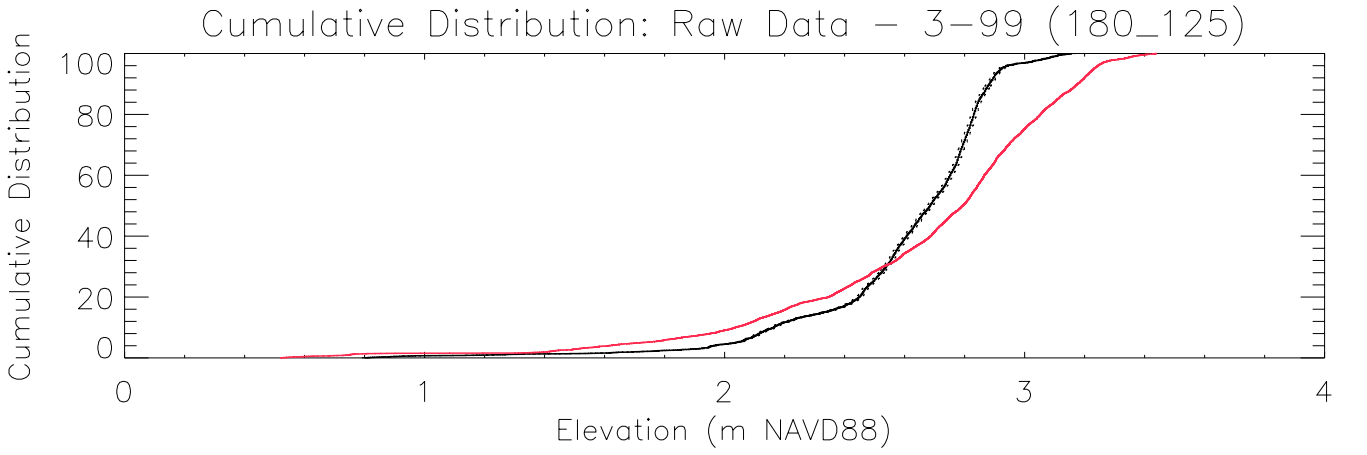

Mean: Water Year – 95% CI – 3A–28_G (110_197)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

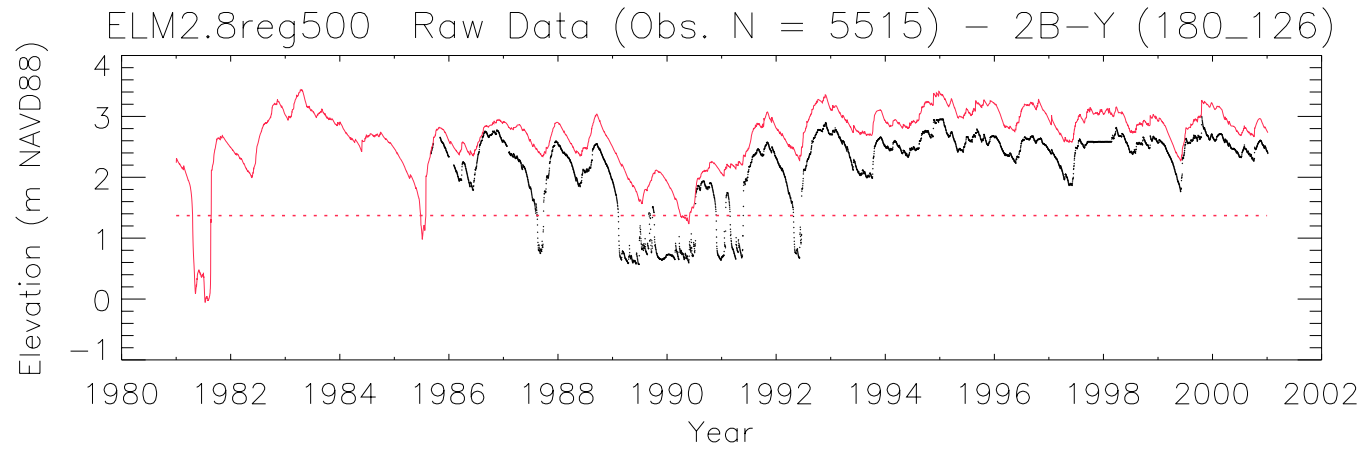

Cumulative Distribution: Raw Data – 3A–28_G (110_197)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

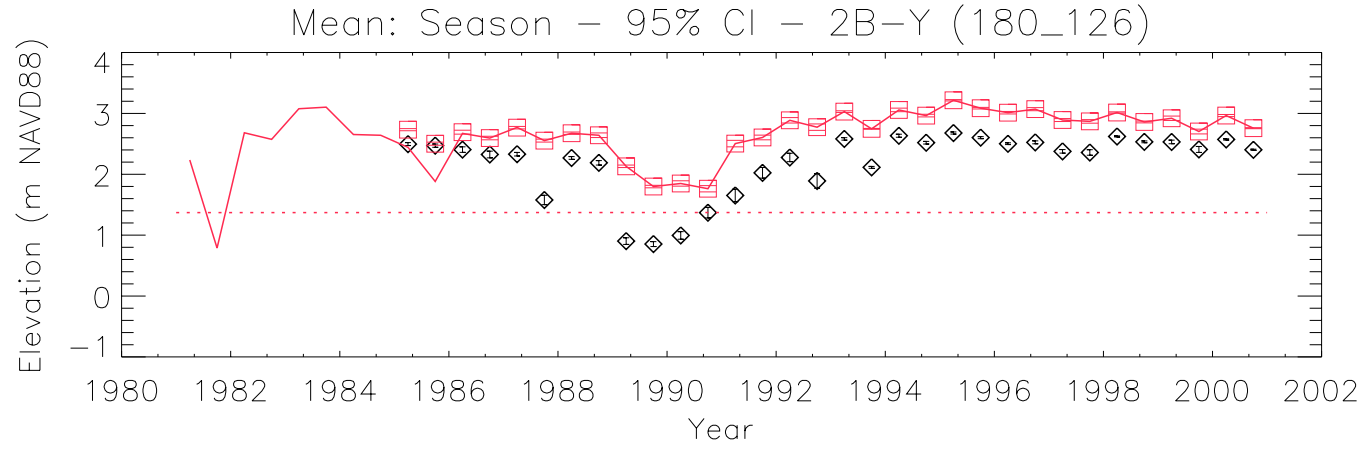

ELM2.8reg500 Raw Data (Obs. N = 3338) – 3–99 (180_125)
Elevation (m NAVD88)
0
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

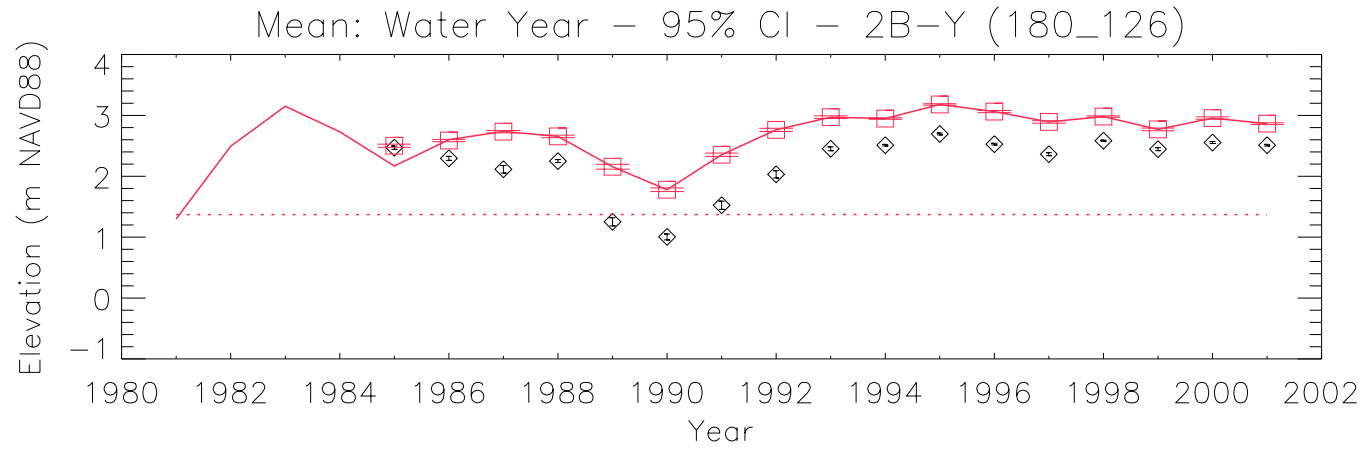

Mean: Season – 95% CI – 3–99 (180_125)
Elevation (m NAVD88)
0
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

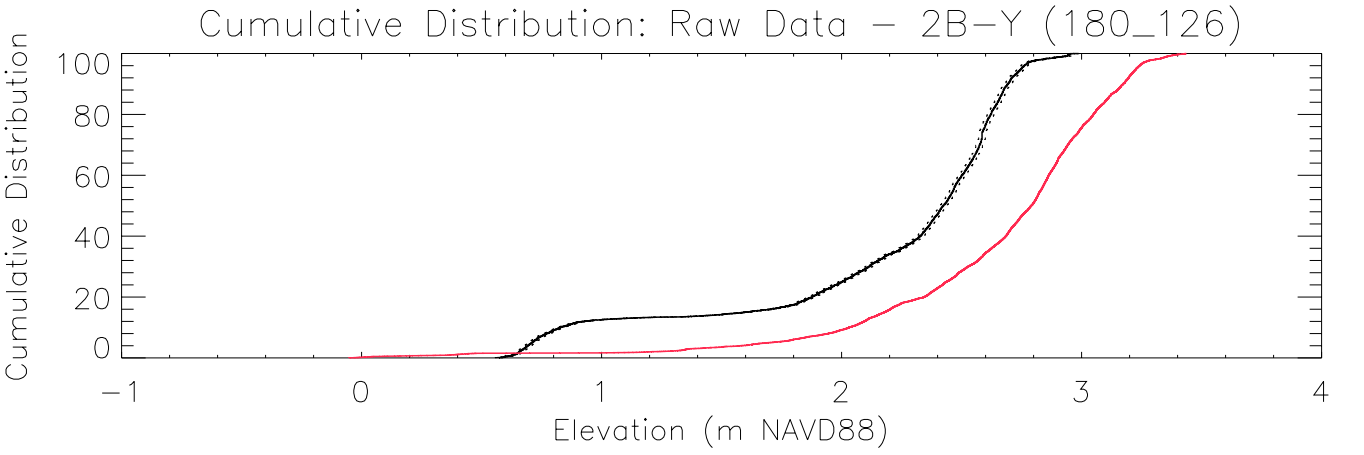

Mean: Water Year – 95% CI – 3–99 (180_125)
Elevation (m NAVD88)
0
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

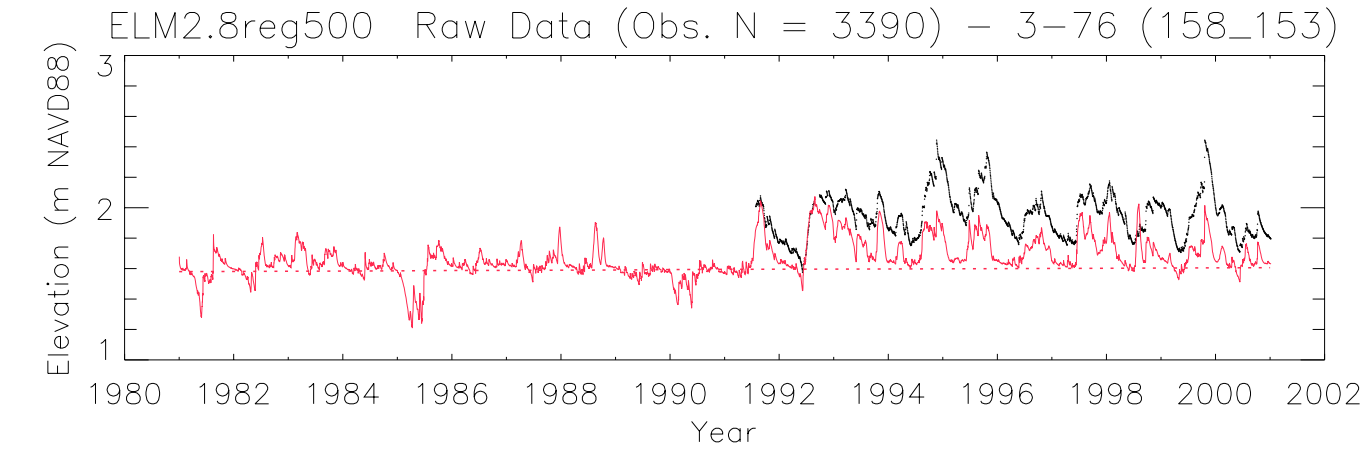

Cumulative Distribution: Raw Data – 3–99 (180_125)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
4
Elevation (m NAVD88)

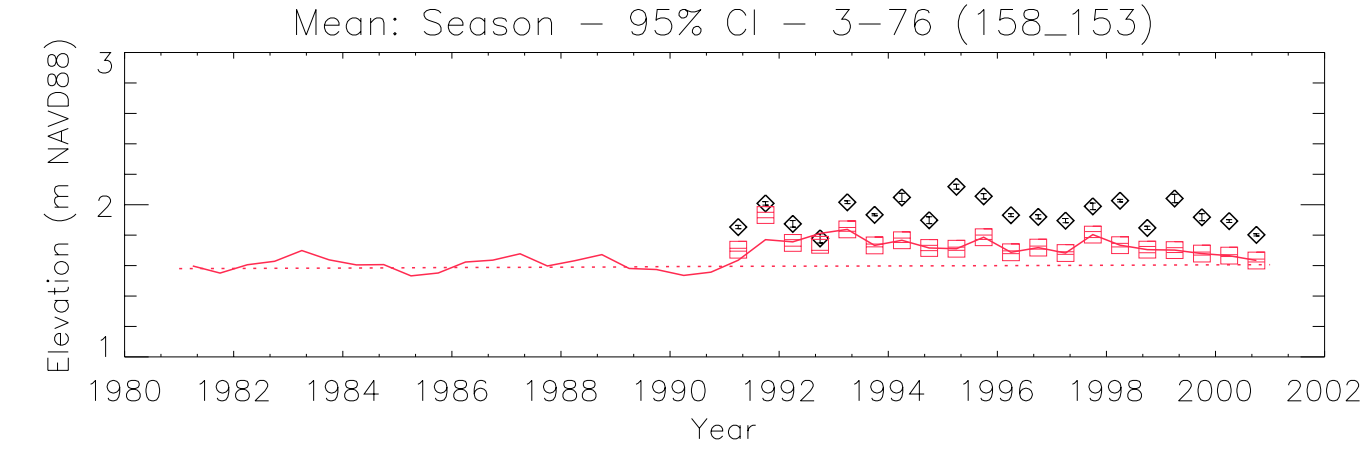

ELM2.8reg500 Raw Data (Obs. N = 5515) – 2B–Y (180_126)
Elevation (m NAVD88)
4
3
2
1
0
–1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

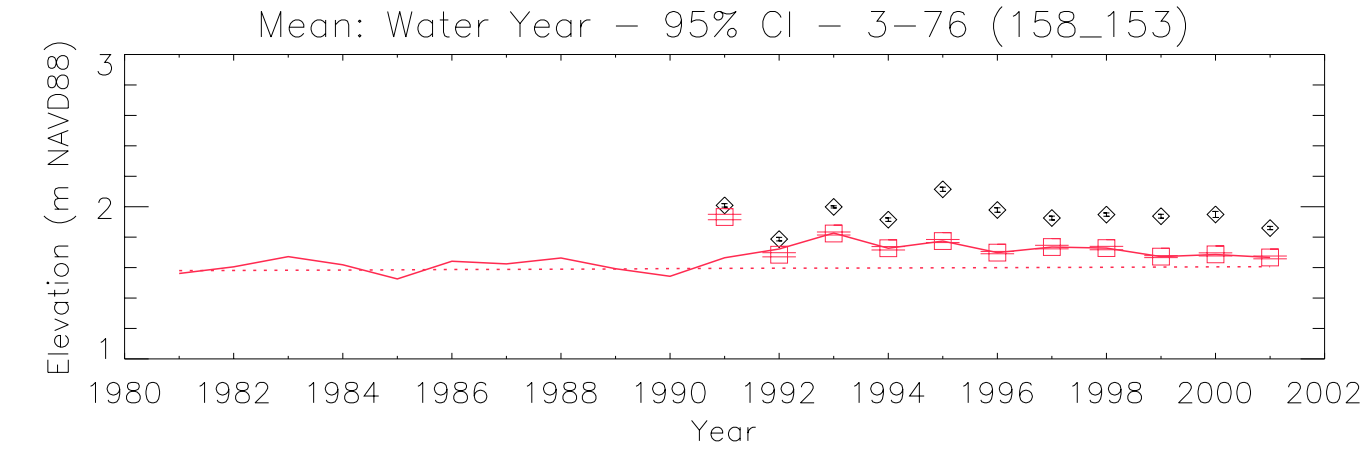

Mean: Season – 95% CI – 2B–Y (180_126)
Elevation (m NAVD88)
4
3
2
1
0
–1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

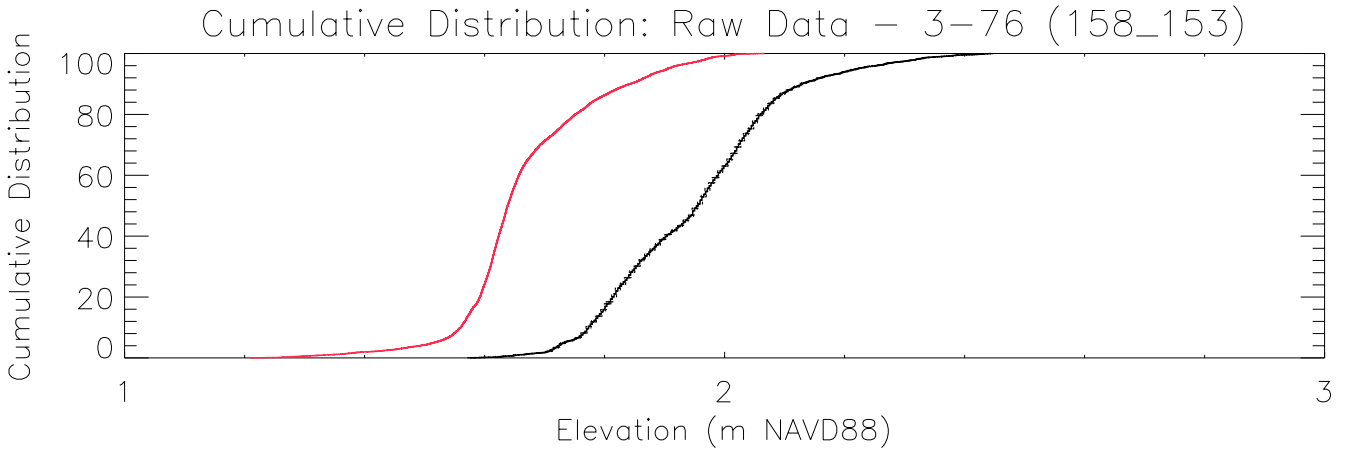

Mean: Water Year – 95% CI – 2B–Y (180_126)
Elevation (m NAVD88)
4
3
2
1
0
–1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

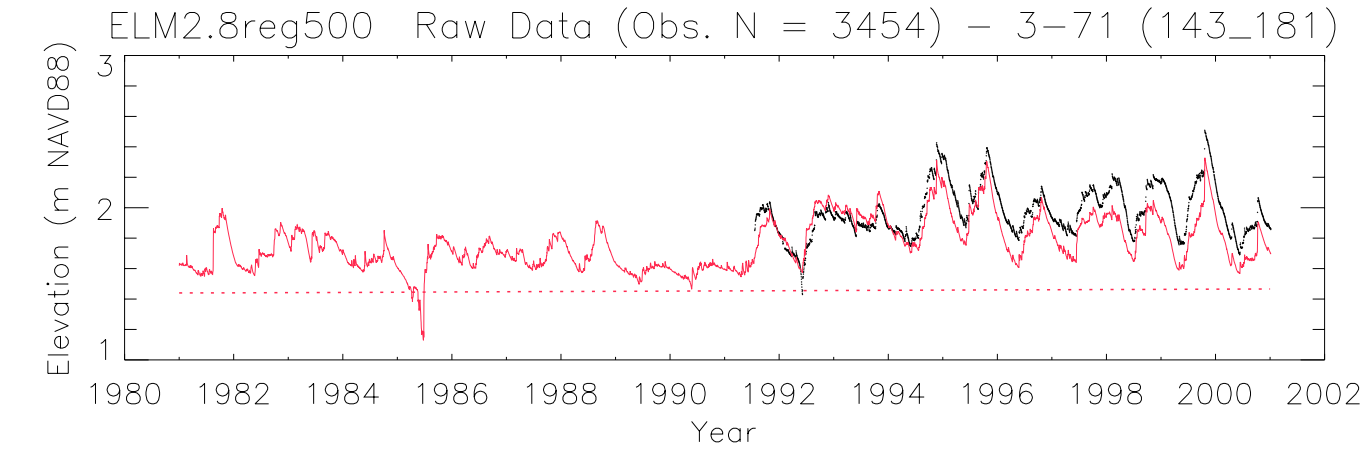

Cumulative Distribution: Raw Data – 2B–Y (180_126)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
2
3
4
Elevation (m NAVD88)

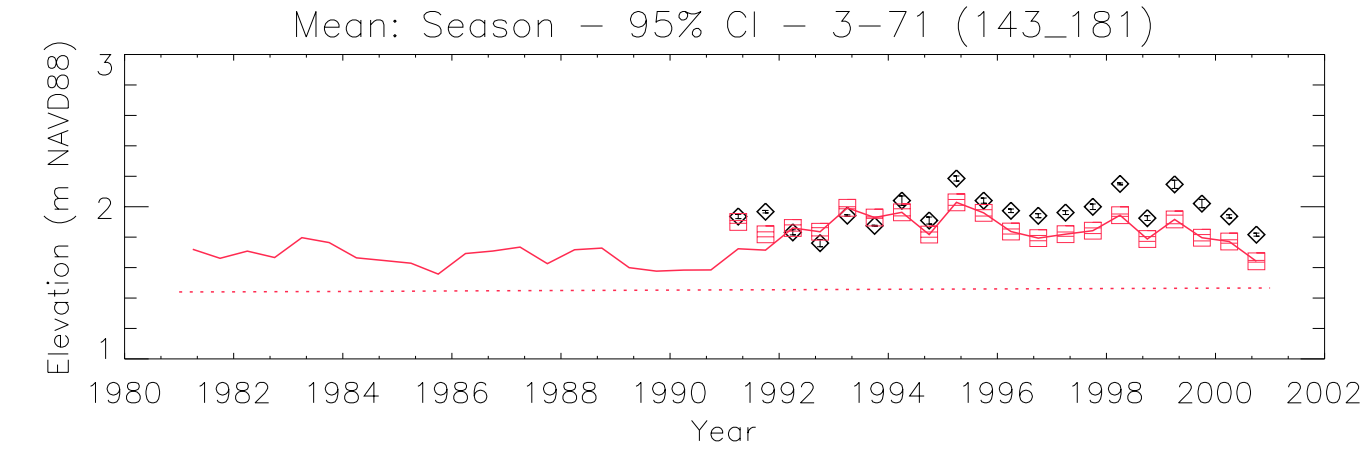

ELM2.8reg500 Raw Data (Obs. N = 3390) – 3–76 (158_153)
Elevation (m NAVD88)
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

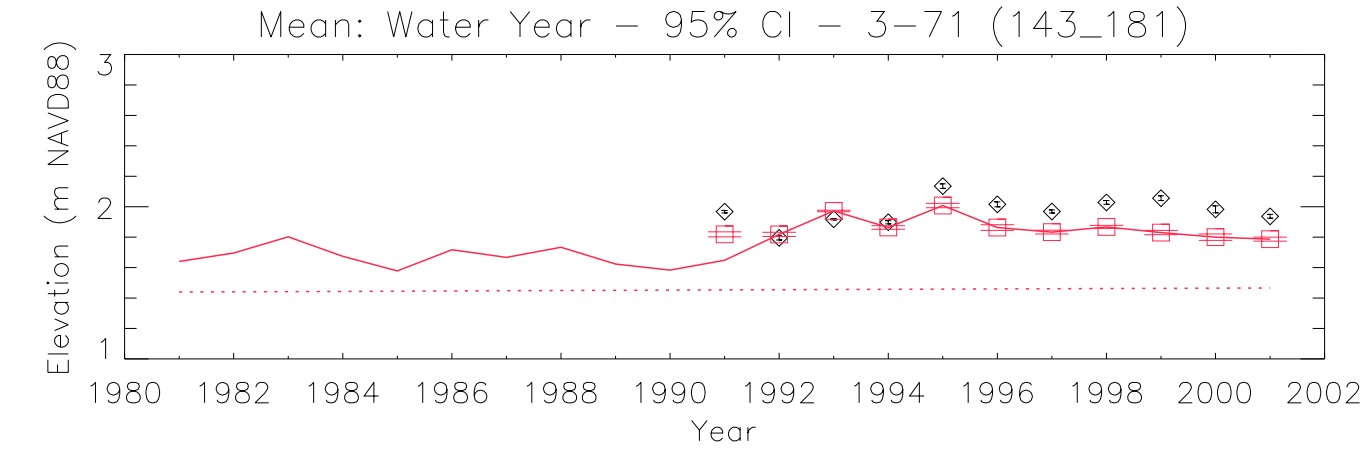

Mean: Season – 95% CI – 3–76 (158_153)
Elevation (m NAVD88)
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

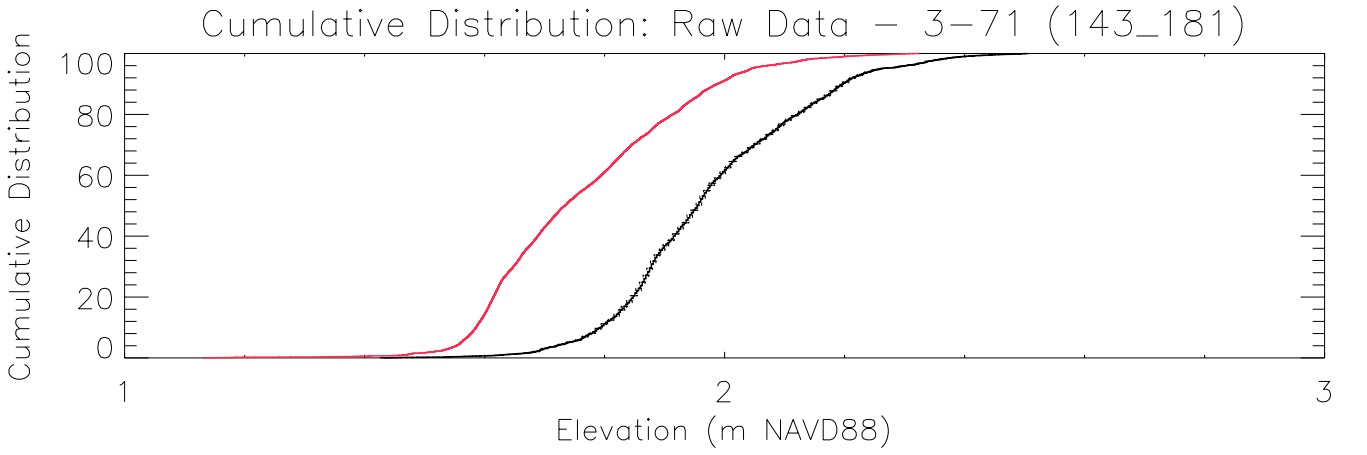

Mean: Water Year – 95% CI – 3–76 (158_153)
Elevation (m NAVD88)
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

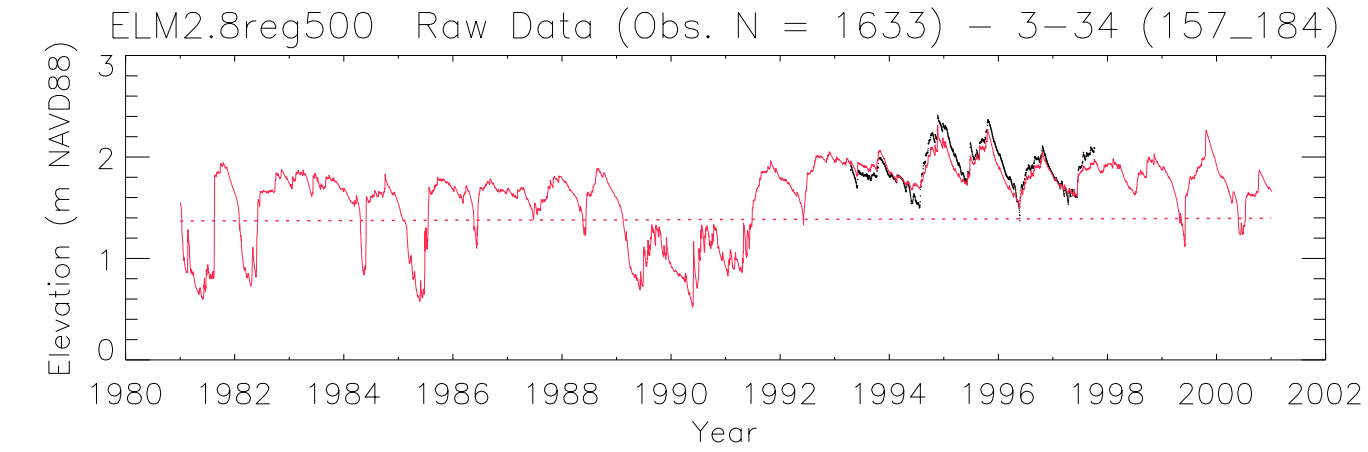

Cumulative Distribution: Raw Data – 3–76 (158_153)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
Elevation (m NAVD88)

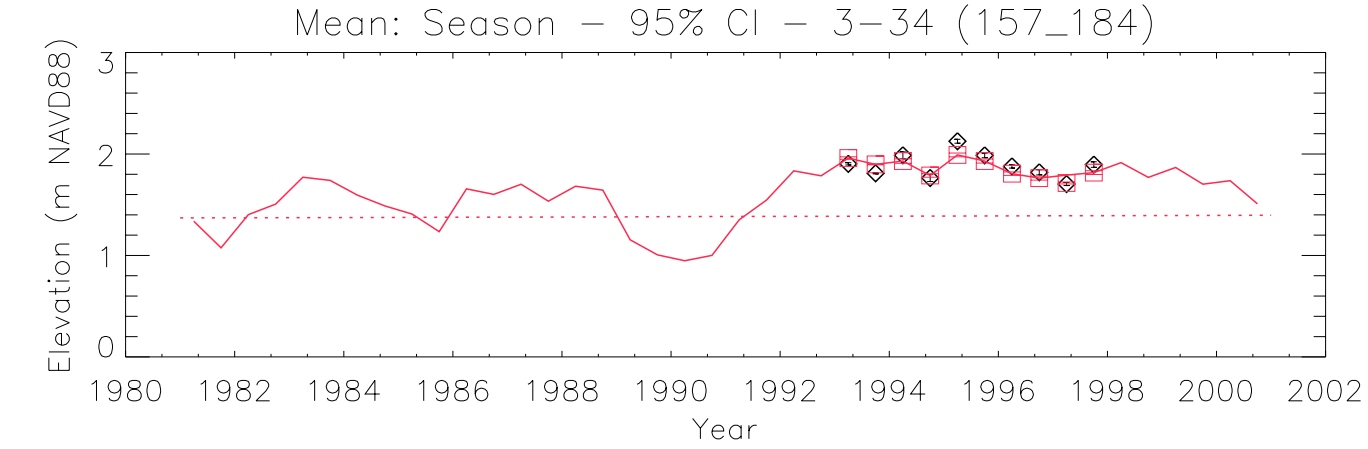

ELM2.8reg500 Raw Data (Obs. N = 3454) – 3–71 (143_181)
Elevation (m NAVD88)
3
2
1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

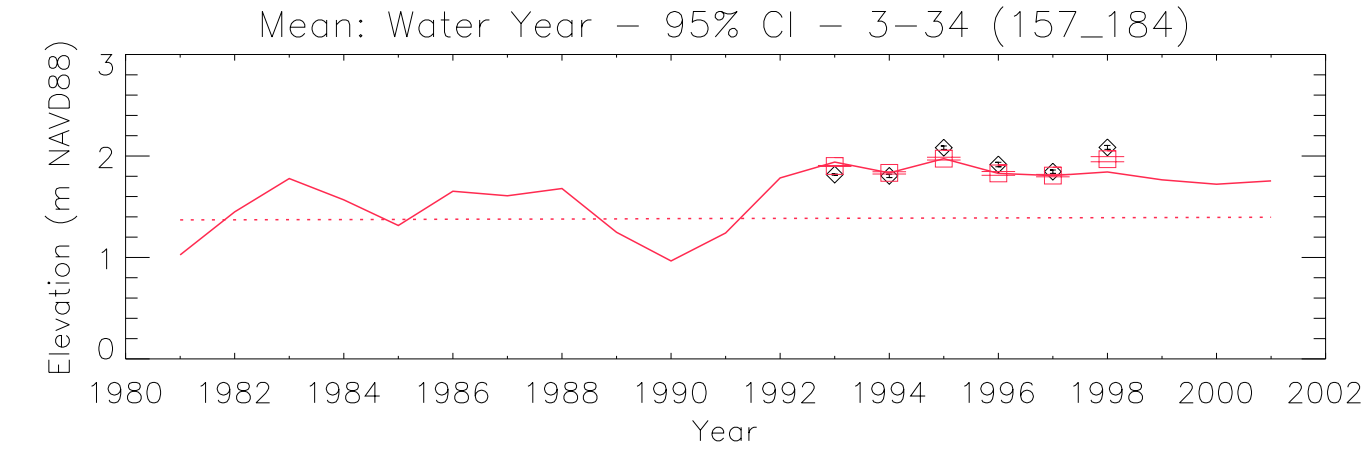

Mean: Season – 95% CI – 3–71 (143_181)
Elevation (m NAVD88)
3
2
1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

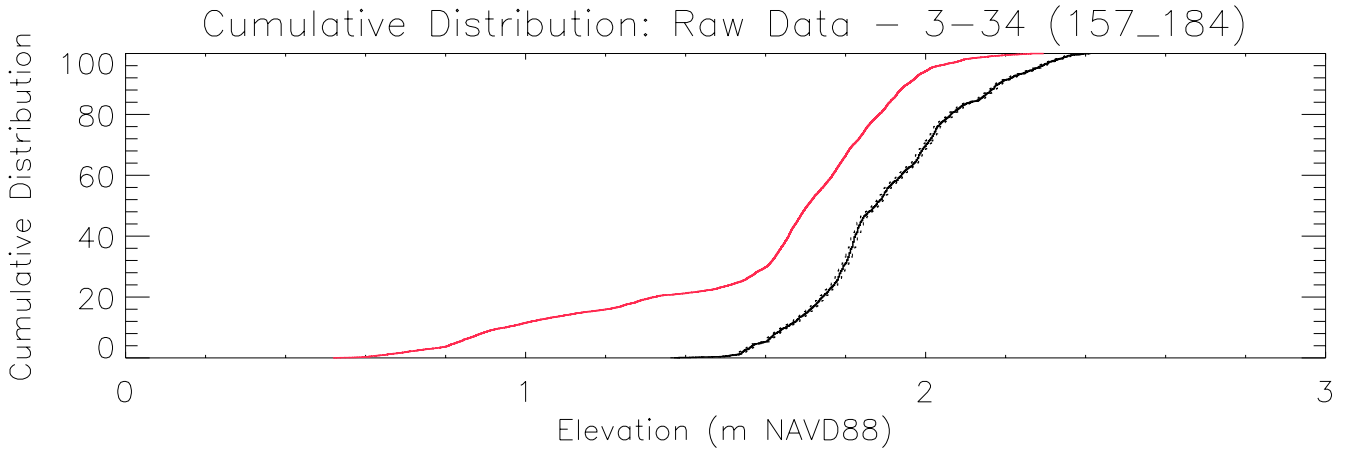

Mean: Water Year – 95% CI – 3–71 (143_181)
Elevation (m NAVD88)
3
2
1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

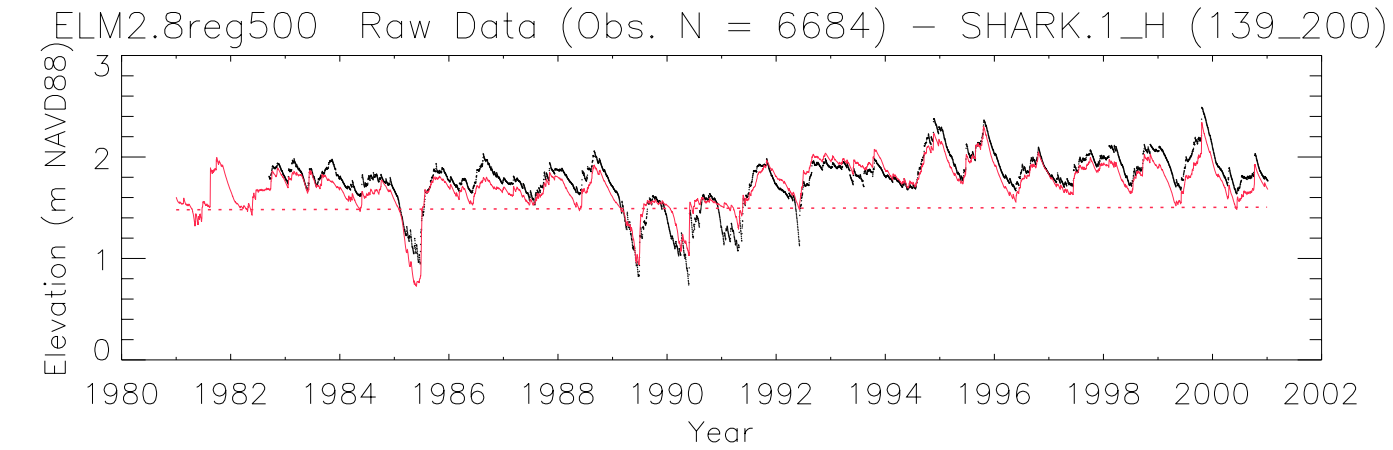

Cumulative Distribution: Raw Data – 3–71 (143_181)
Cumulative Distribution
100
80
60
40
20
0
1
2
3
Elevation (m NAVD88)

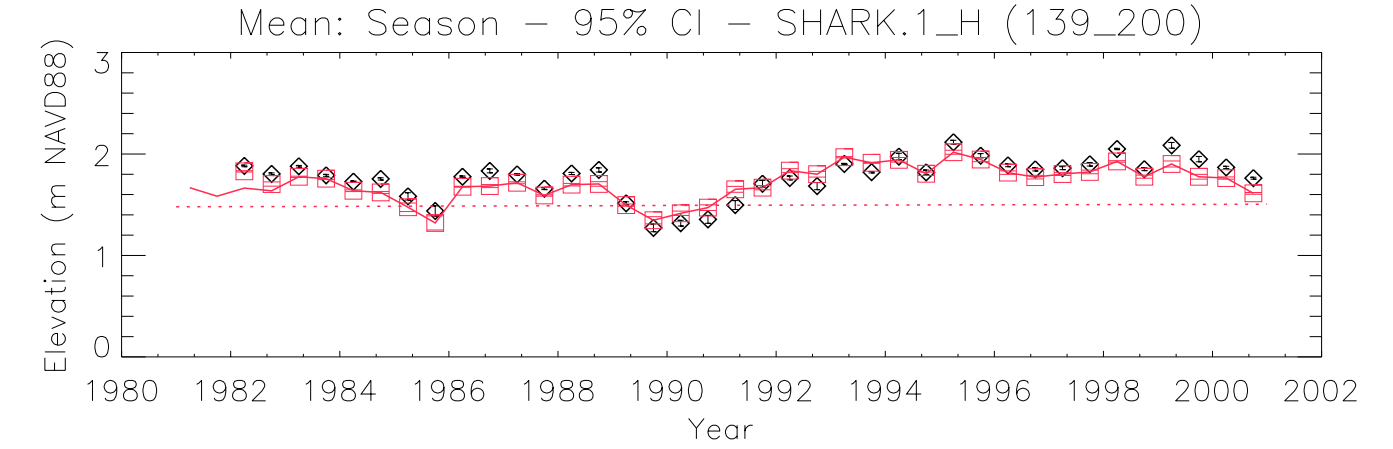

ELM2.8reg500 Raw Data (Obs. N = 1633) – 3–34 (157_184)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

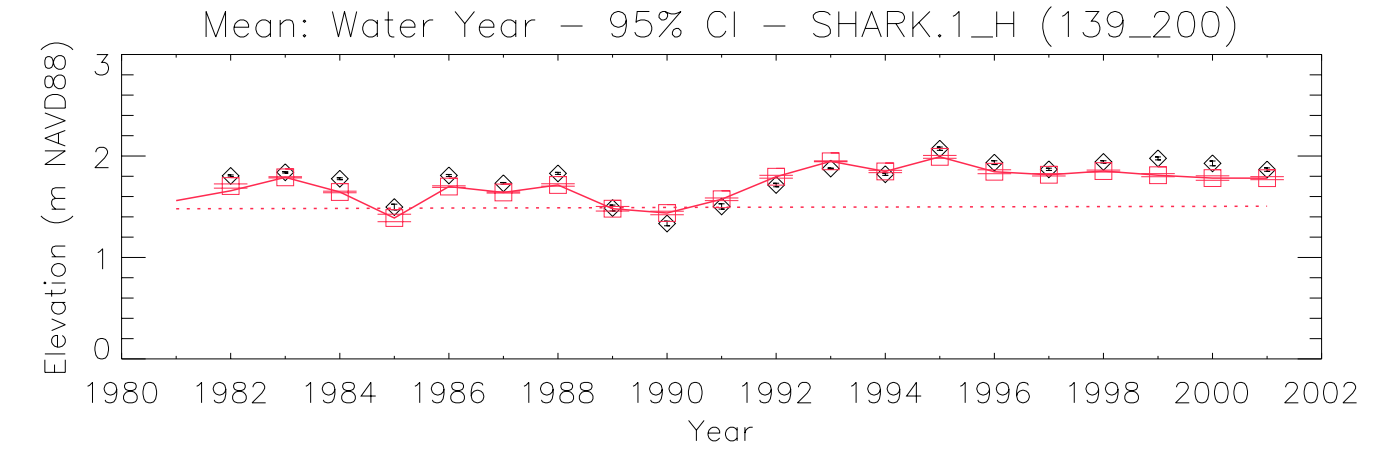

Mean: Season – 95% CI – 3–34 (157_184)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

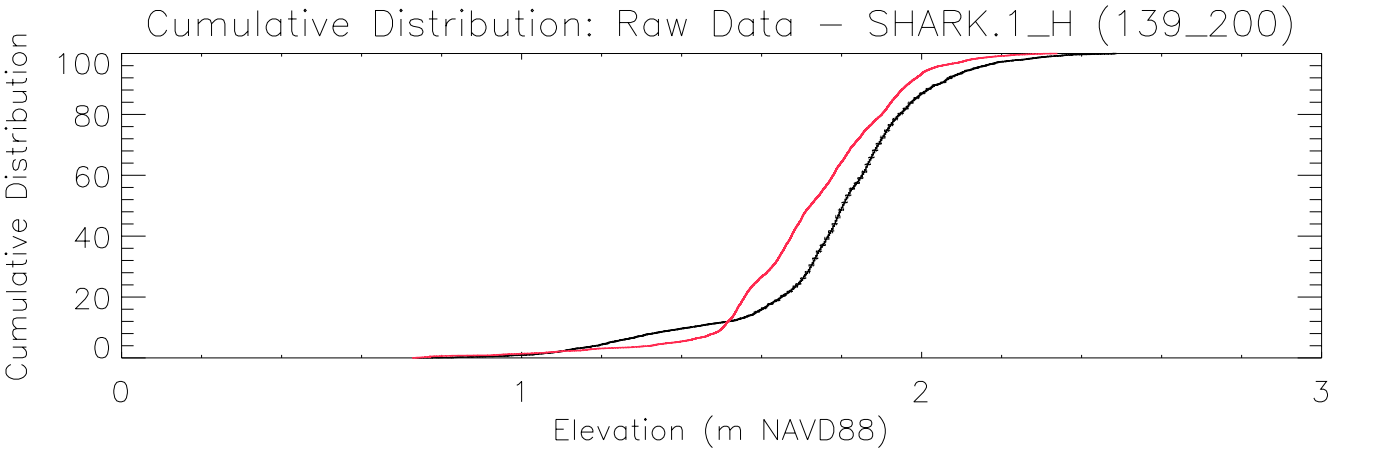

Mean: Water Year – 95% CI – 3–34 (157_184)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

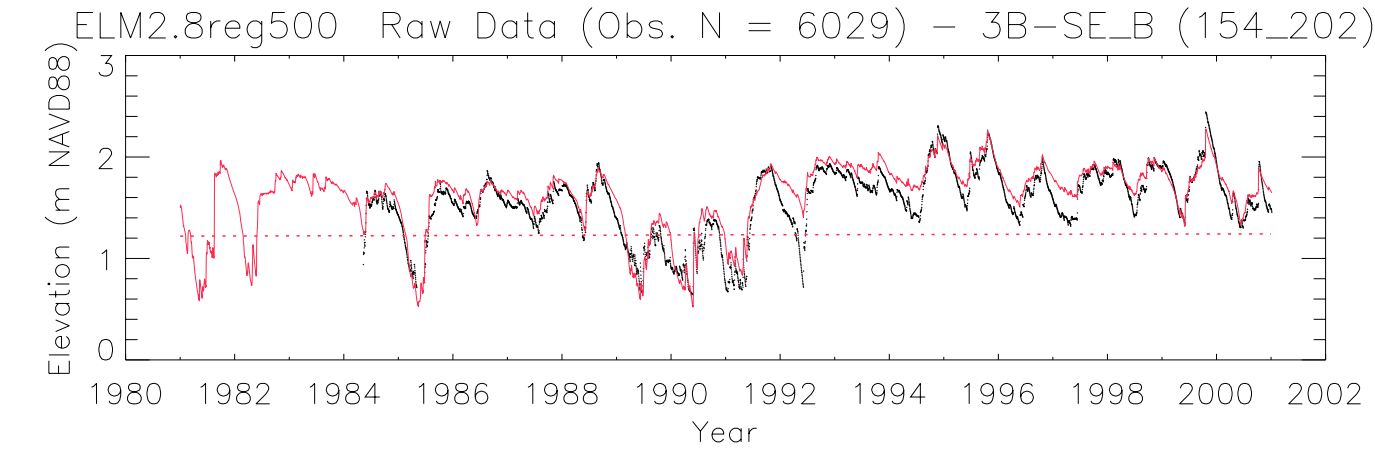

Cumulative Distribution: Raw Data – 3–34 (157_184)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

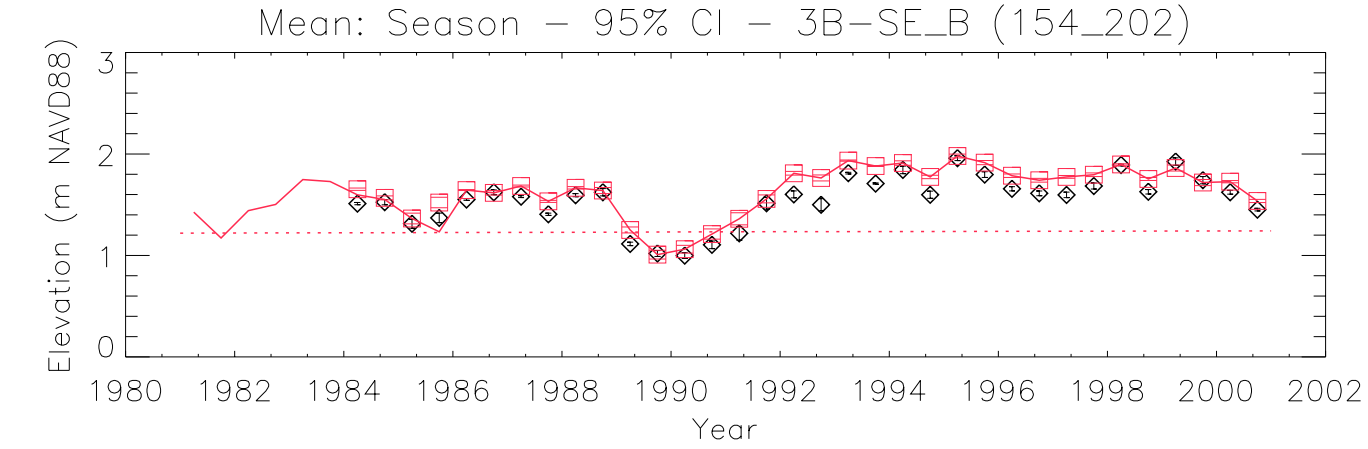

ELM2.8reg500 Raw Data (Obs. N = 6684) – SHARK.1_H (139_200)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

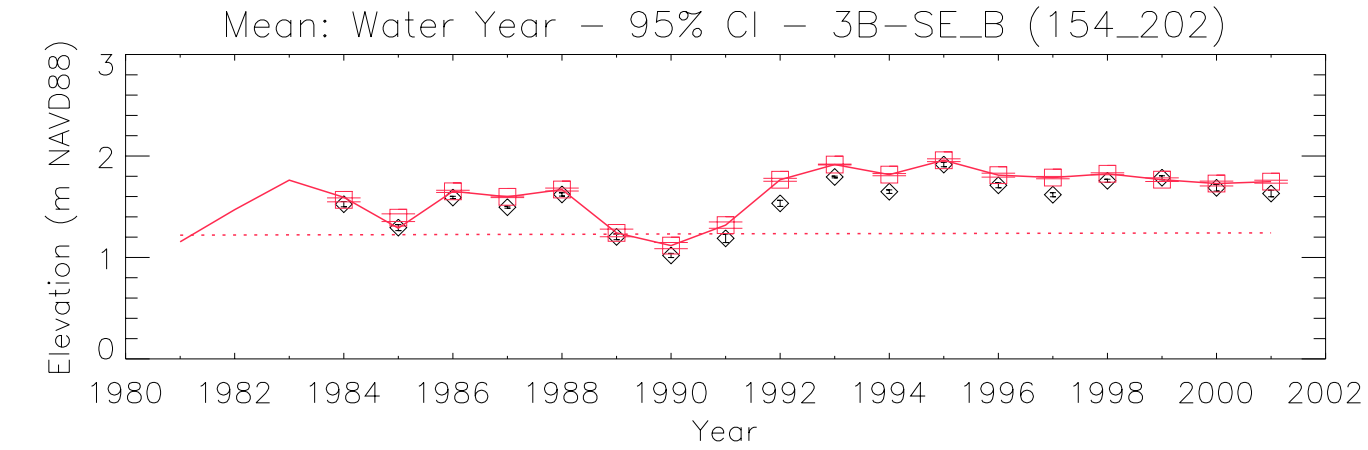

Mean: Season – 95% CI – SHARK.1_H (139_200)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

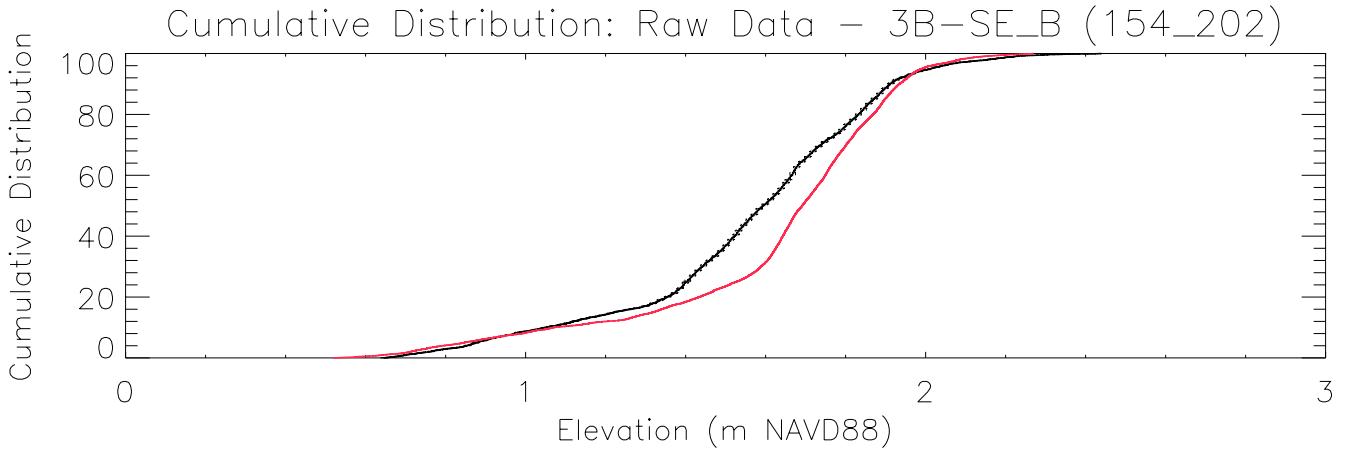

Mean: Water Year – 95% CI – SHARK.1_H (139_200)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

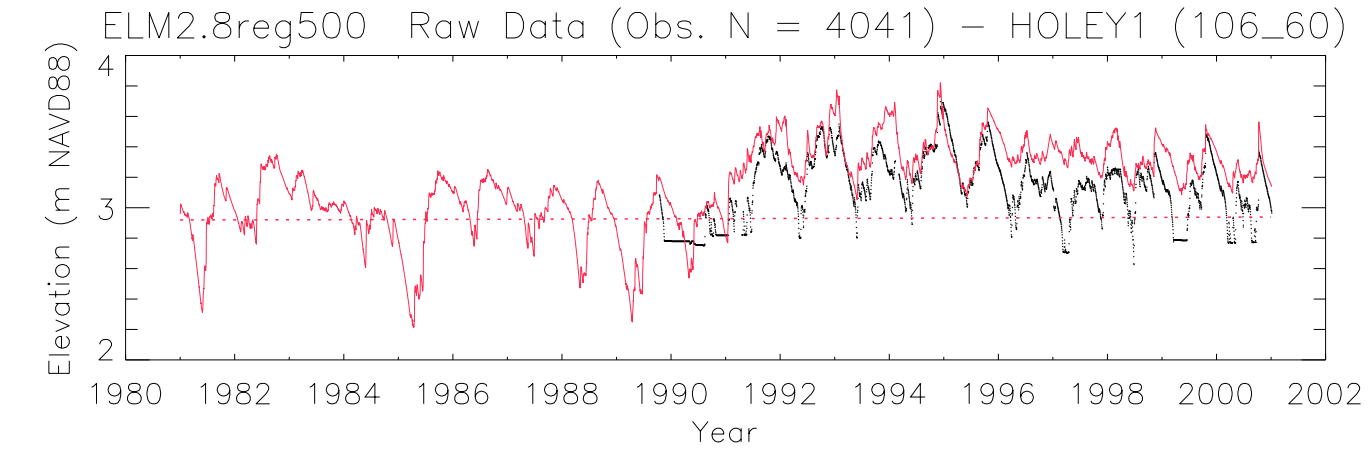

Cumulative Distribution: Raw Data – SHARK.1_H (139_200)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

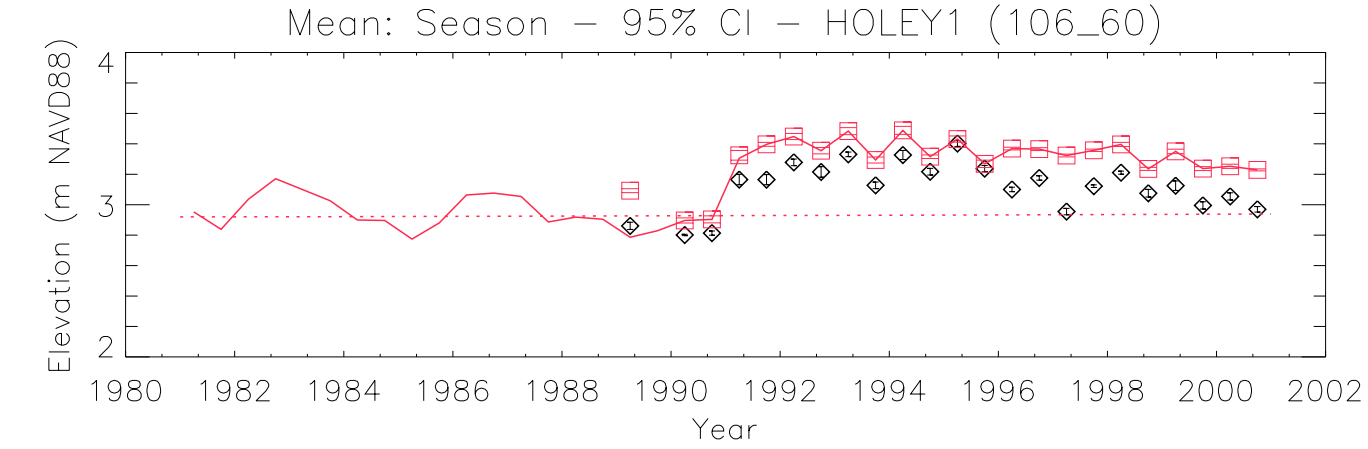

ELM2.8reg500 Raw Data (Obs. N = 6029) – 3B–SE_B (154_202)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

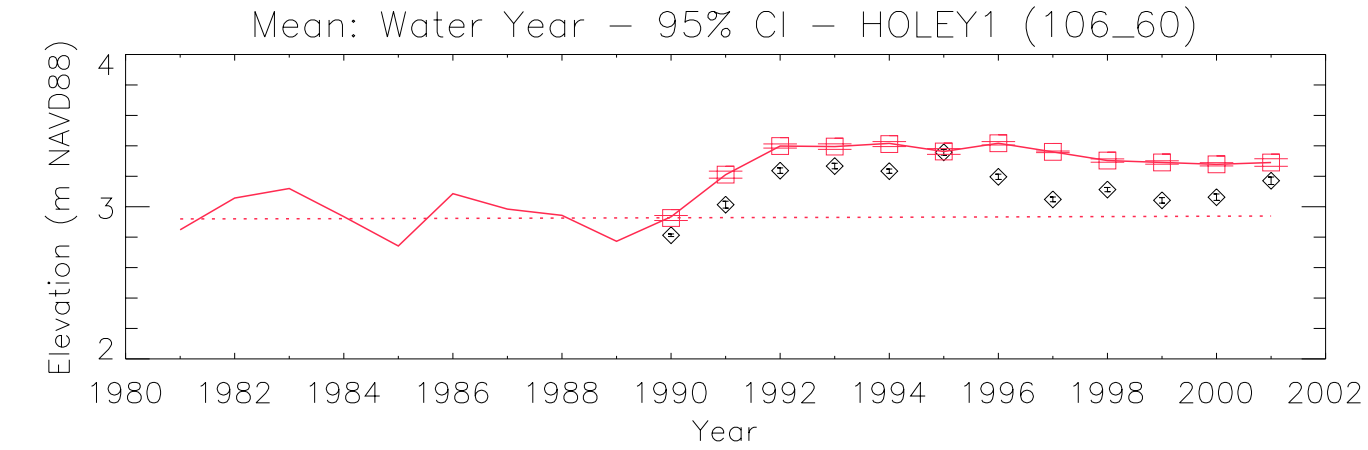

Mean: Season – 95% CI – 3B–SE_B (154_202)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

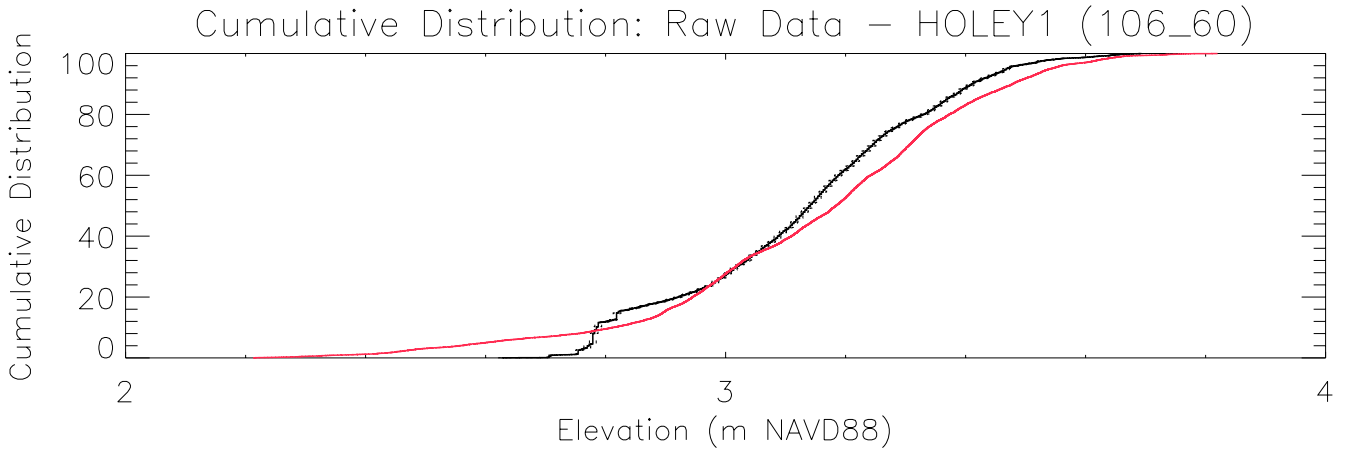

Mean: Water Year – 95% CI – 3B–SE_B (154_202)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

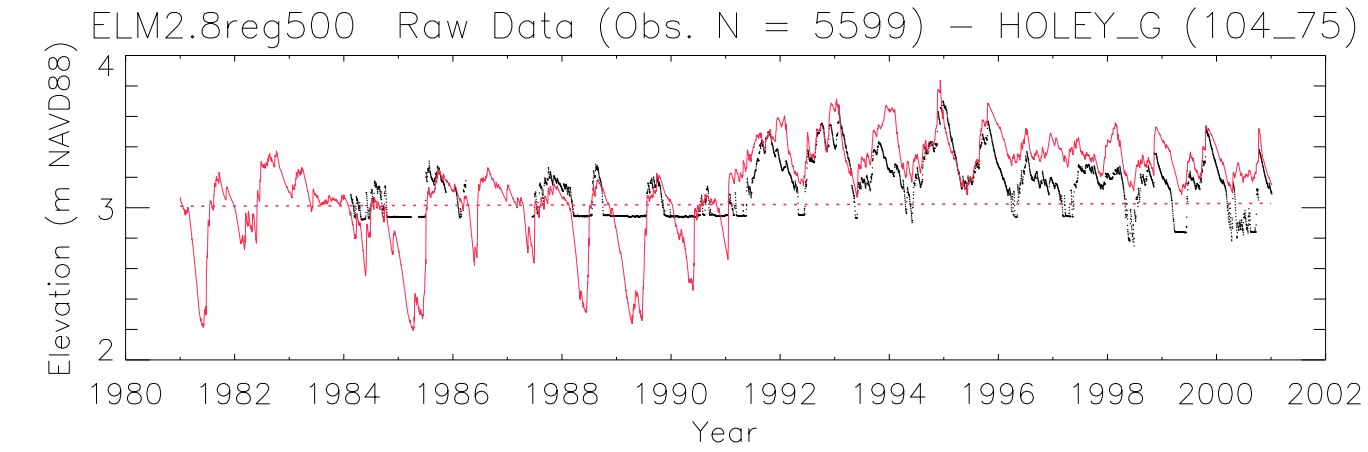

Cumulative Distribution: Raw Data – 3B–SE_B (154_202)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

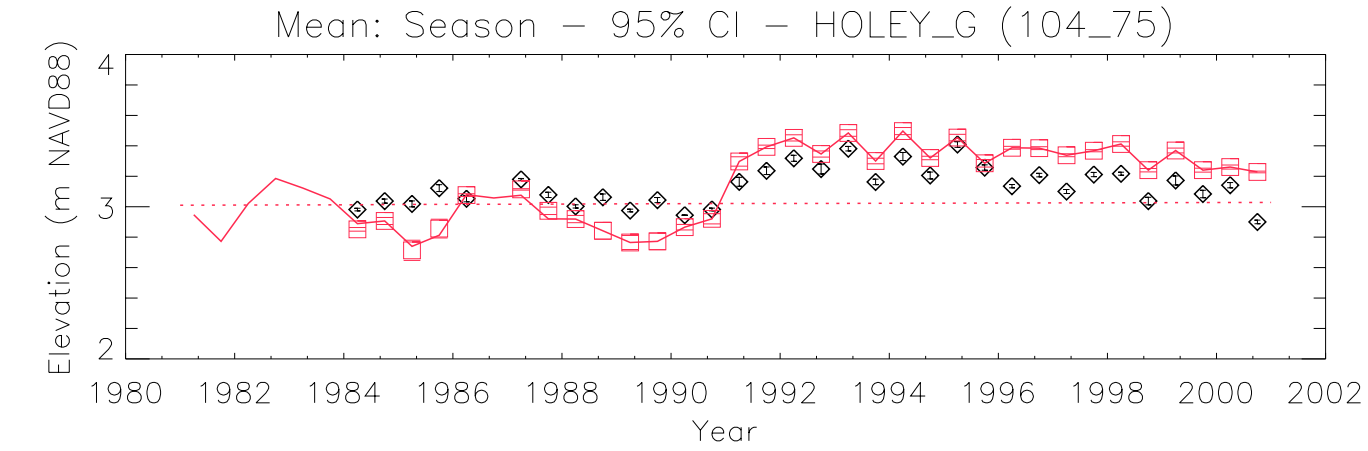

ELM2.8reg500 Raw Data (Obs. N = 4041) – HOLEY1 (106_60)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

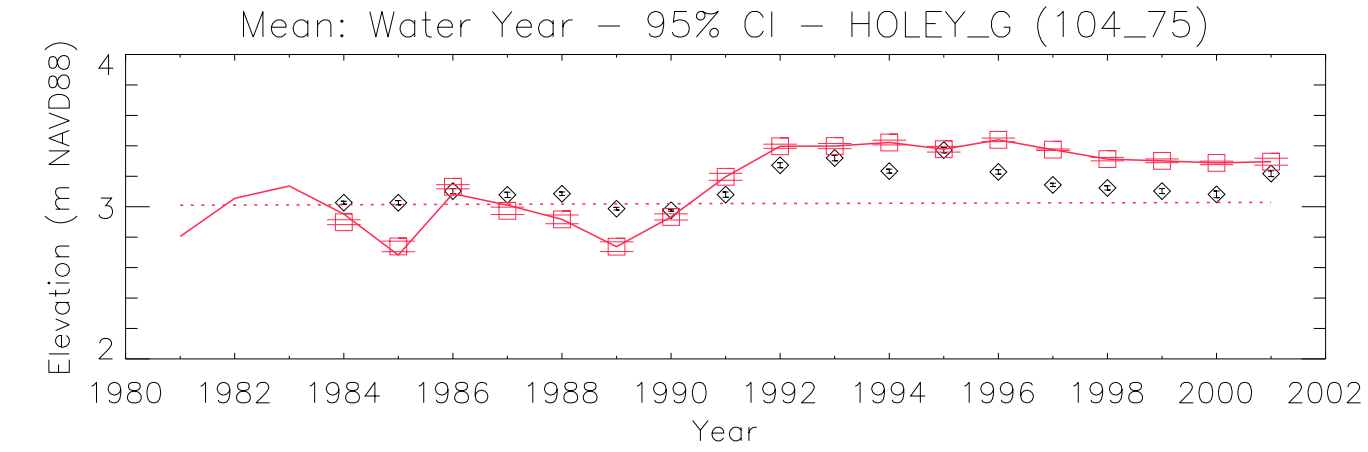

Mean: Season – 95% CI – HOLEY1 (106_60)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

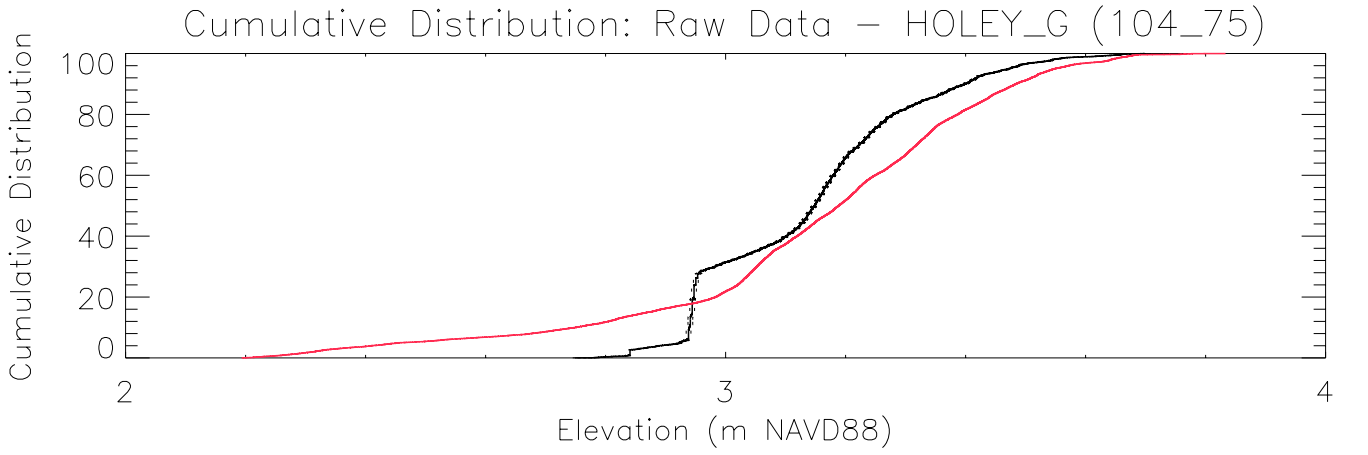

Mean: Water Year – 95% CI – HOLEY1 (106_60)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

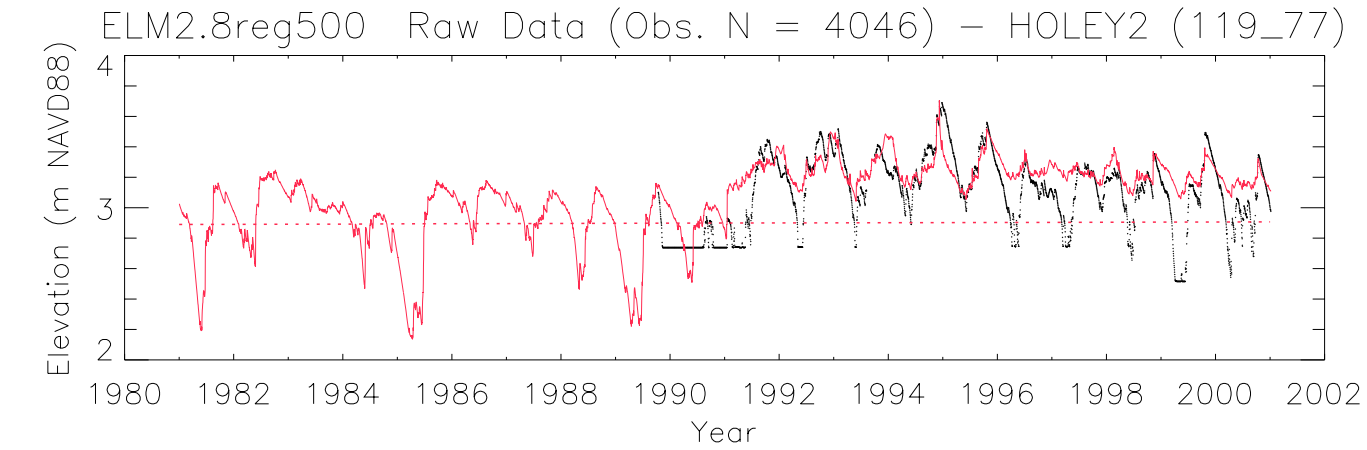

Cumulative Distribution: Raw Data – HOLEY1 (106_60)
Cumulative Distribution
100
80
60
40
20
0
2
3
4
Elevation (m NAVD88)

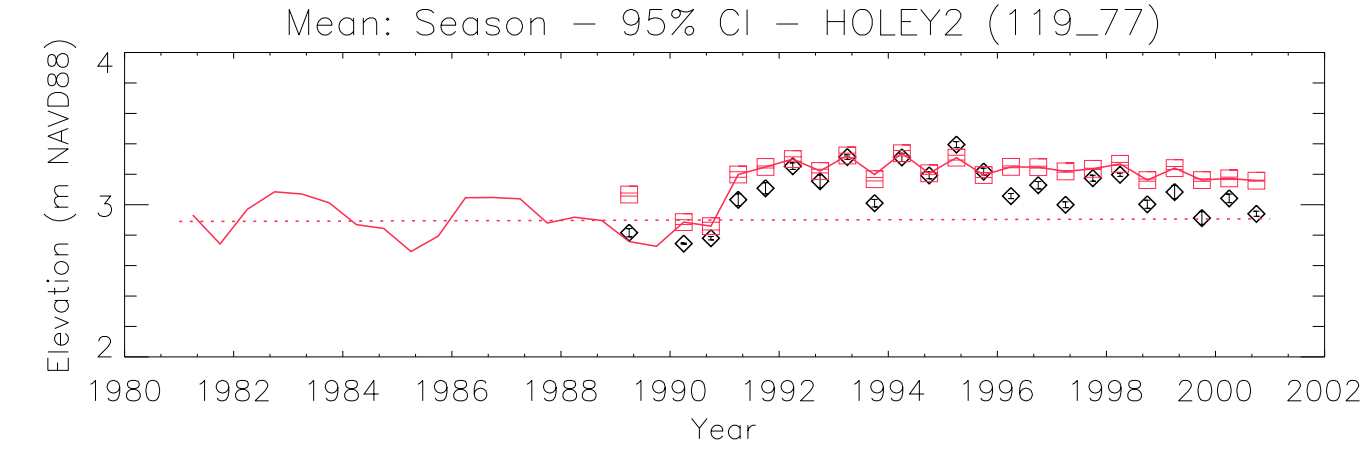

ELM2.8reg500 Raw Data (Obs. N = 5599) – HOLEY_G (104_75)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

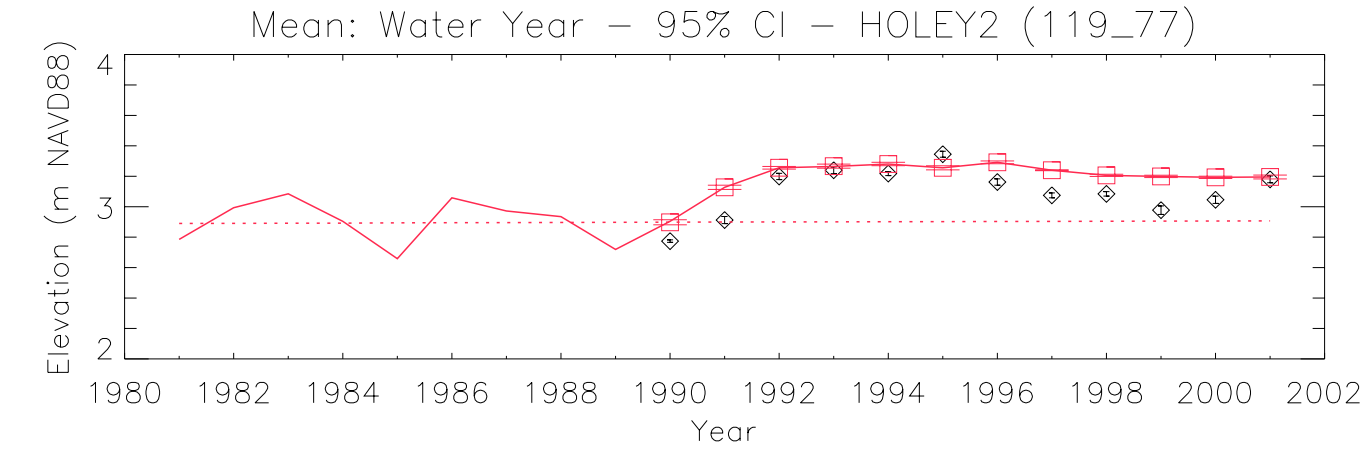

Mean: Season – 95% CI – HOLEY_G (104_75)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

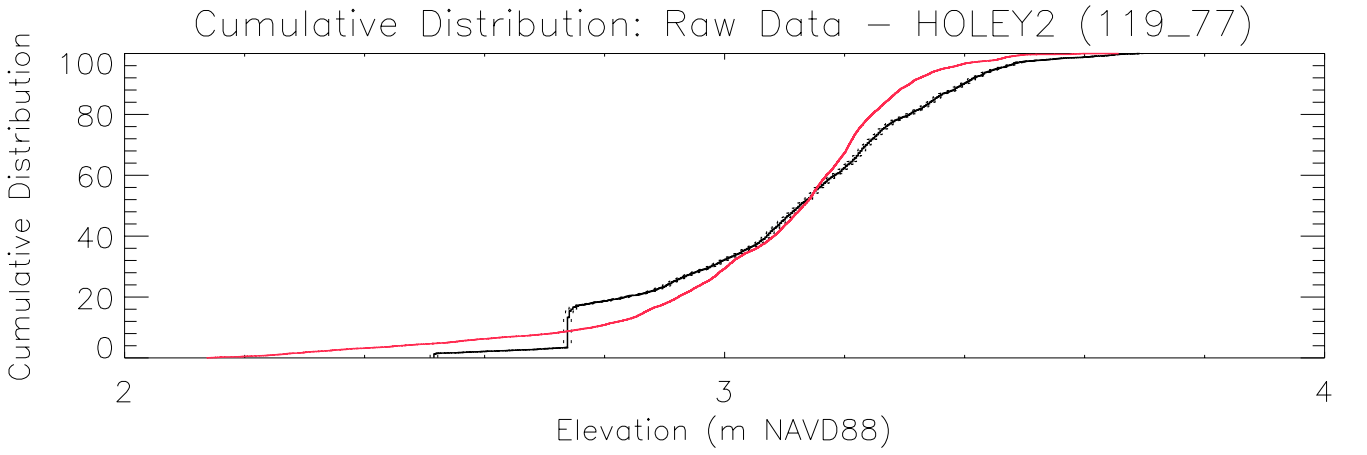

Mean: Water Year – 95% CI – HOLEY_G (104_75)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

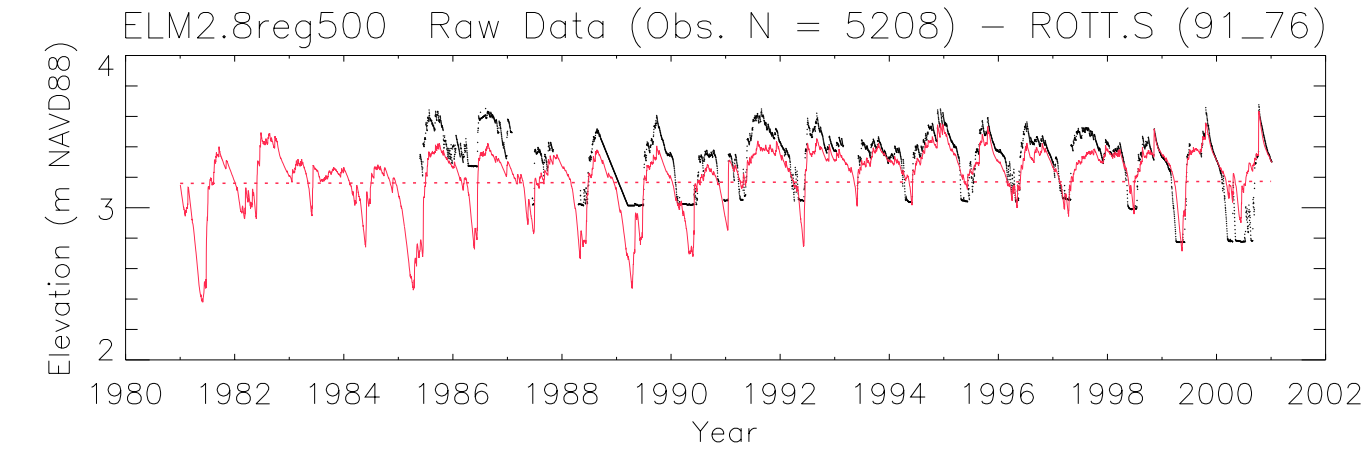

Cumulative Distribution: Raw Data – HOLEY_G (104_75)
Cumulative Distribution
100
80
60
40
20
0
2
3
4
Elevation (m NAVD88)

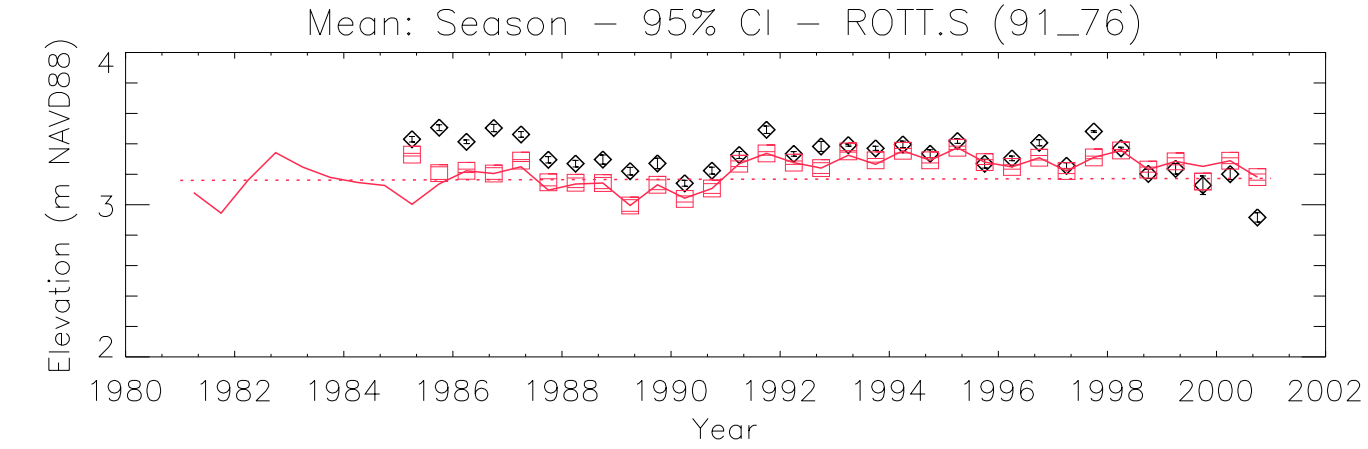

ELM2.8reg500 Raw Data (Obs. N = 4046) – HOLEY2 (119_77)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

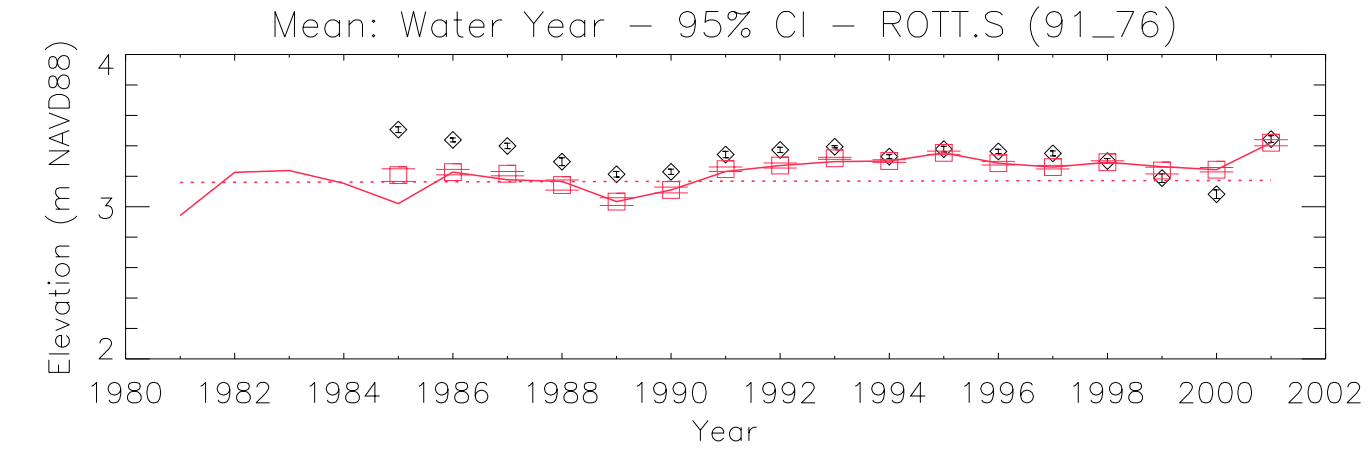

Mean: Season – 95% CI – HOLEY2 (119_77)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

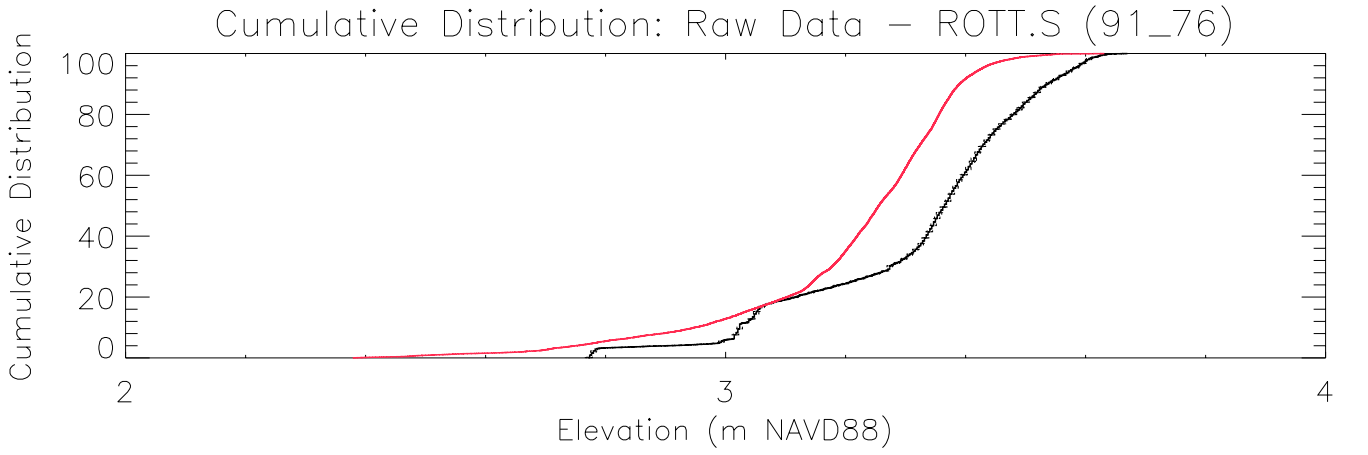

Mean: Water Year – 95% CI – HOLEY2 (119_77)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

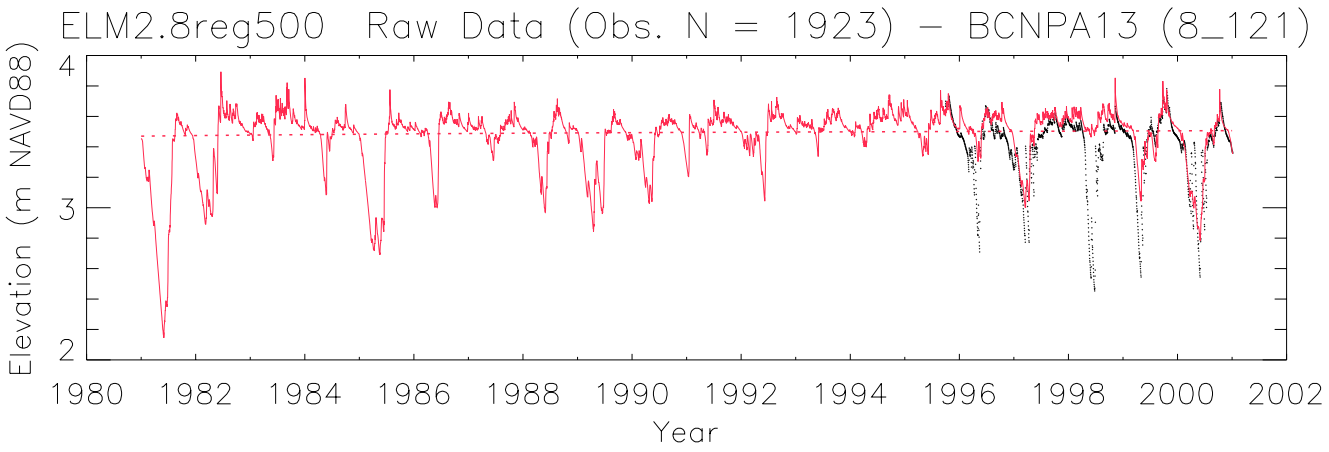

Cumulative Distribution: Raw Data – HOLEY2 (119_77)
Cumulative Distribution
100
80
60
40
20
0
2
3
4
Elevation (m NAVD88)

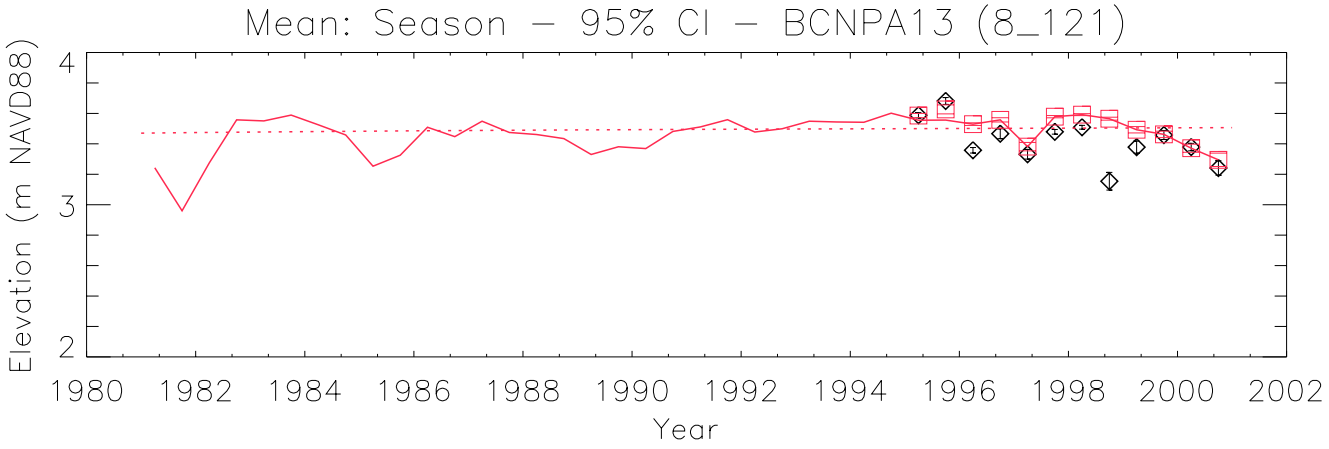

ELM2.8reg500 Raw Data (Obs. N = 5208) – ROTT.S (91_76)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

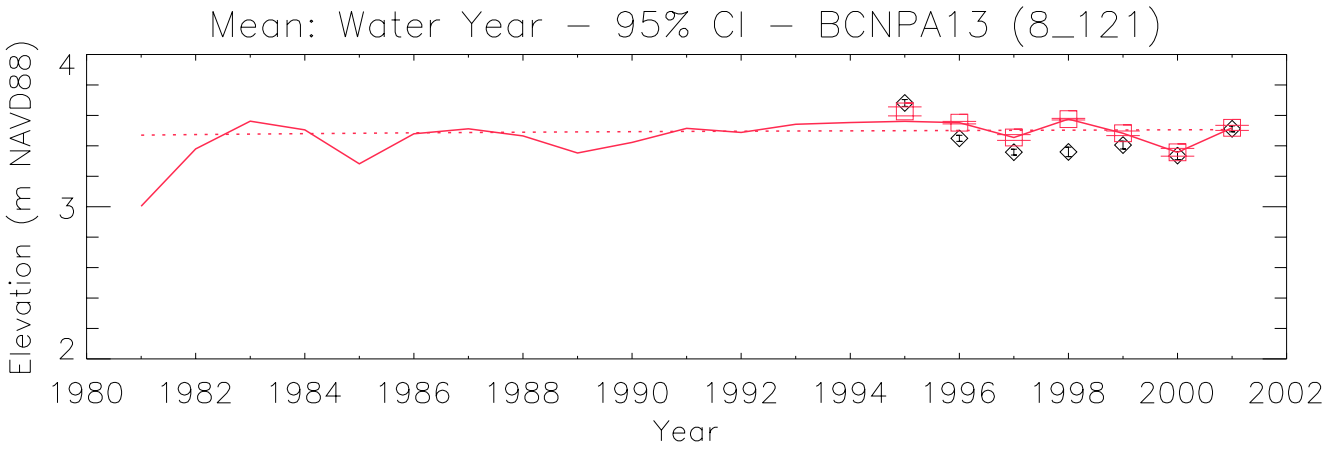

Mean: Season – 95% CI – ROTT.S (91_76)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

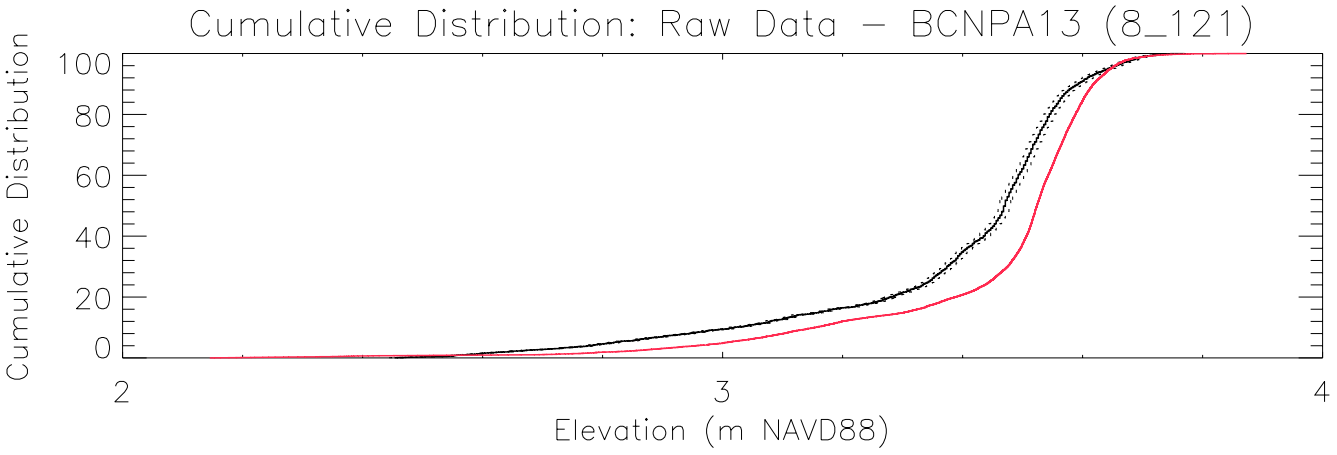

Mean: Water Year – 95% CI – ROTT.S (91_76)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

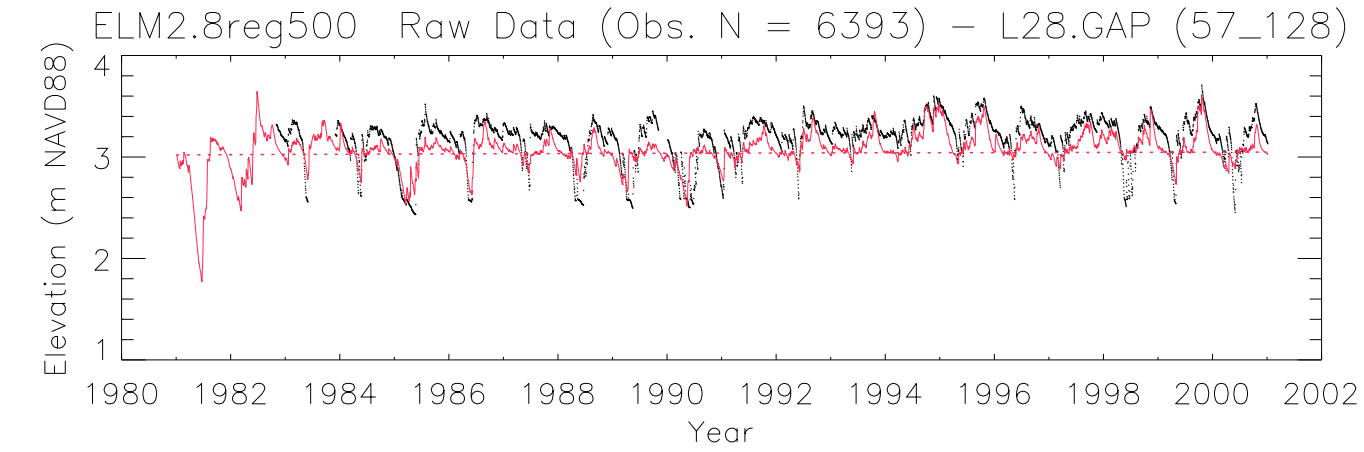

Cumulative Distribution: Raw Data – ROTT.S (91_76)
Cumulative Distribution
100
80
60
40
20
0
2
3
4
Elevation (m NAVD88)

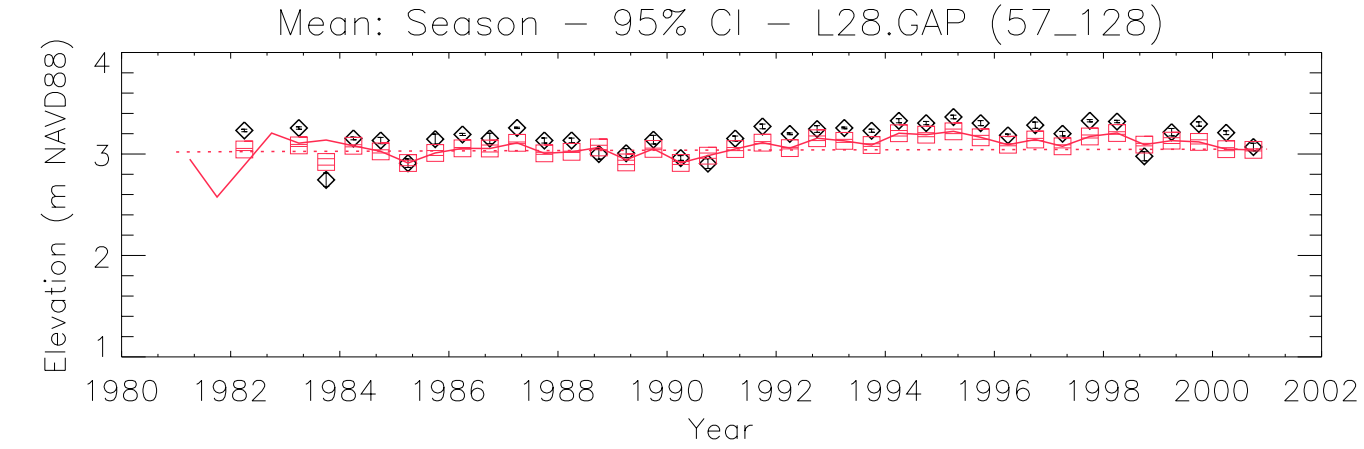

ELM2.8reg500 Raw Data (Obs. N = 1923) – BCNPA13 (8_121)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

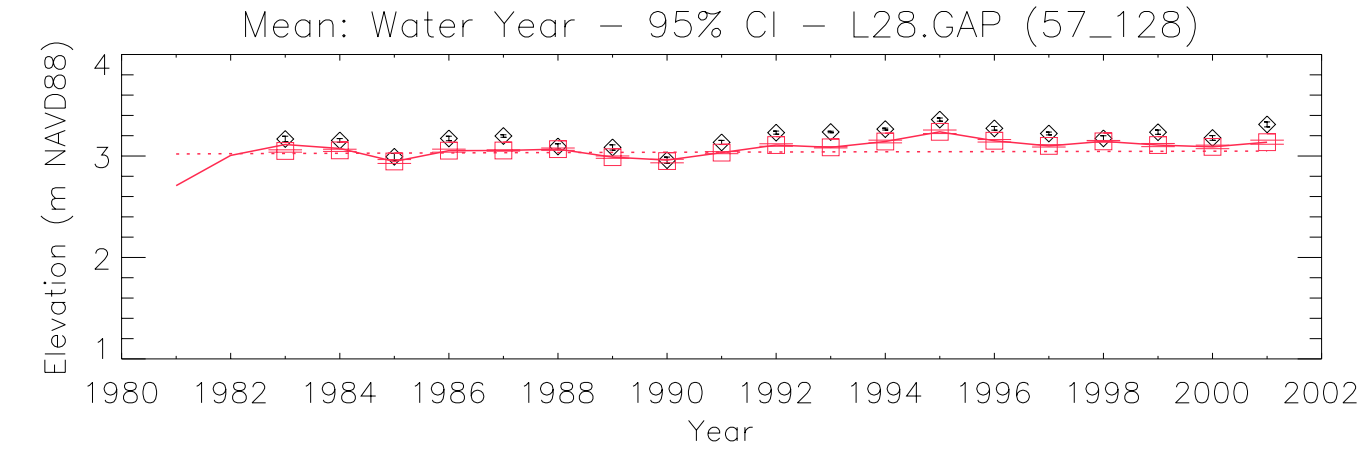

Mean: Season – 95% CI – BCNPA13 (8_121)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

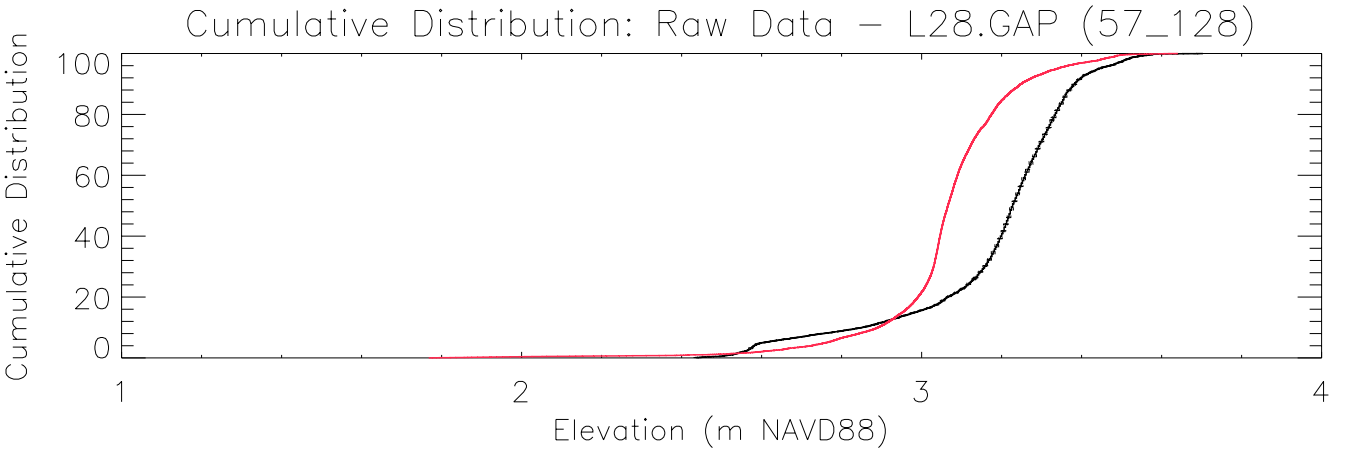

Mean: Water Year – 95% CI – BCNPA13 (8_121)
Elevation (m NAVD88)
4
3
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

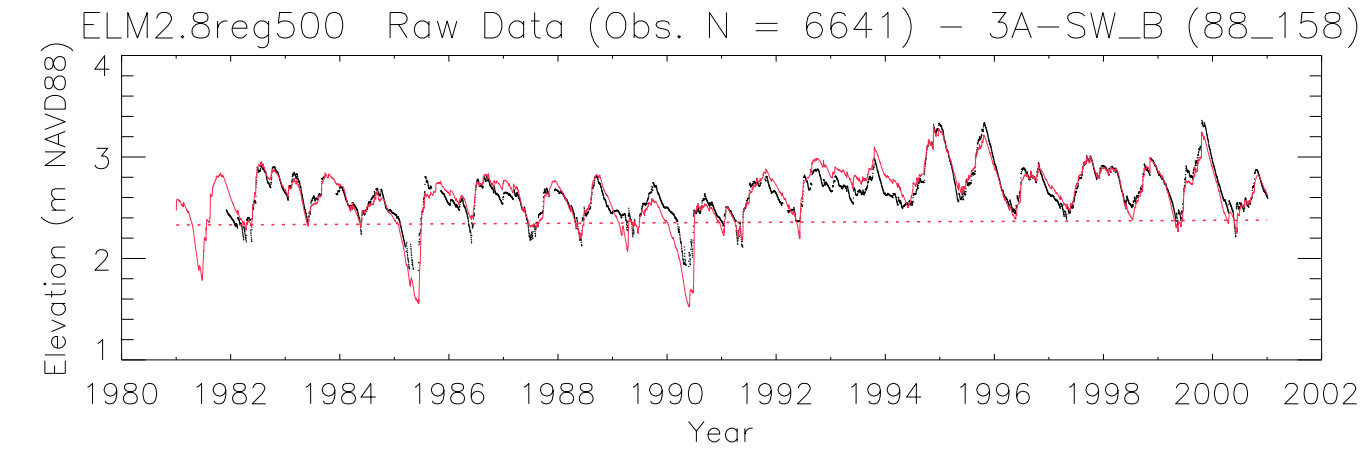

Cumulative Distribution: Raw Data – BCNPA13 (8_121)
Cumulative Distribution
100
80
60
40
20
0
2
3
4
Elevation (m NAVD88)

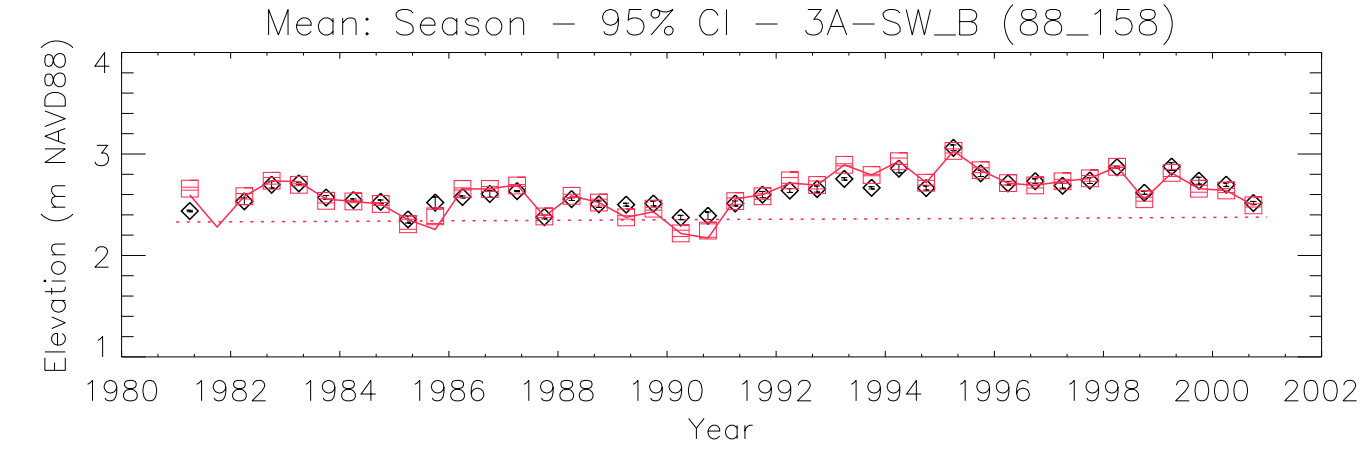

ELM2.8reg500 Raw Data (Obs. N = 6393) – L28.GAP (57_128)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

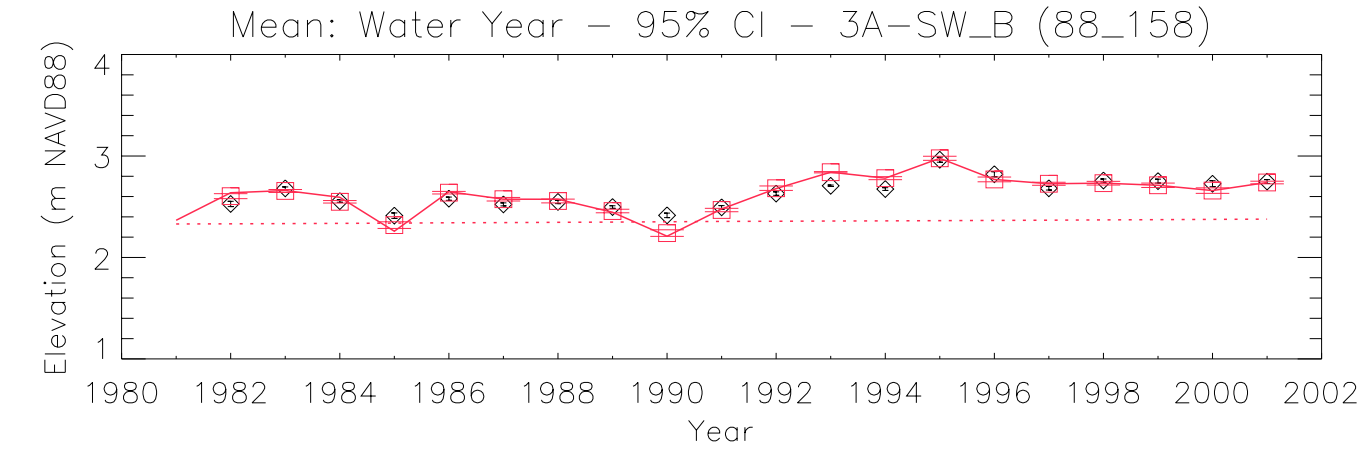

Mean: Season – 95% CI – L28.GAP (57_128)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

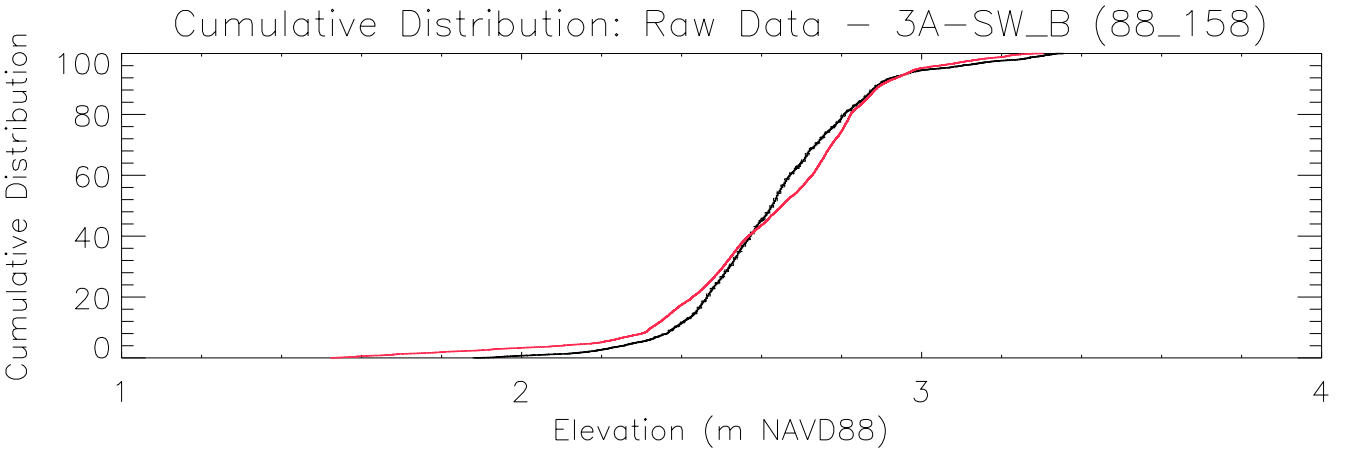

Mean: Water Year – 95% CI – L28.GAP (57_128)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

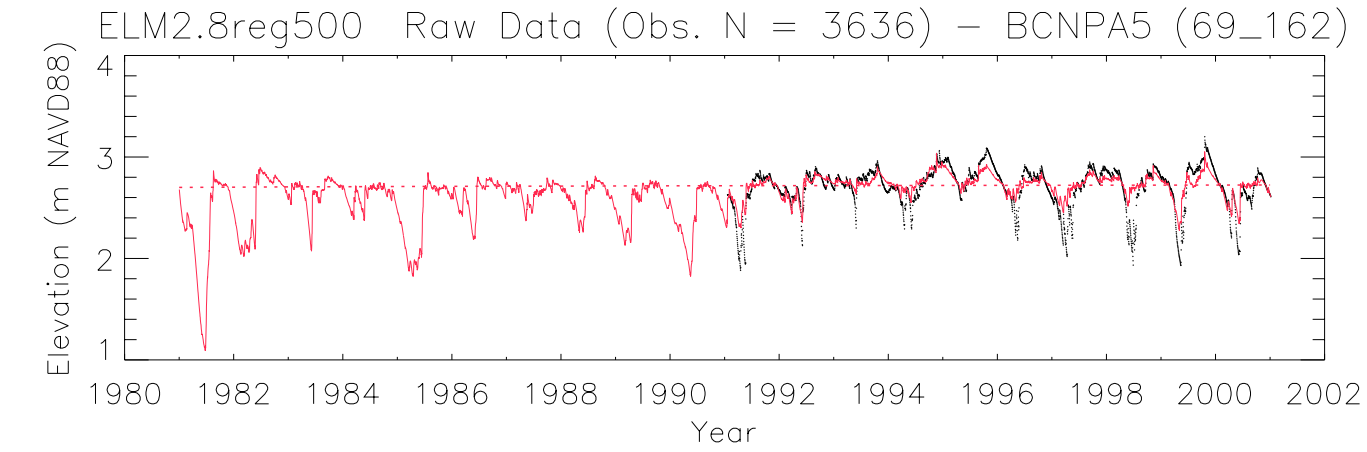

Cumulative Distribution: Raw Data – L28.GAP (57_128)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

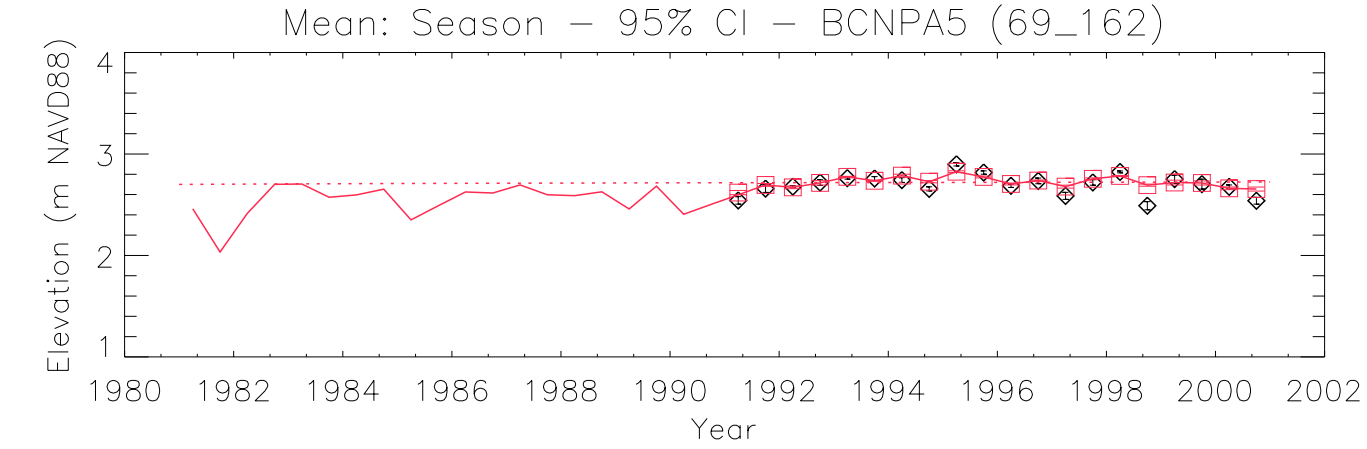

ELM2.8reg500 Raw Data (Obs. N = 6641) – 3A–SW_B (88_158)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

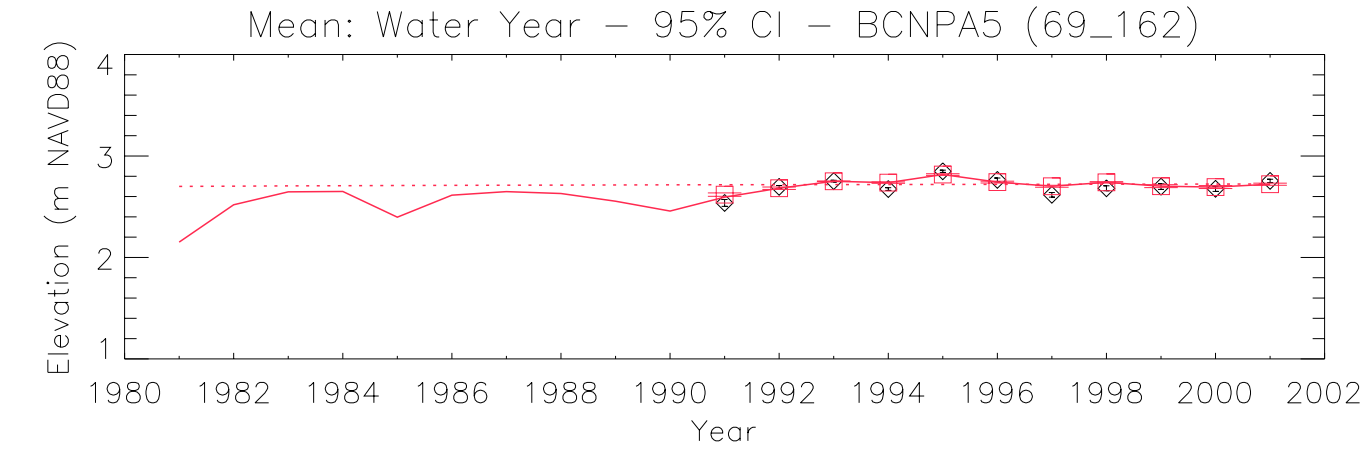

Mean: Season – 95% CI – 3A–SW_B (88_158)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

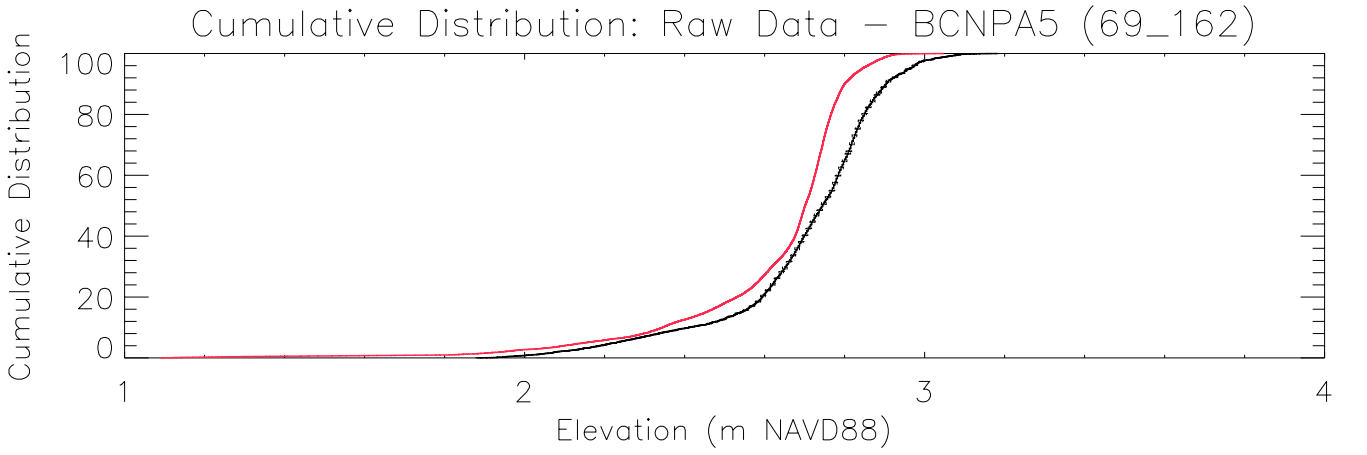

Mean: Water Year – 95% CI – 3A–SW_B (88_158)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

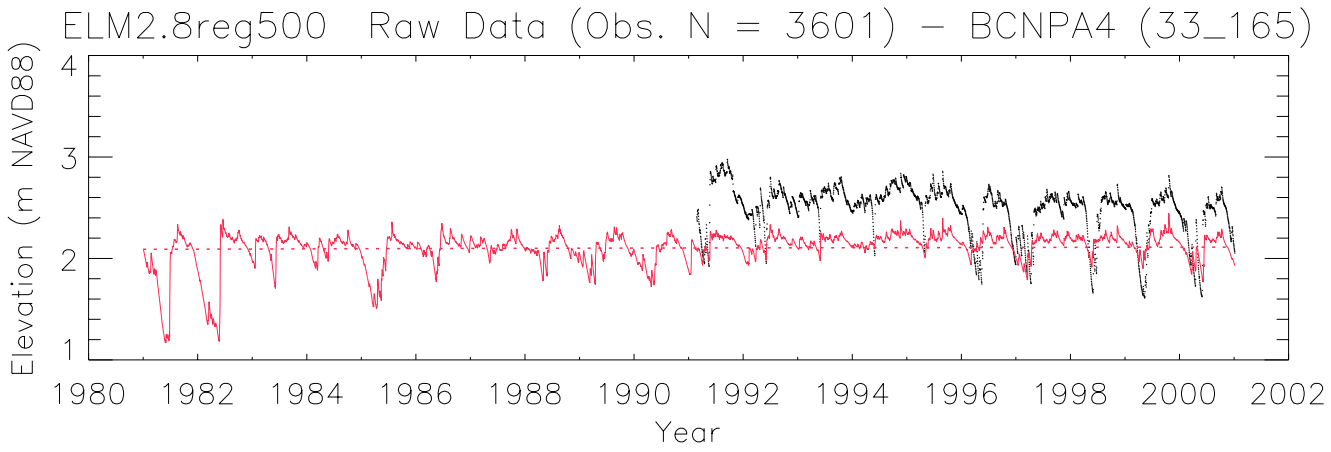

Cumulative Distribution: Raw Data – 3A–SW_B (88_158)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

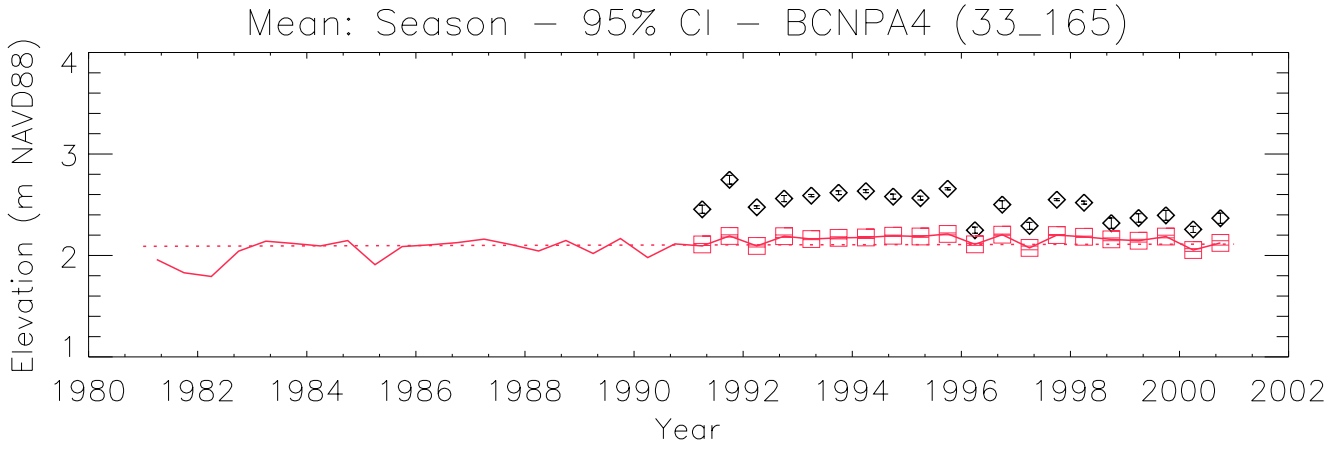

ELM2.8reg500 Raw Data (Obs. N = 3636) – BCNPA5 (69_162)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

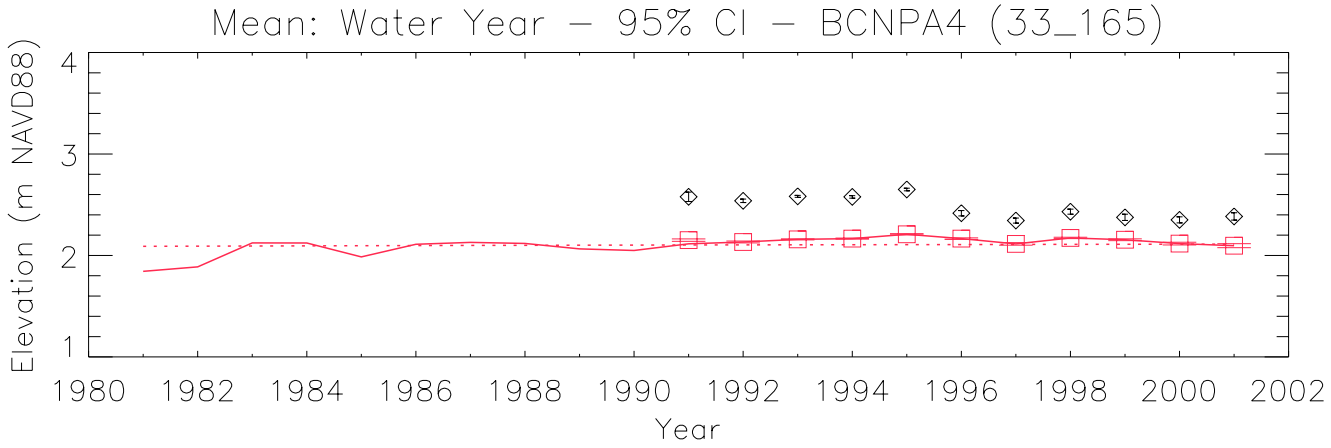

Mean: Season – 95% CI – BCNPA5 (69_162)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

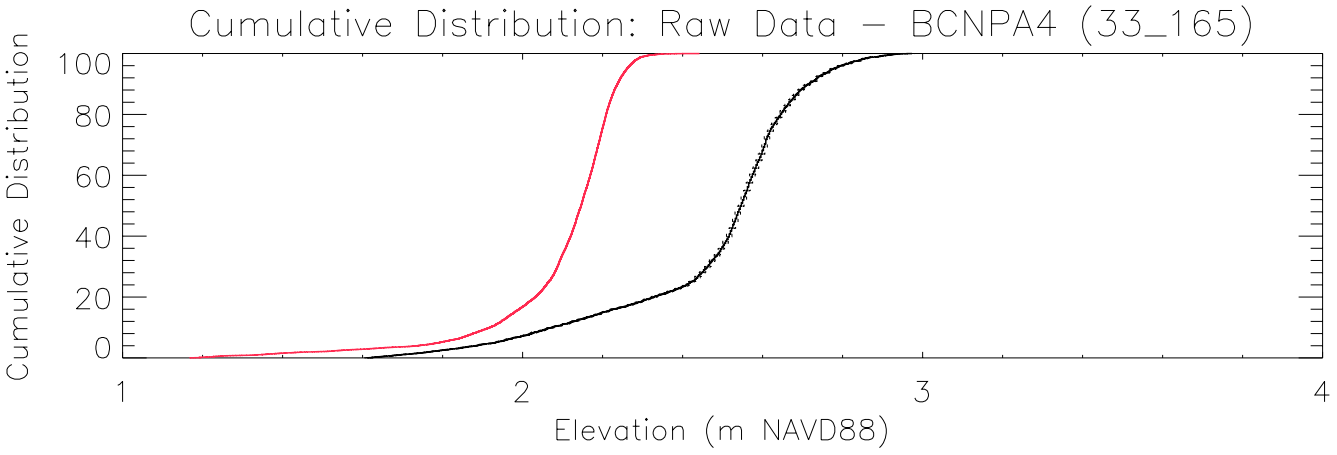

Mean: Water Year – 95% CI – BCNPA5 (69_162)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

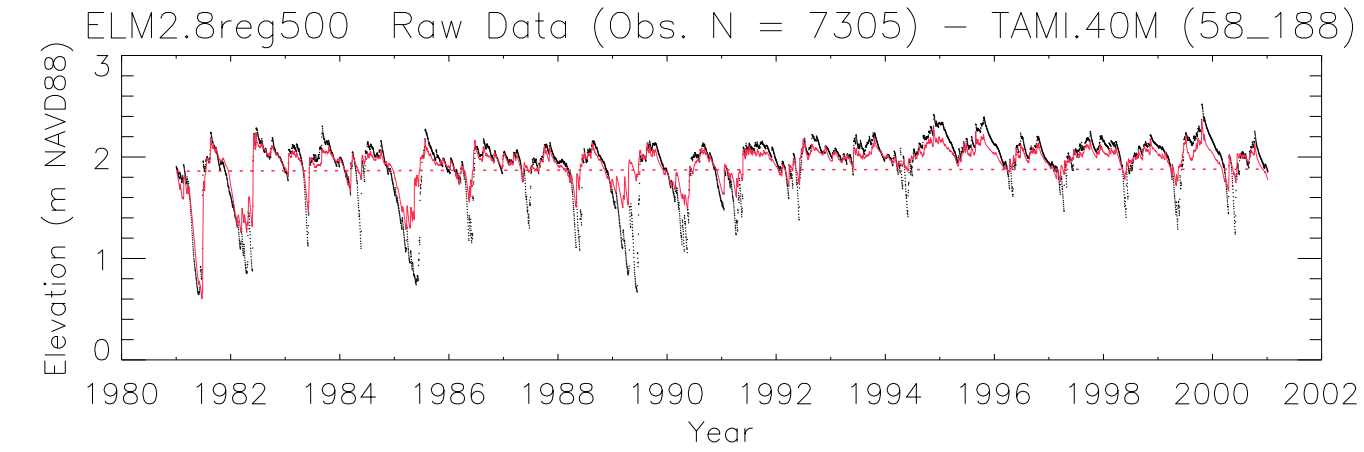

Cumulative Distribution: Raw Data – BCNPA5 (69_162)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

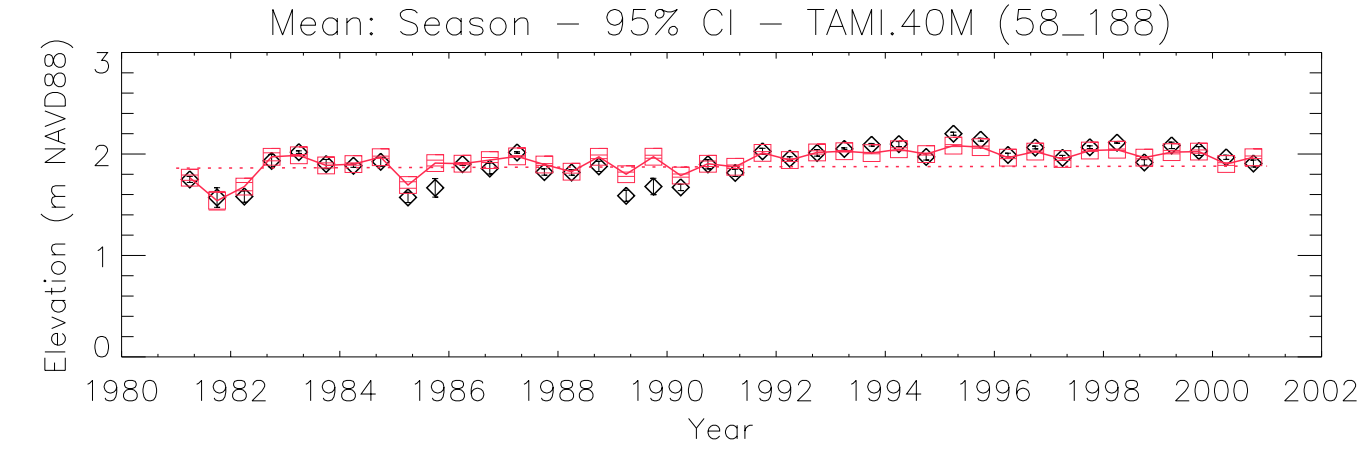

ELM2.8reg500 Raw Data (Obs. N = 3601) – BCNPA4 (33_165)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

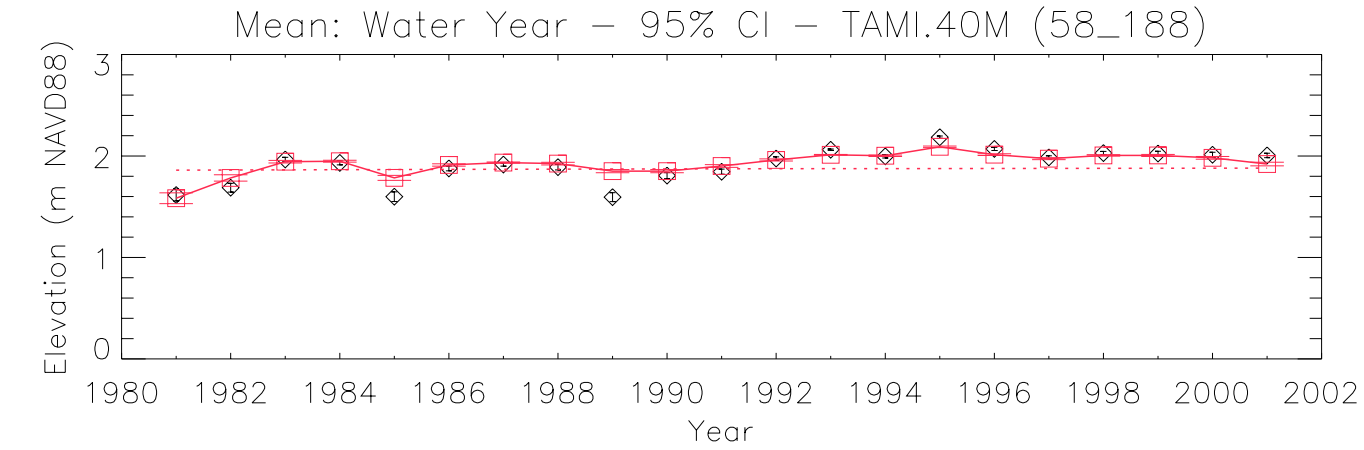

Mean: Season – 95% CI – BCNPA4 (33_165)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

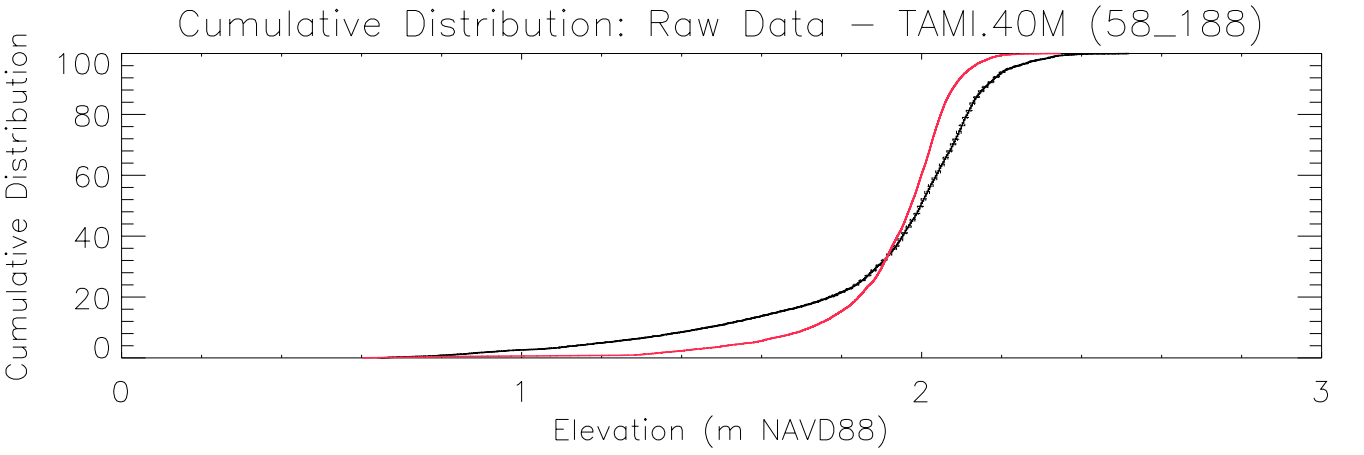

Mean: Water Year – 95% CI – BCNPA4 (33_165)
Elevation (m NAVD88)
1
2
3
4
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

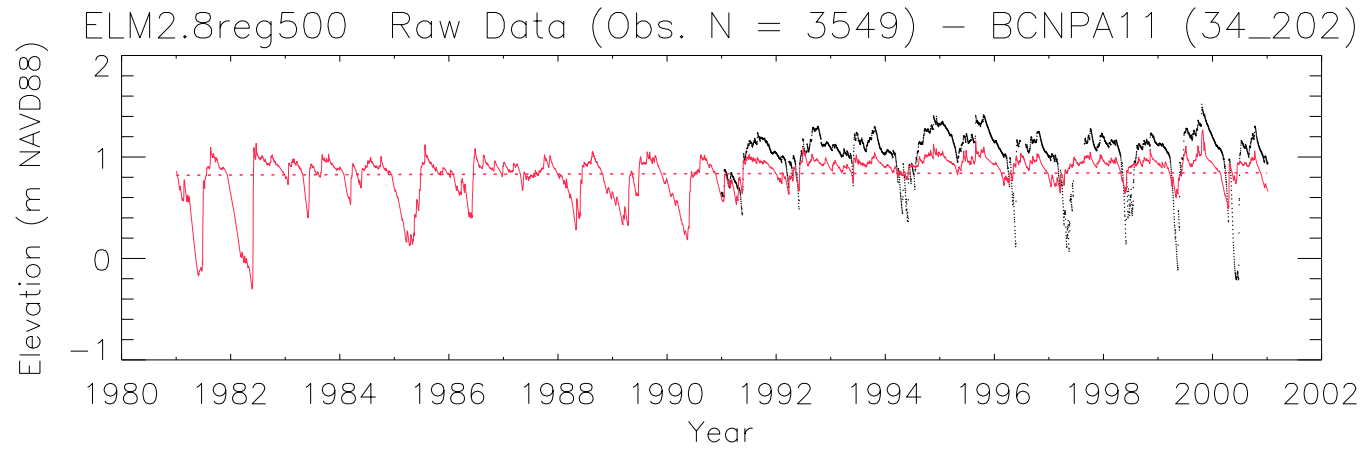

Cumulative Distribution: Raw Data – BCNPA4 (33_165)
Cumulative Distribution
0
20
40
60
80
100
1
2
3
4
Elevation (m NAVD88)

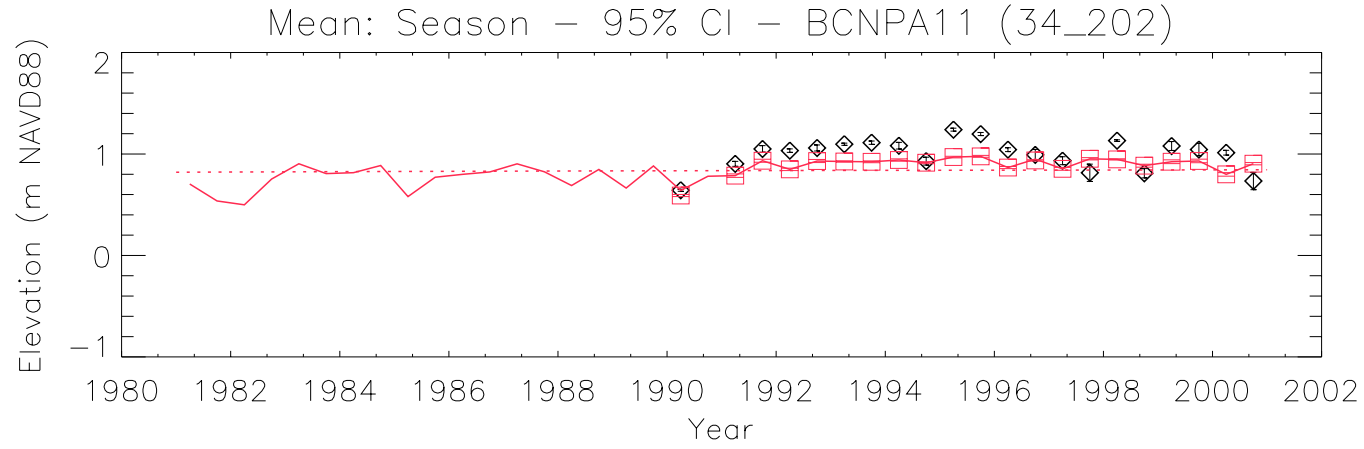

ELM2.8reg500 Raw Data (Obs. N = 7305) – TAMI.40M (58_188)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

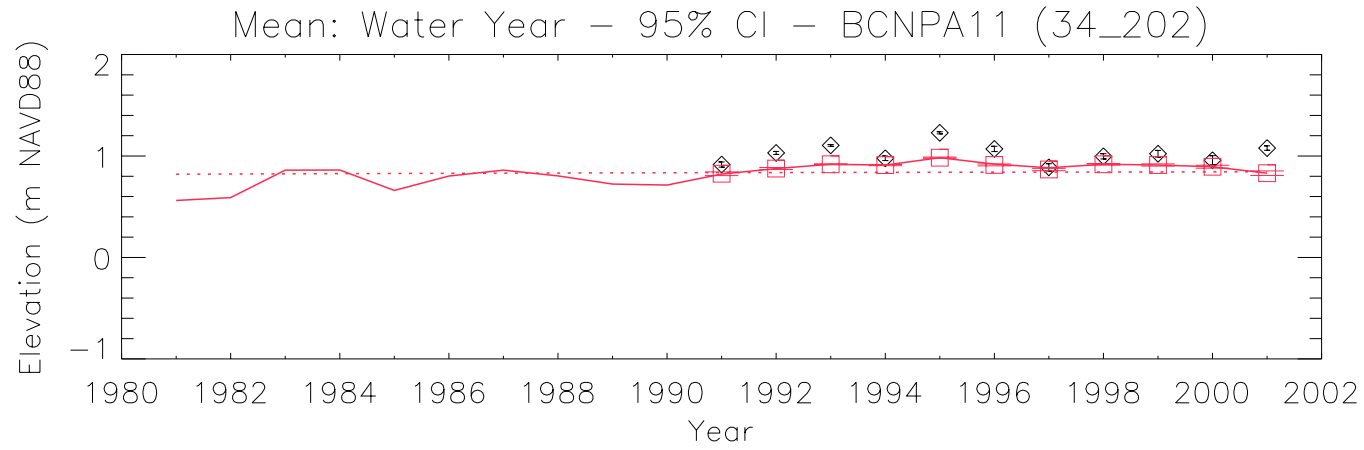

Mean: Season – 95% CI – TAMI.40M (58_188)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

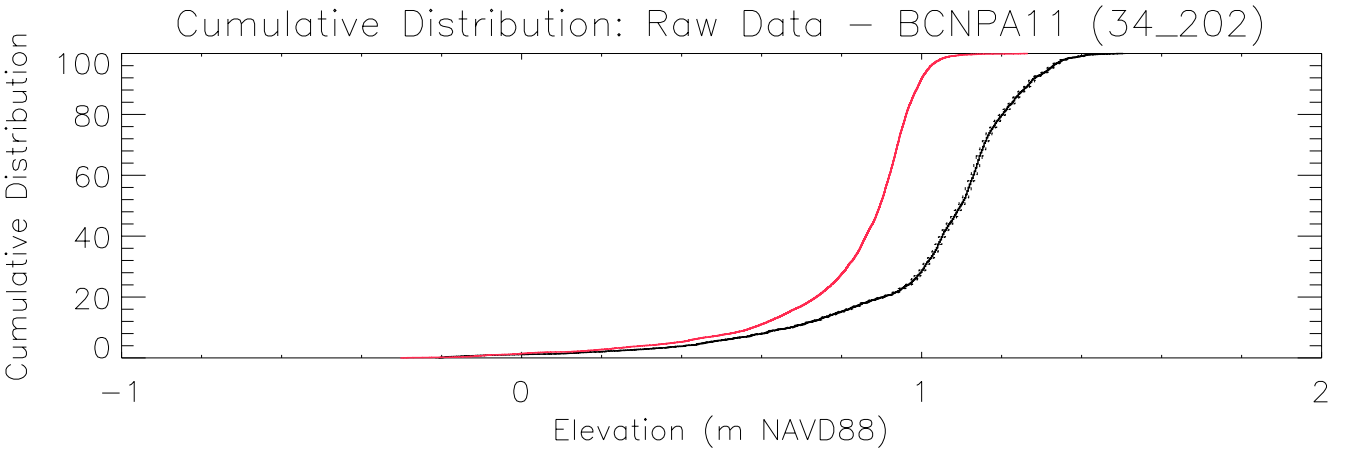

Mean: Water Year – 95% CI – TAMI.40M (58_188)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

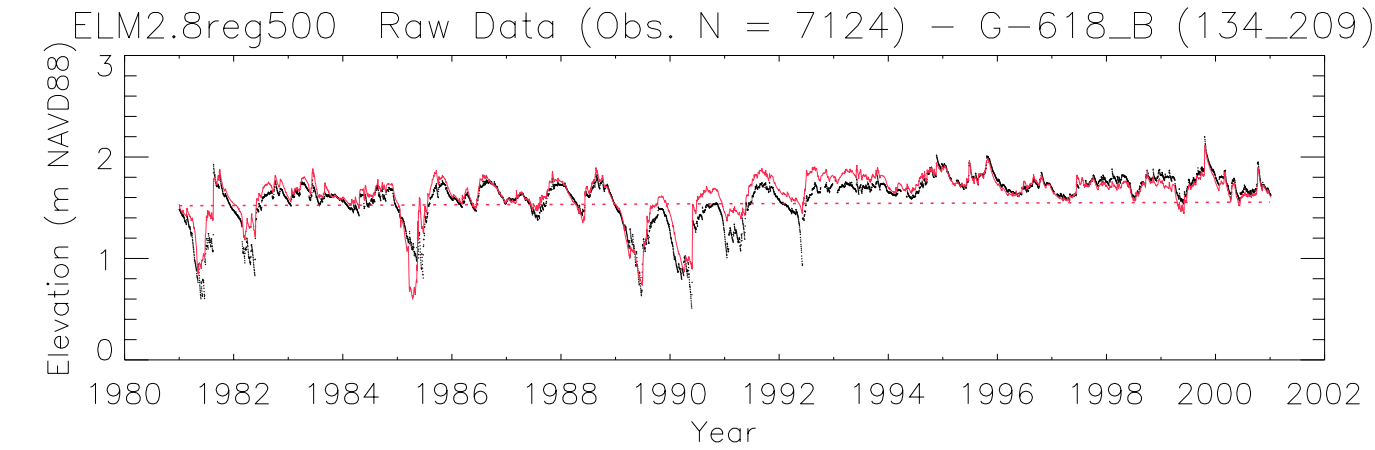

Cumulative Distribution: Raw Data – TAMI.40M (58_188)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

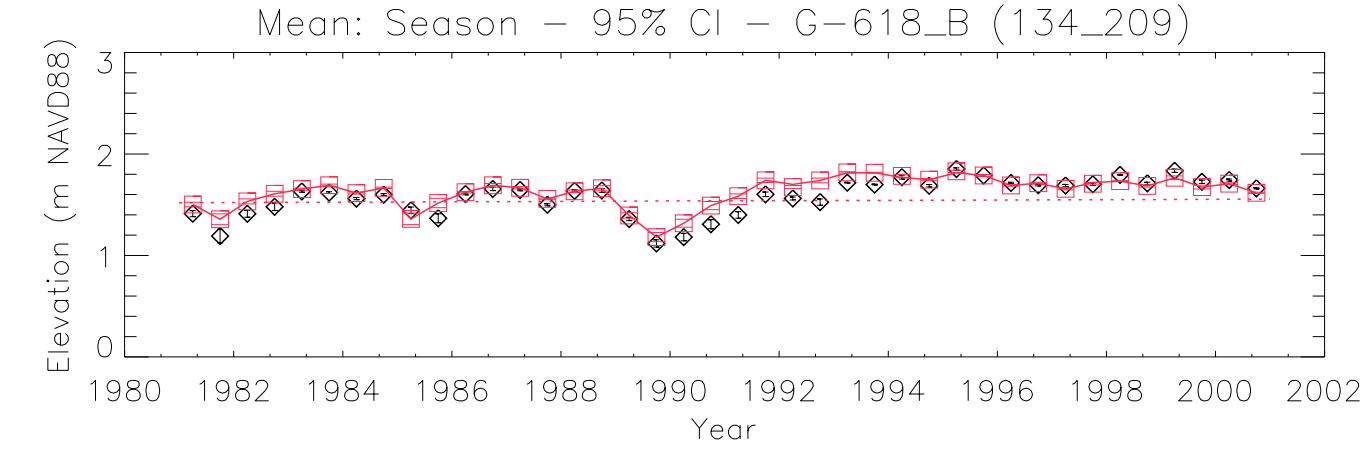

ELM2.8reg500 Raw Data (Obs. N = 3549) – BCNPA11 (34_202)
Elevation (m NAVD88)
2
1
0
−1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

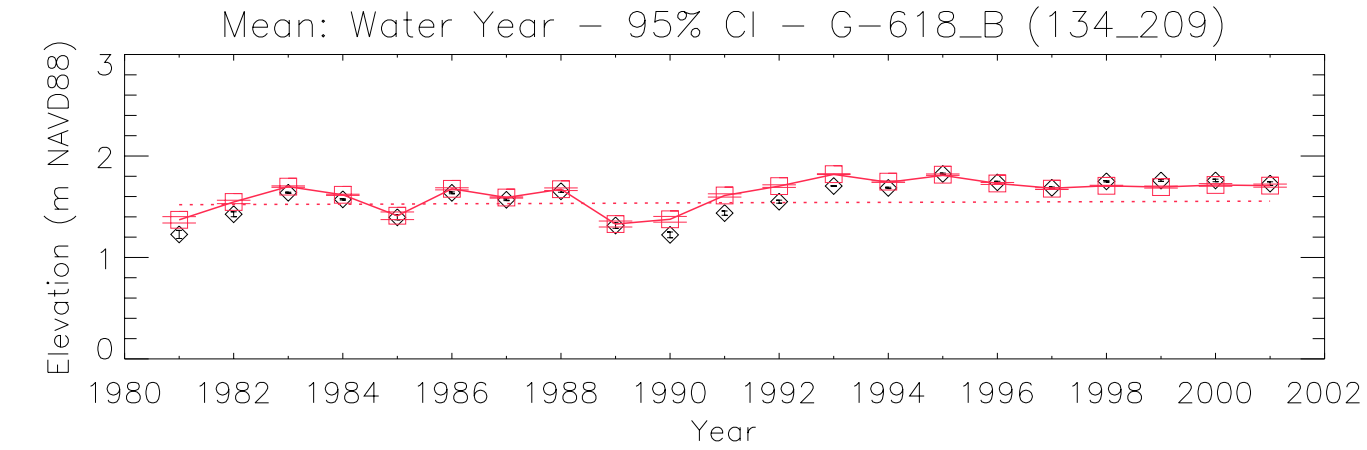

Mean: Season – 95% CI – BCNPA11 (34_202)
Elevation (m NAVD88)
2
1
0
−1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

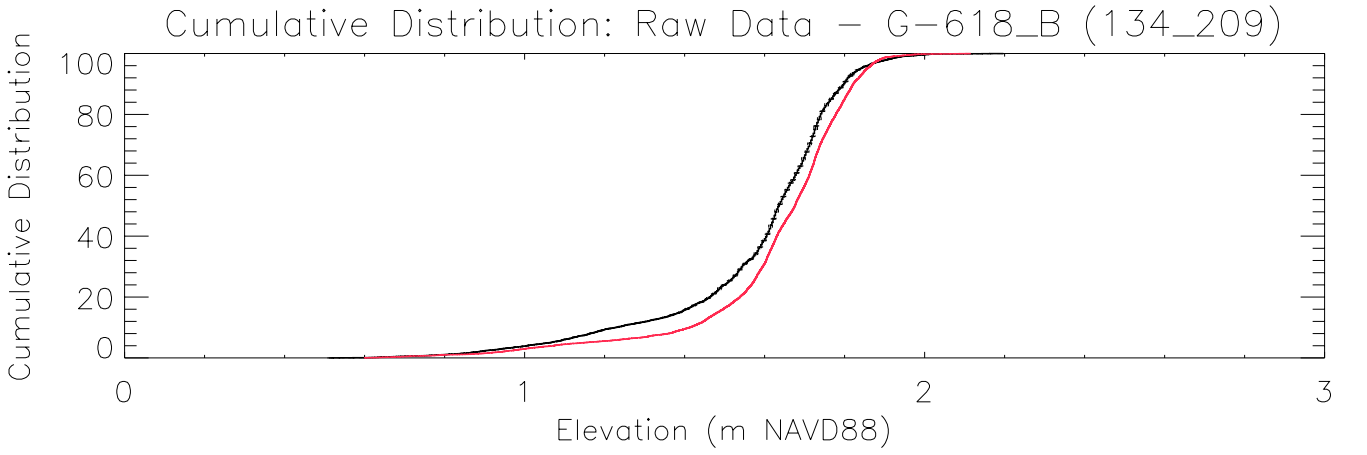

Mean: Water Year – 95% CI – BCNPA11 (34_202)
Elevation (m NAVD88)
2
1
0
−1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

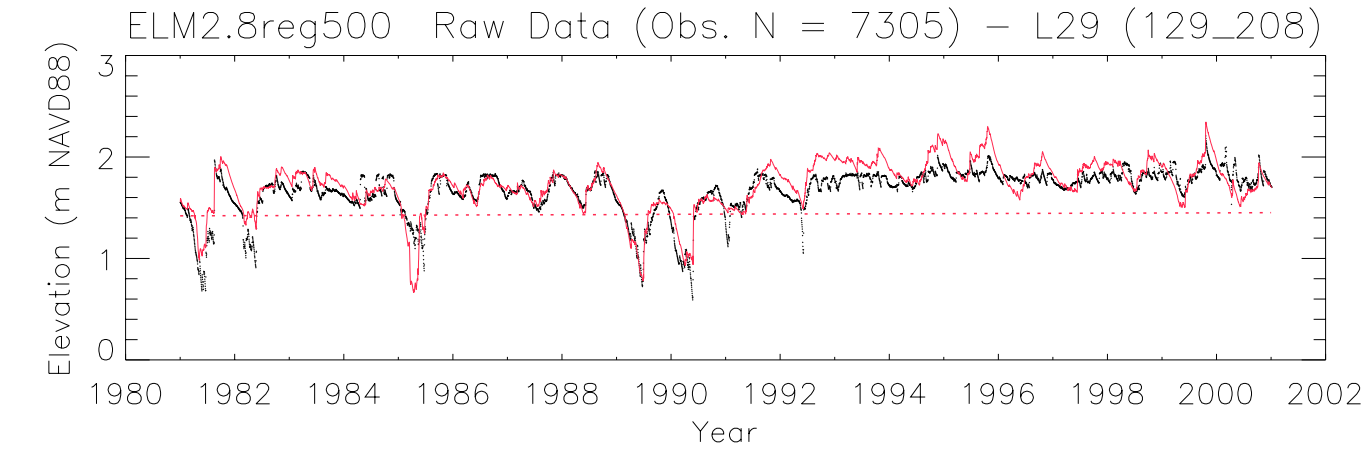

Cumulative Distribution: Raw Data – BCNPA11 (34_202)
Cumulative Distribution
100
80
60
40
20
0
−1
0
1
2
Elevation (m NAVD88)

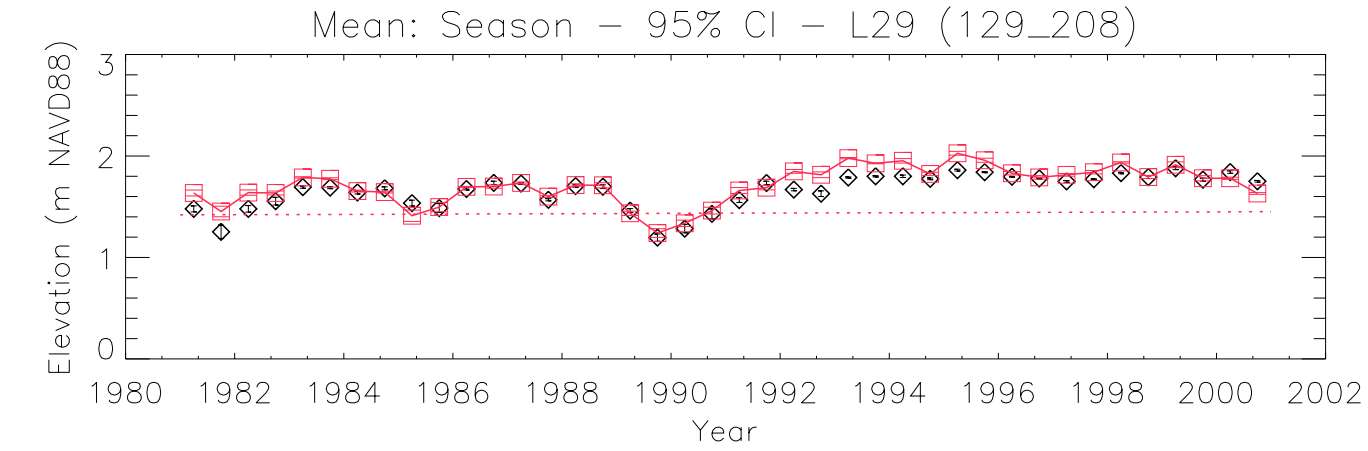

ELM2.8reg500 Raw Data (Obs. N = 7124) – G–618_B (134_209)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

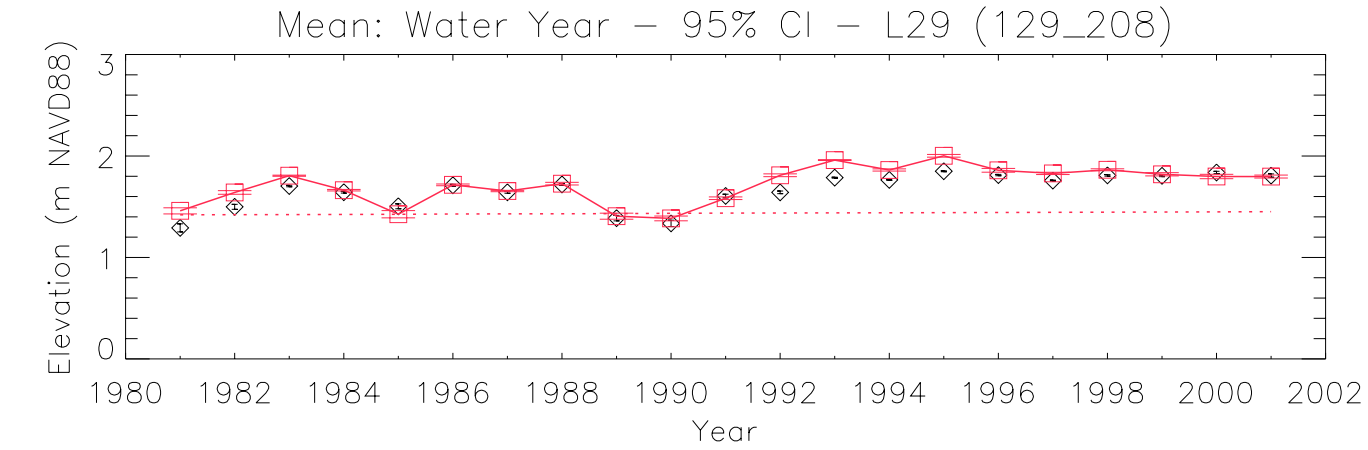

Mean: Season – 95% CI – G–618_B (134_209)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

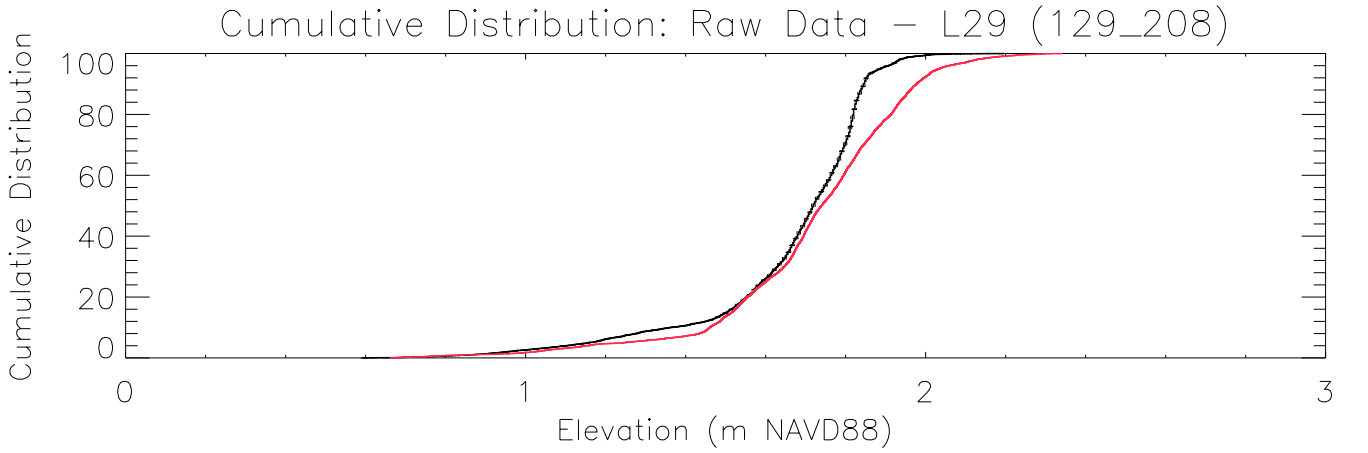

Mean: Water Year – 95% CI – G–618_B (134_209)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

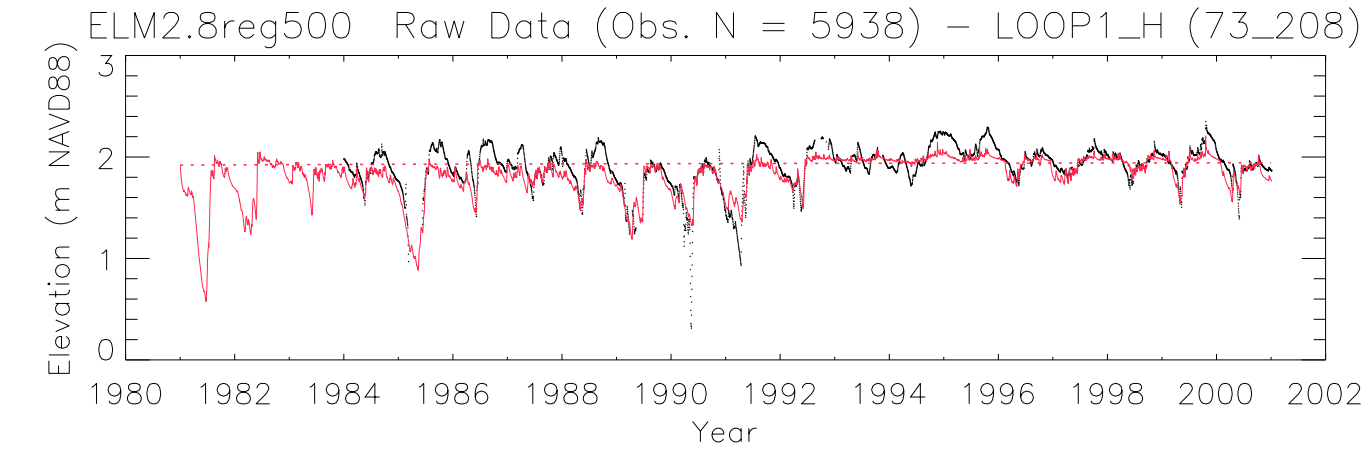

Cumulative Distribution: Raw Data – G–618_B (134_209)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

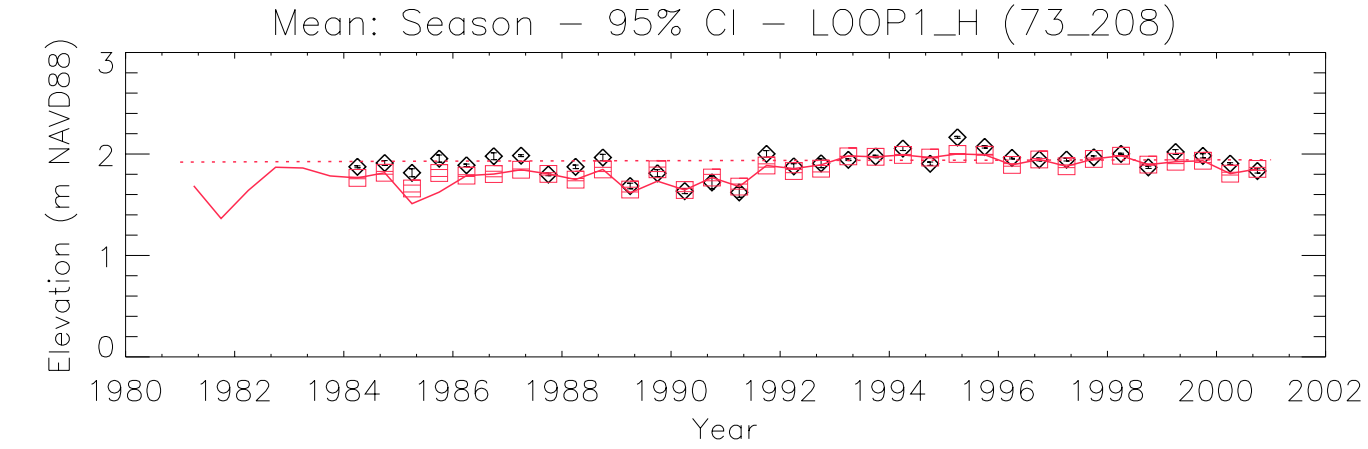

ELM2.8reg500 Raw Data (Obs. N = 7305) – L29 (129_208)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

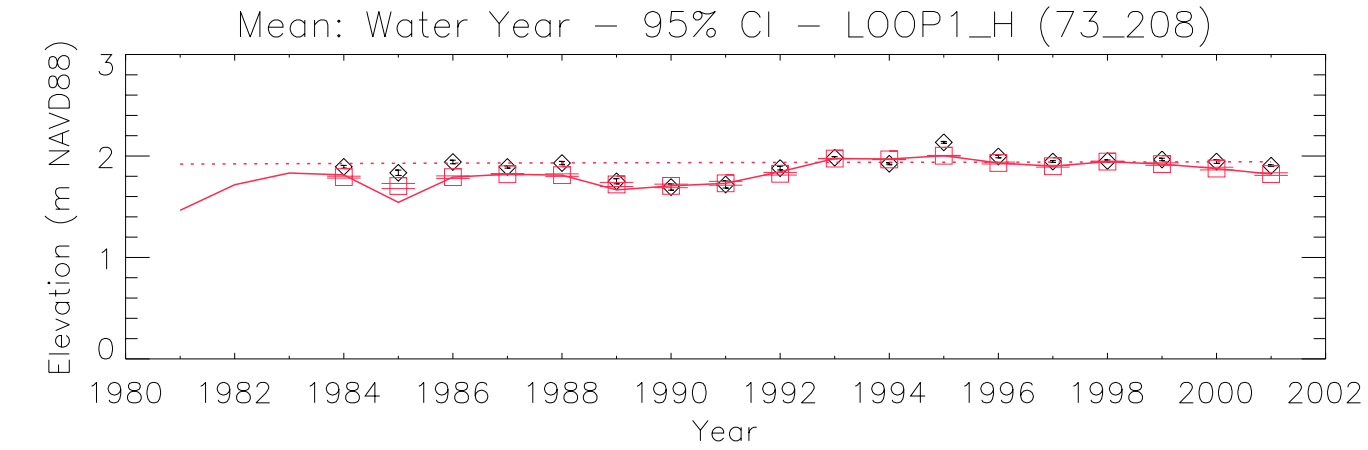

Mean: Season – 95% CI – L29 (129_208)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

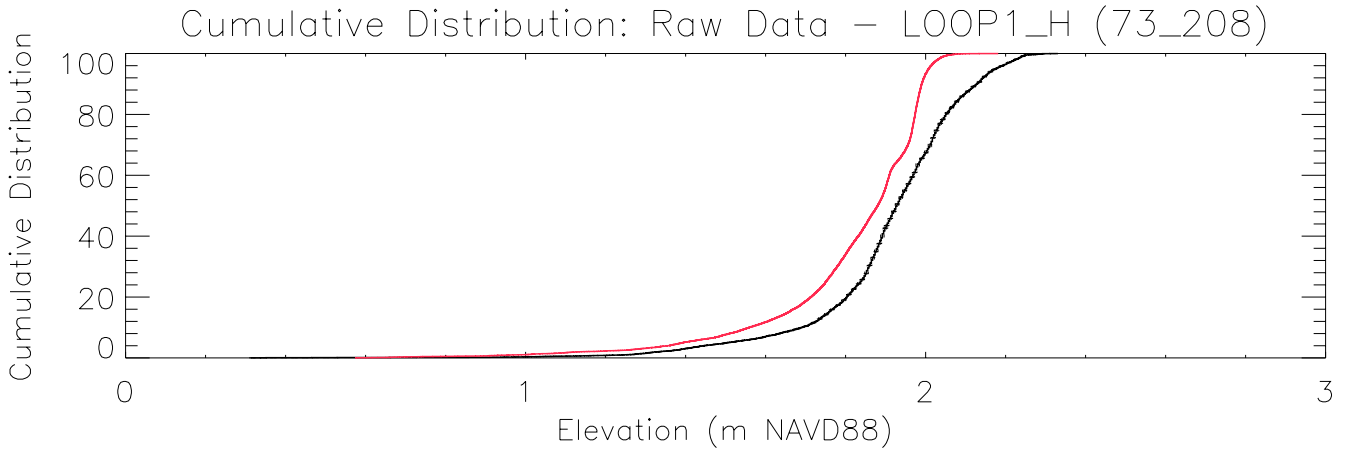

Mean: Water Year – 95% CI – L29 (129_208)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

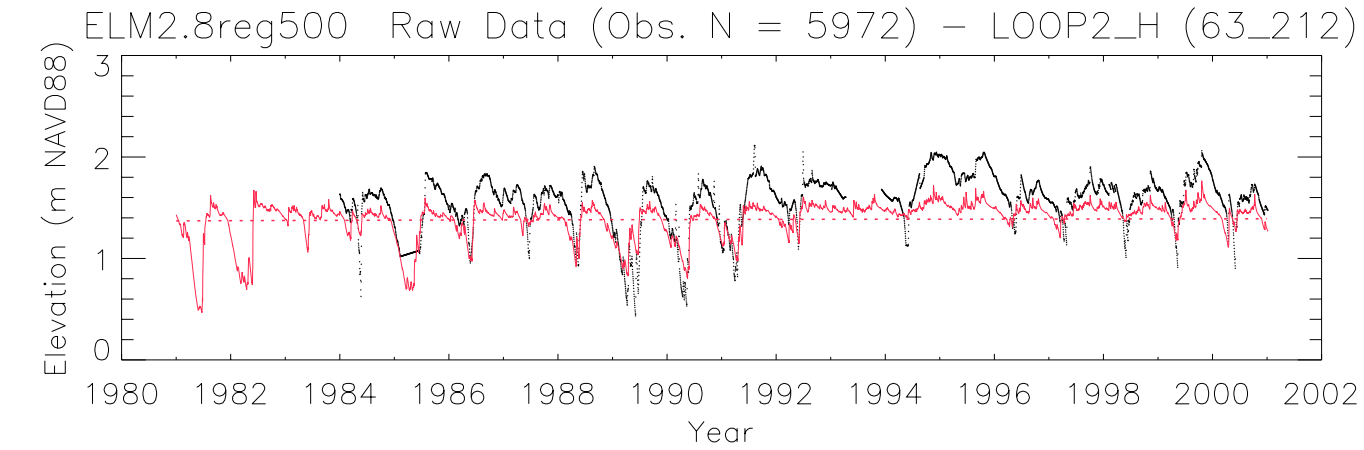

Cumulative Distribution: Raw Data – L29 (129_208)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

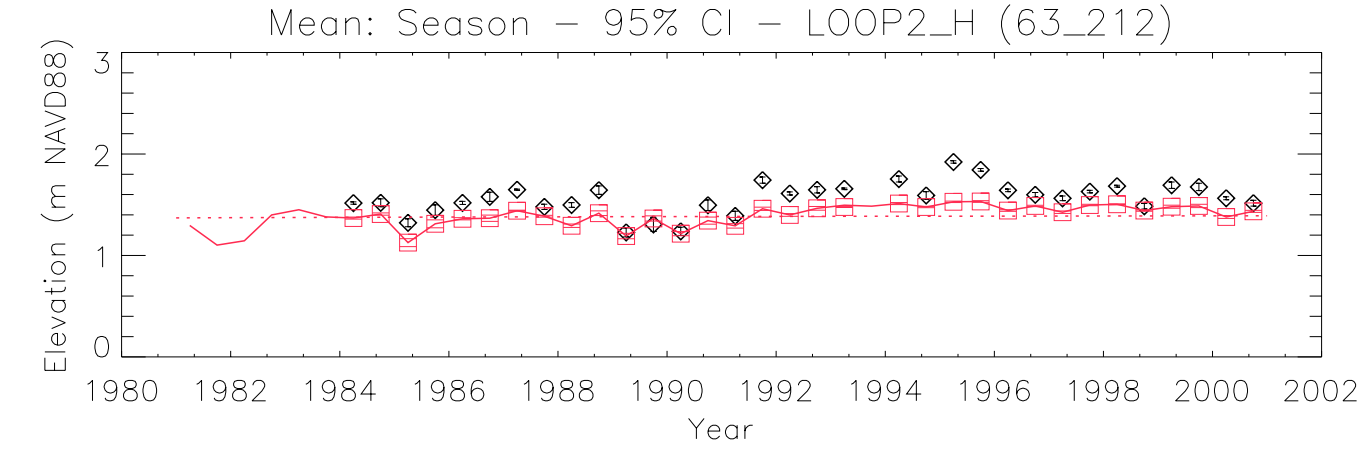

ELM2.8reg500 Raw Data (Obs. N = 5938) – LOOP1_H (73_208)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

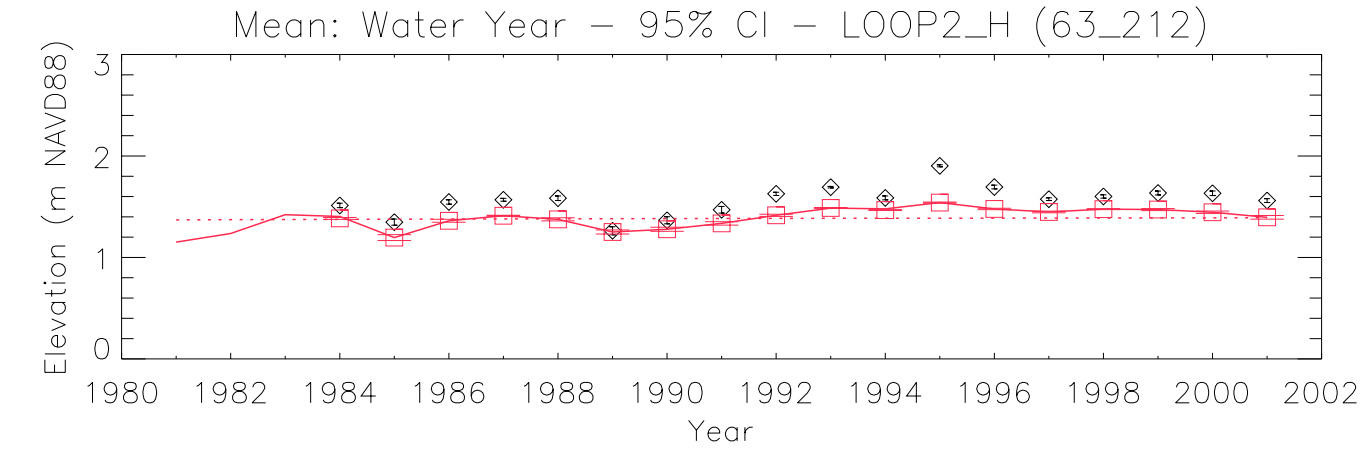

Mean: Season – 95% CI – LOOP1_H (73_208)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

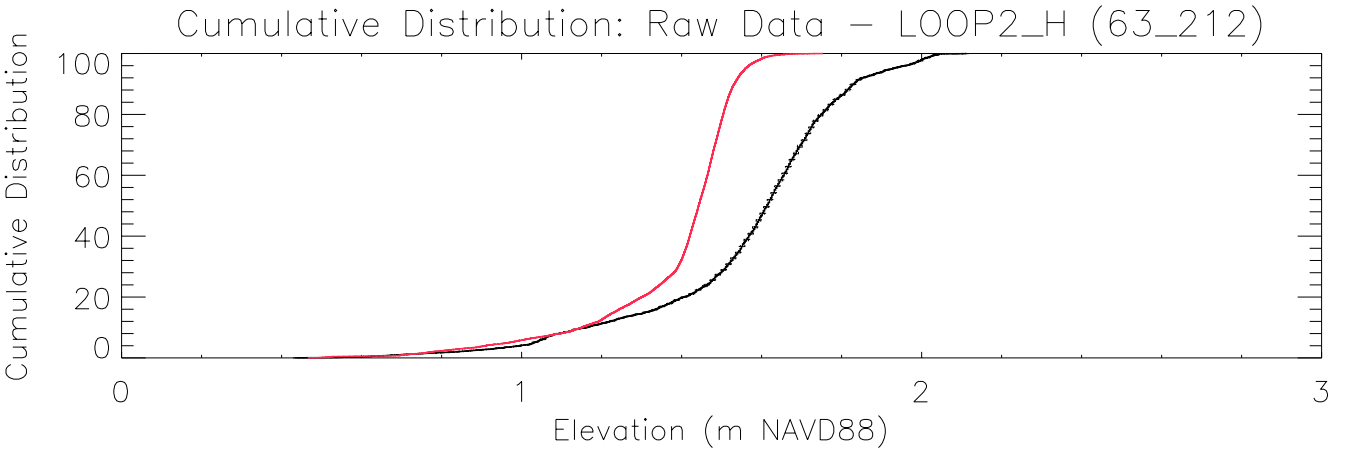

Mean: Water Year – 95% CI – LOOP1_H (73_208)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

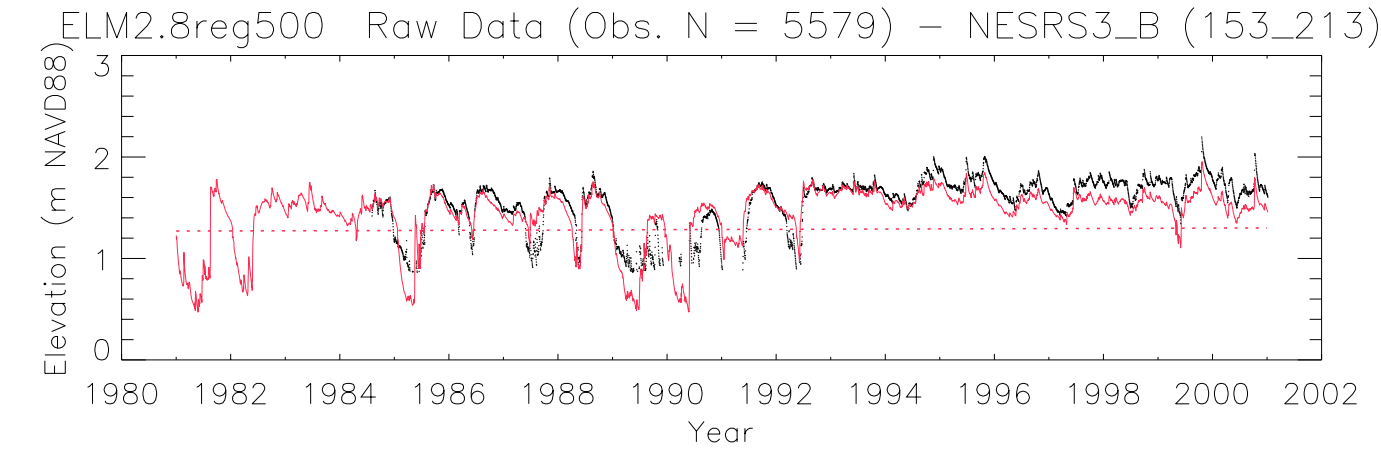

Cumulative Distribution: Raw Data – LOOP1_H (73_208)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

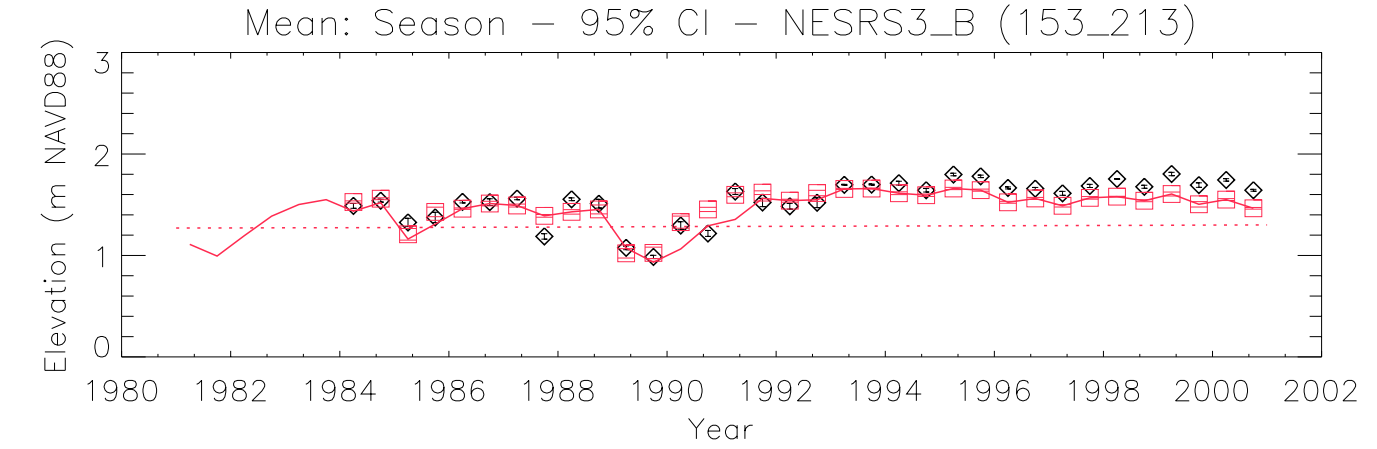

ELM2.8reg500 Raw Data (Obs. N = 5972) – LOOP2_H (63_212)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

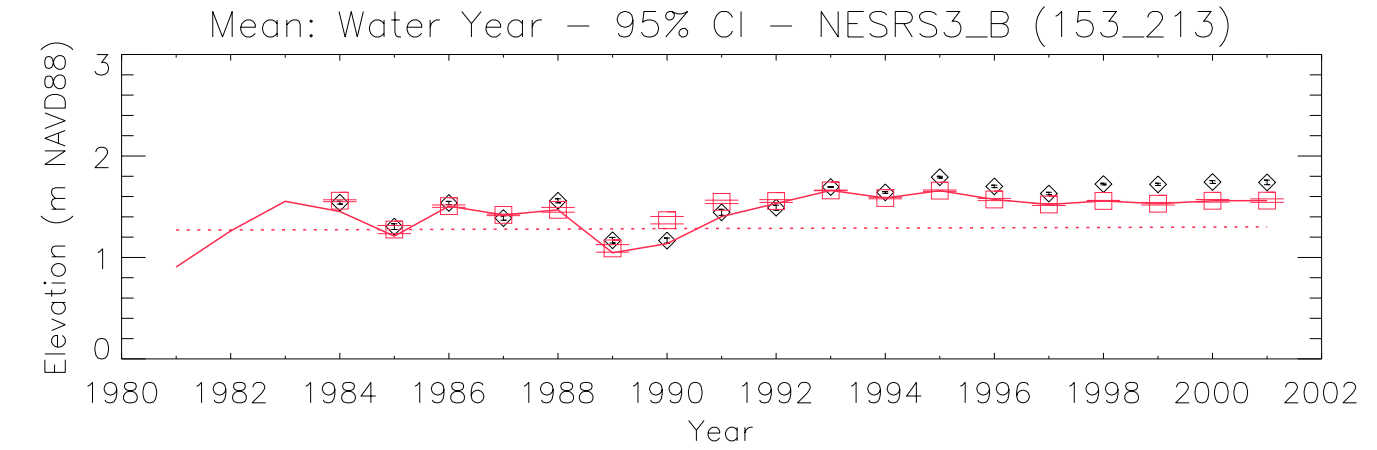

Mean: Season – 95% CI – LOOP2_H (63_212)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

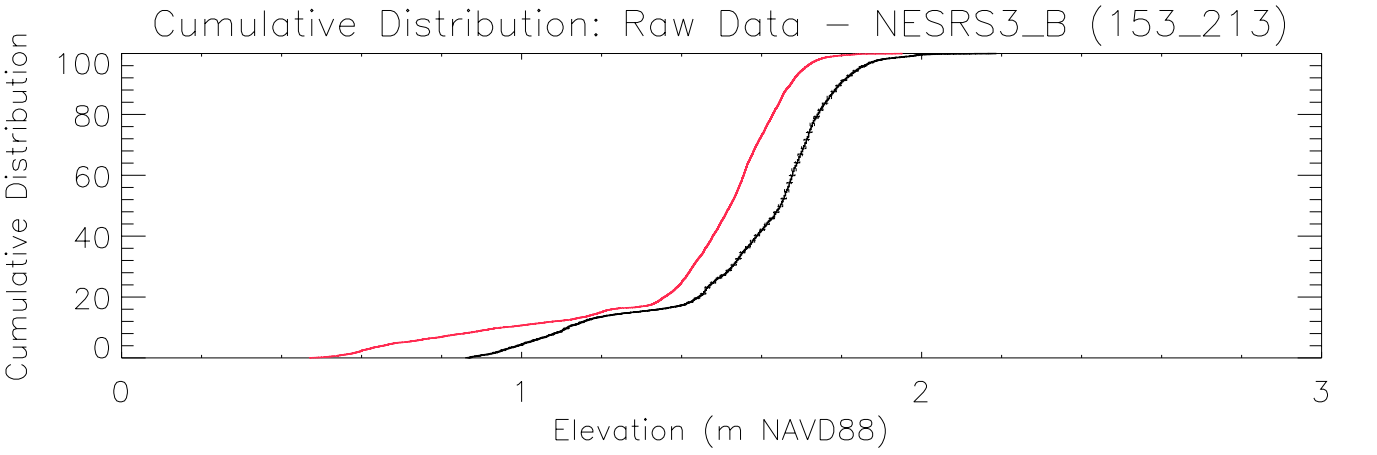

Mean: Water Year – 95% CI – LOOP2_H (63_212)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

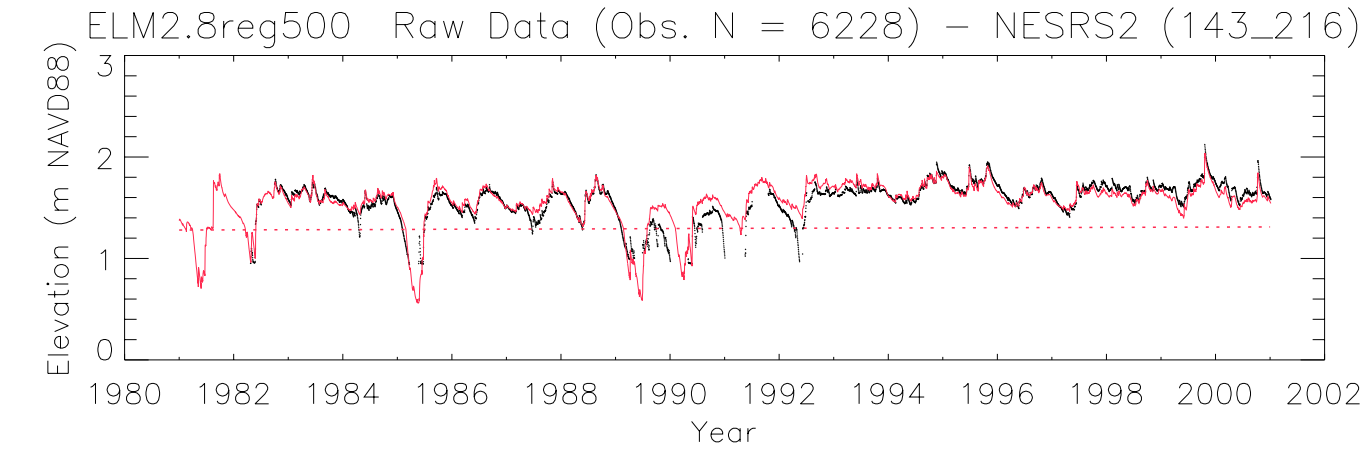

Cumulative Distribution: Raw Data – LOOP2_H (63_212)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

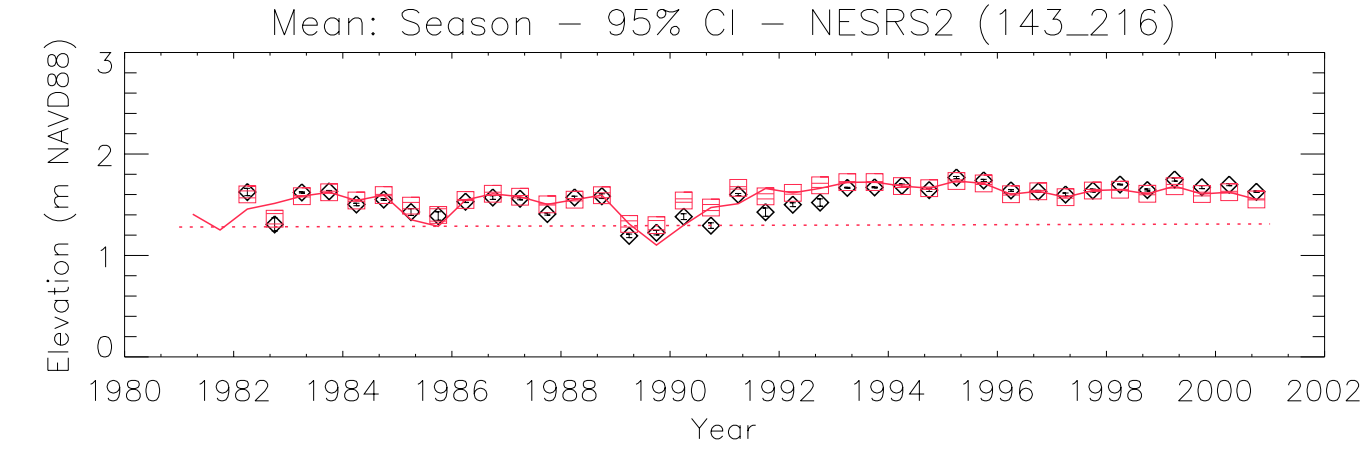

ELM2.8reg500 Raw Data (Obs. N = 5579) – NESRS3_B (153_213)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

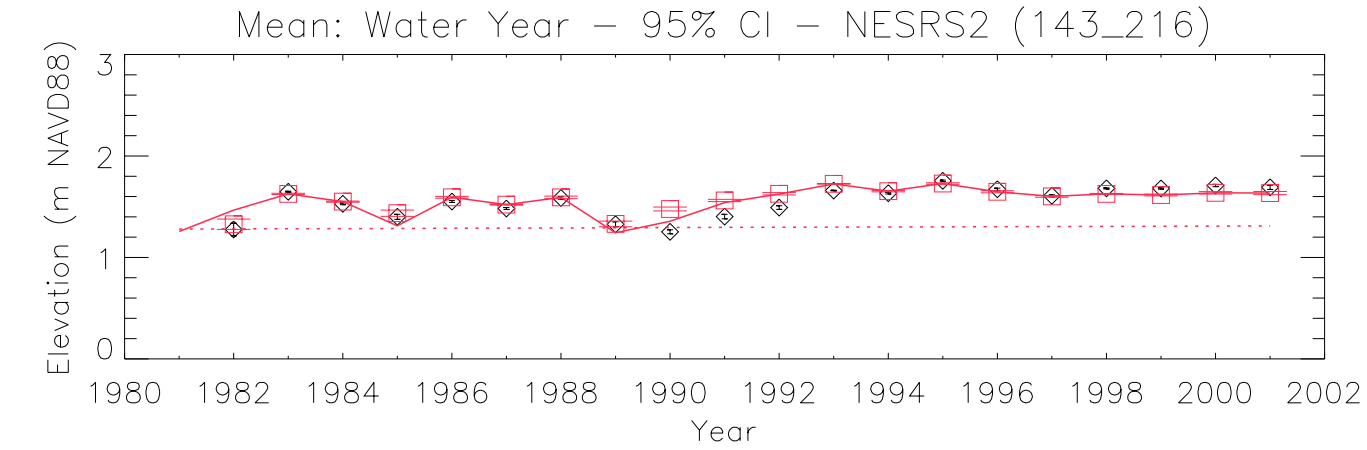

Mean: Season – 95% CI – NESRS3_B (153_213)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

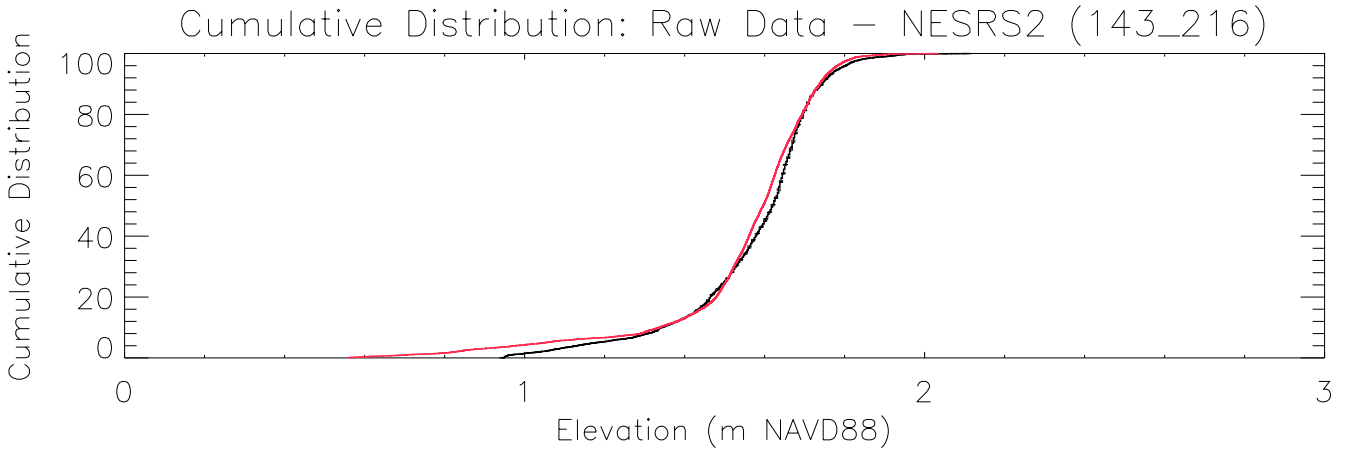

Mean: Water Year – 95% CI – NESRS3_B (153_213)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

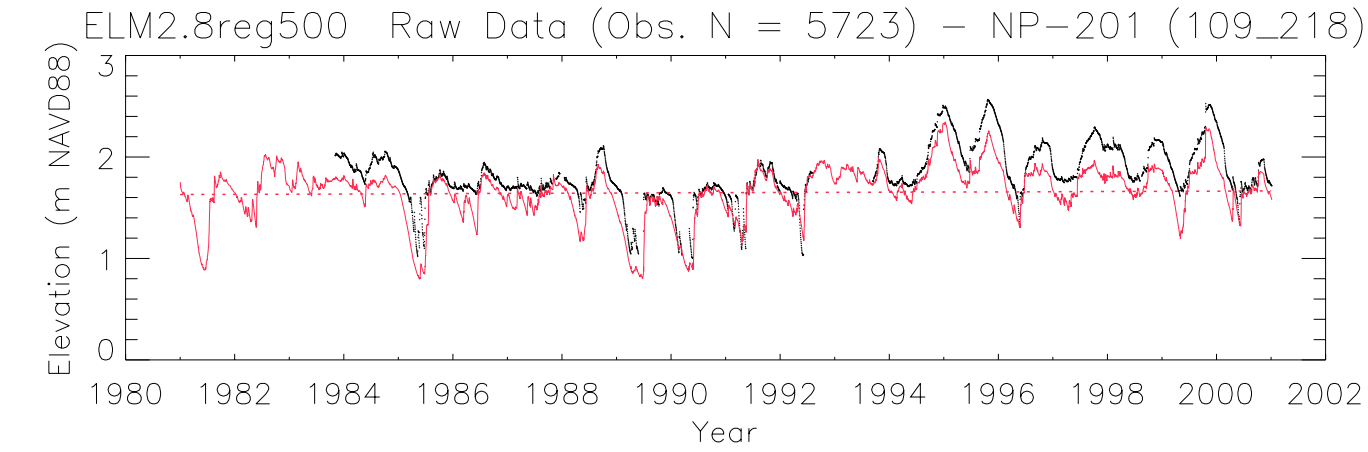

Cumulative Distribution: Raw Data – NESRS3_B (153_213)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

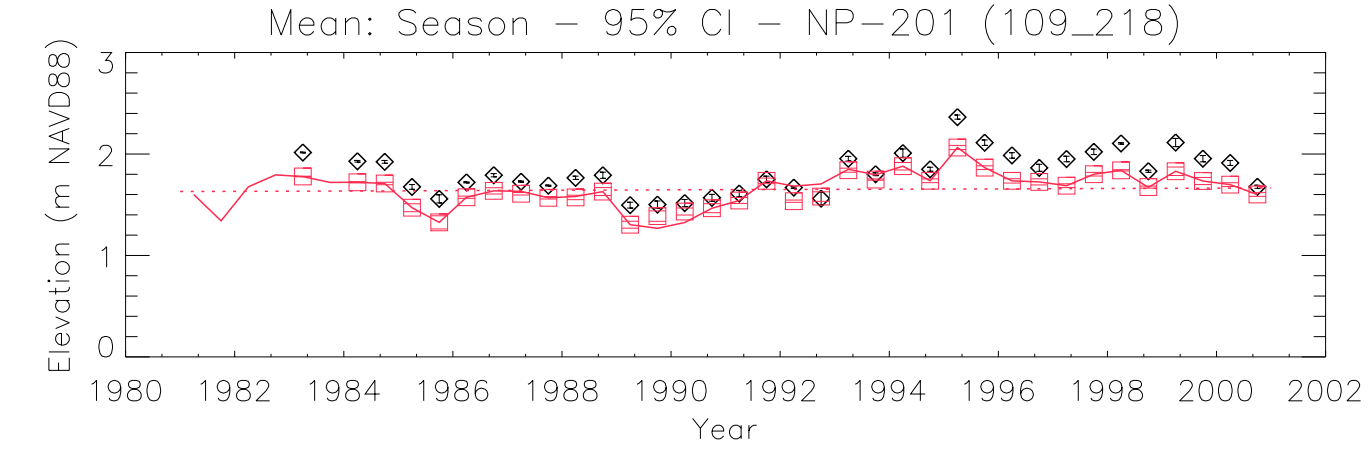

ELM2.8reg500 Raw Data (Obs. N = 6228) – NESRS2 (143_216)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

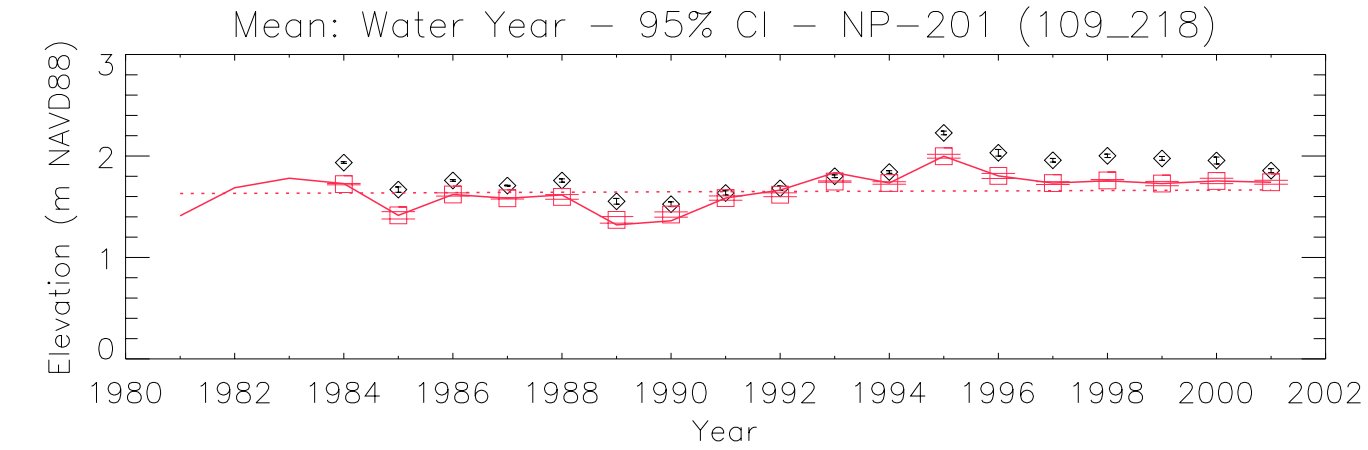

Mean: Season – 95% CI – NESRS2 (143_216)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

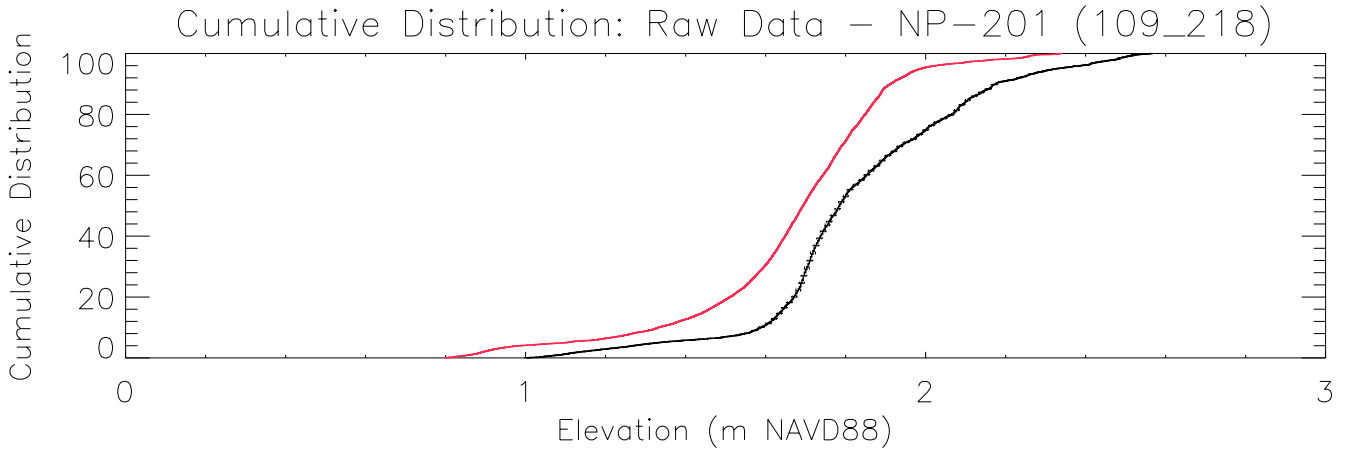

Mean: Water Year – 95% CI – NESRS2 (143_216)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

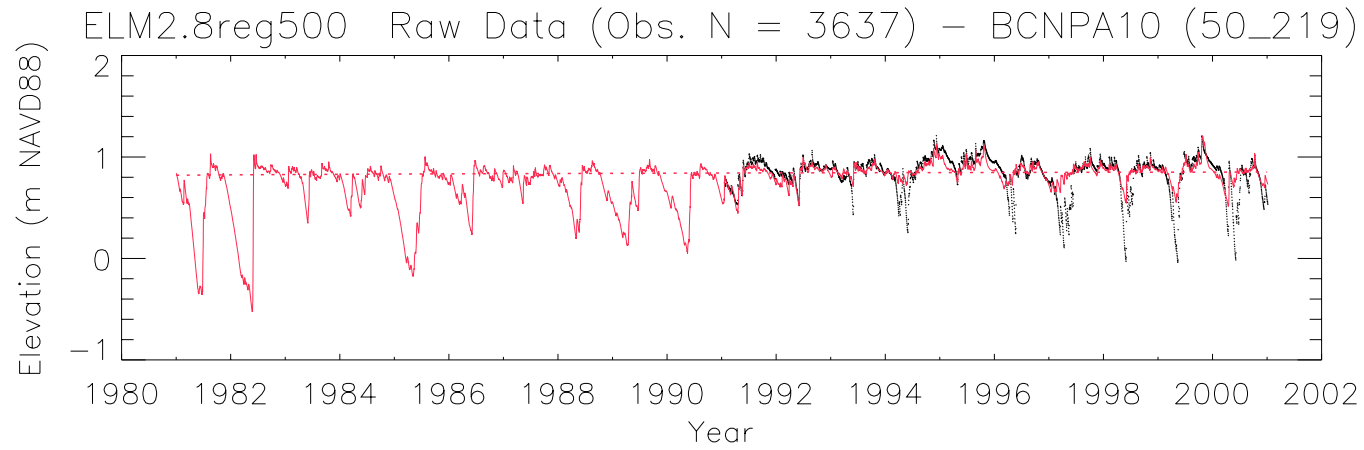

Cumulative Distribution: Raw Data – NESRS2 (143_216)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

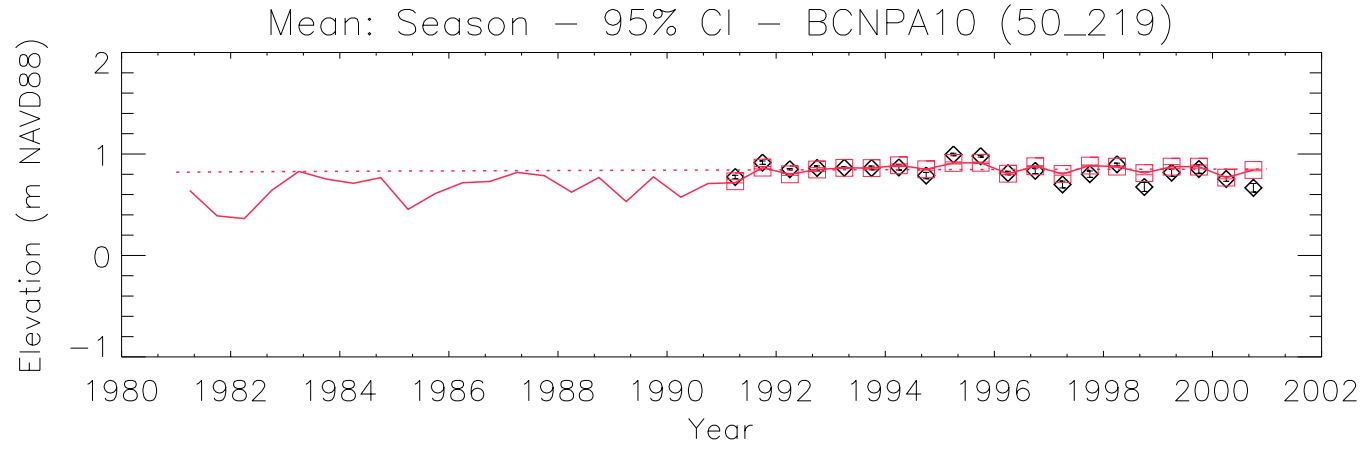

ELM2.8reg500 Raw Data (Obs. N = 5723) – NP–201 (109_218)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

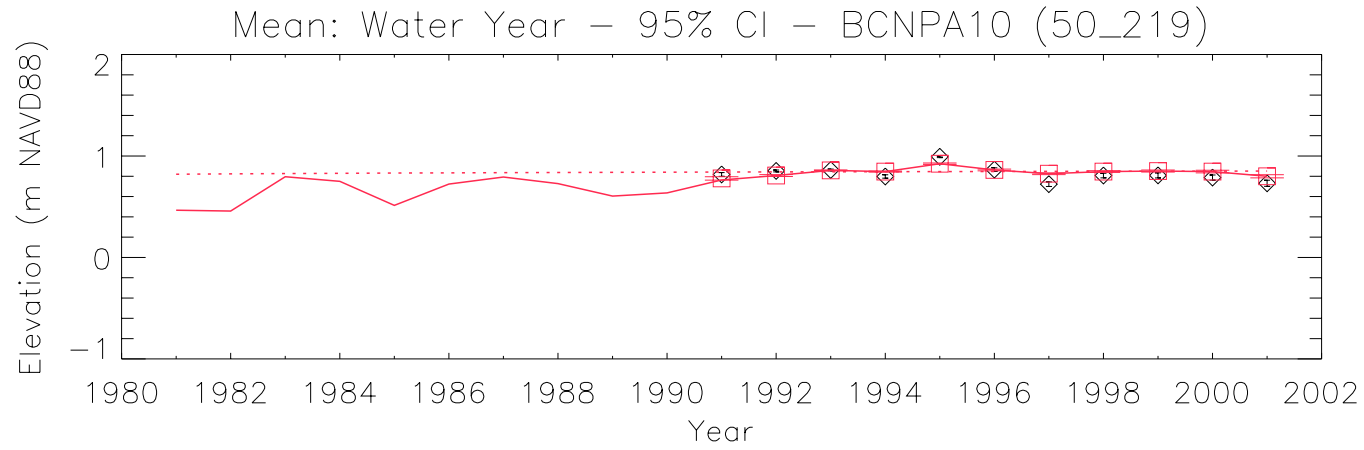

Mean: Season – 95% CI – NP–201 (109_218)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

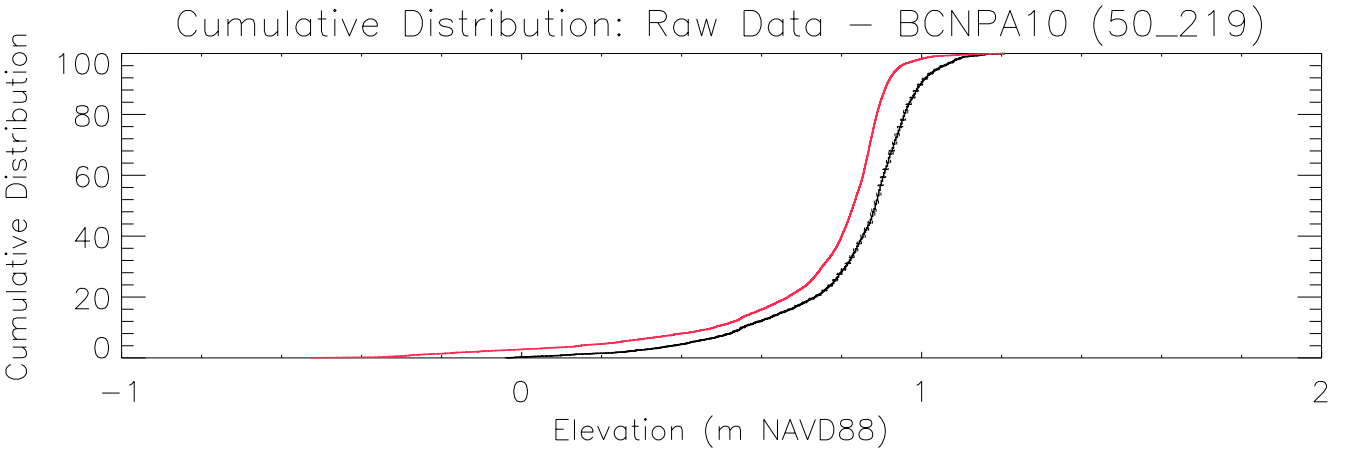

Mean: Water Year – 95% CI – NP–201 (109_218)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

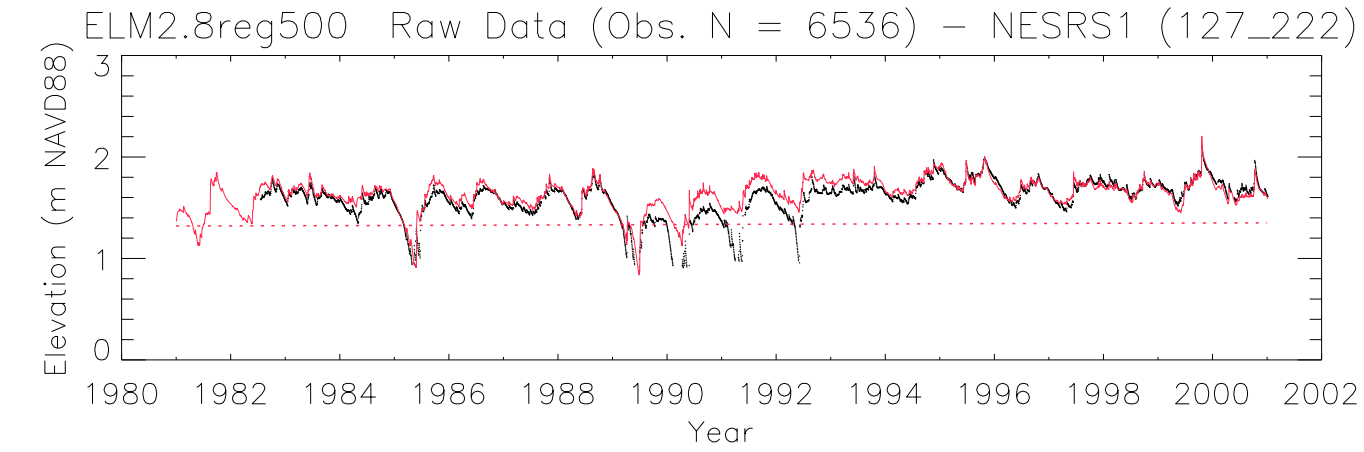

Cumulative Distribution: Raw Data – NP–201 (109_218)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

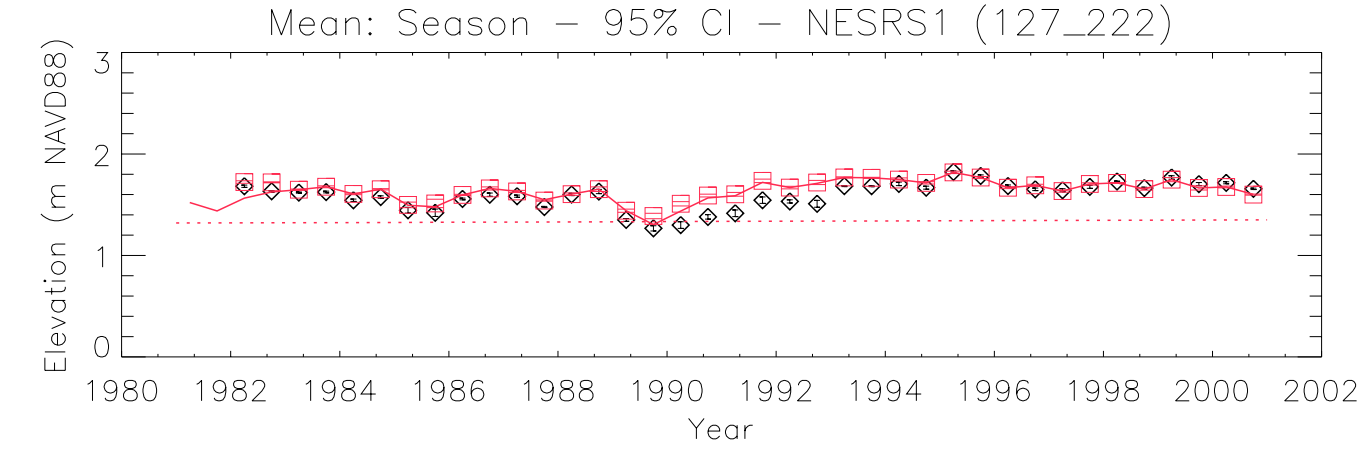

ELM2.8reg500 Raw Data (Obs. N = 3637) – BCNPA10 (50_219)
Elevation (m NAVD88)
2
1
0
−1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

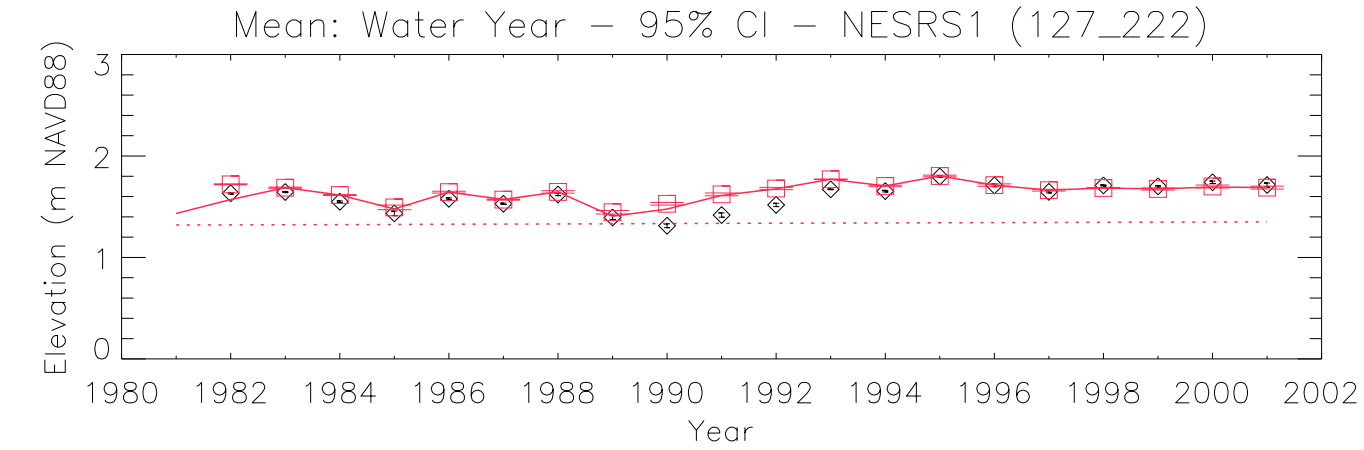

Mean: Season – 95% CI – BCNPA10 (50_219)
Elevation (m NAVD88)
2
1
0
−1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

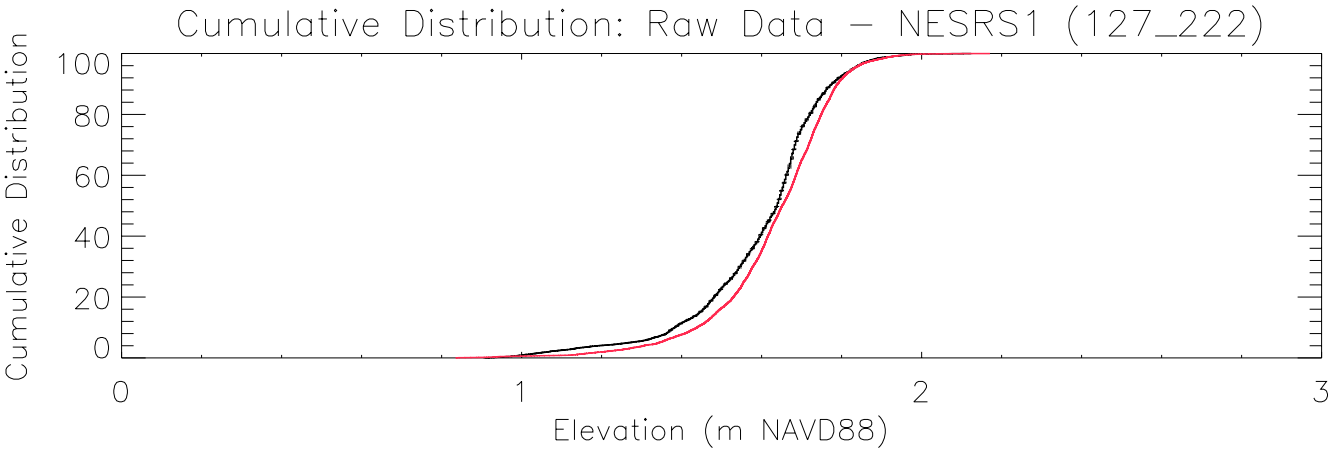

Mean: Water Year – 95% CI – BCNPA10 (50_219)
Elevation (m NAVD88)
2
1
0
−1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

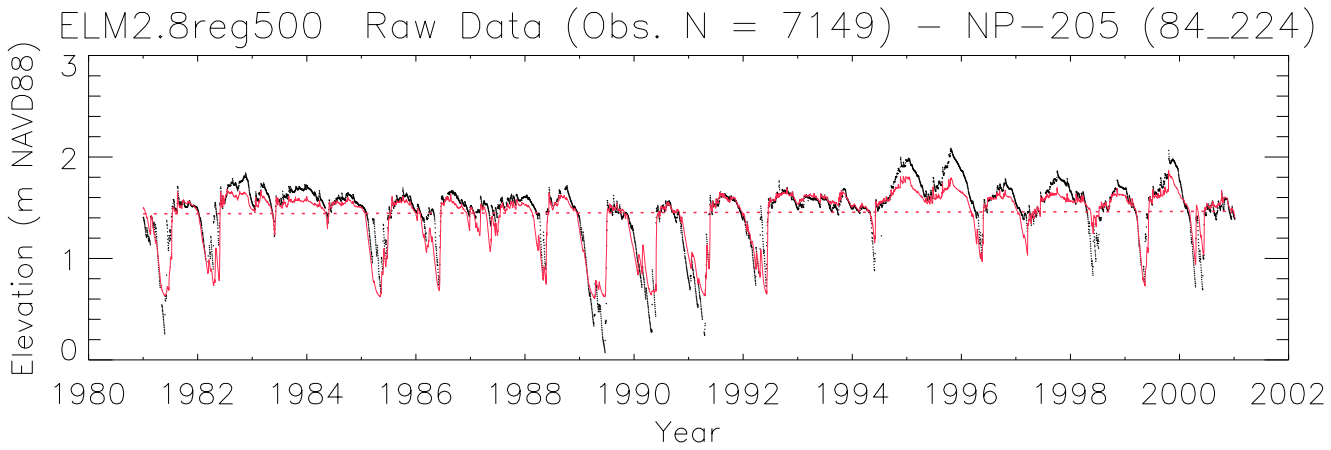

Cumulative Distribution: Raw Data – BCNPA10 (50_219)
Cumulative Distribution
100
80
60
40
20
0
−1
0
1
2
Elevation (m NAVD88)

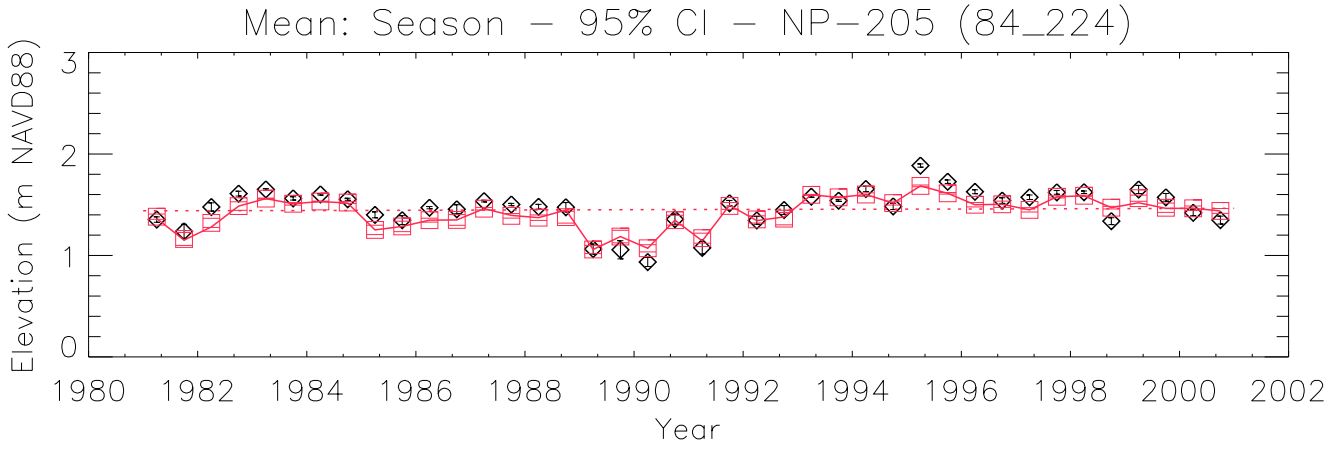

ELM2.8reg500 Raw Data (Obs. N = 6536) – NESRS1 (127_222)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

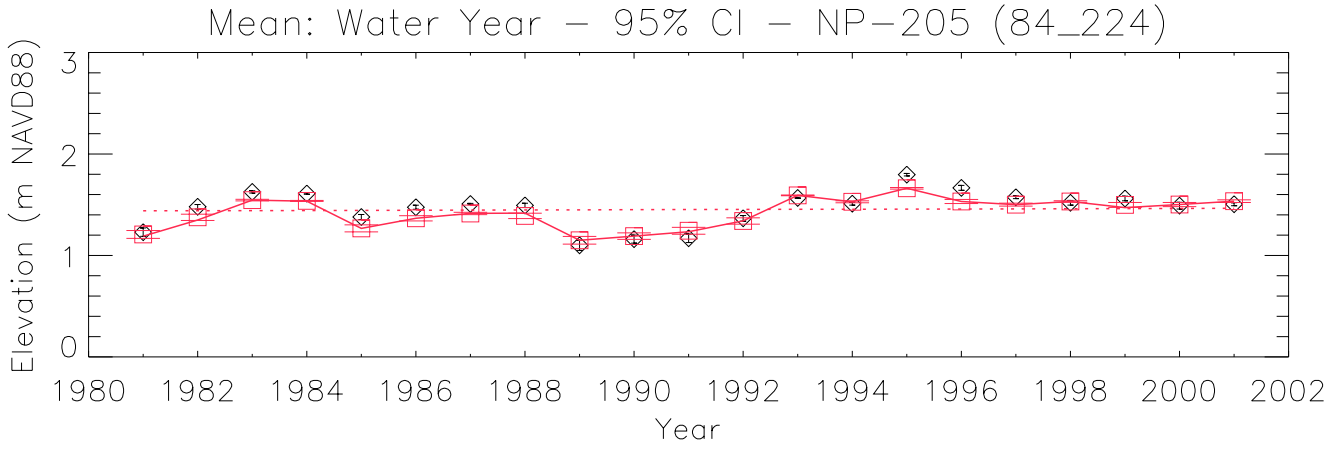

Mean: Season – 95% CI – NESRS1 (127_222)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

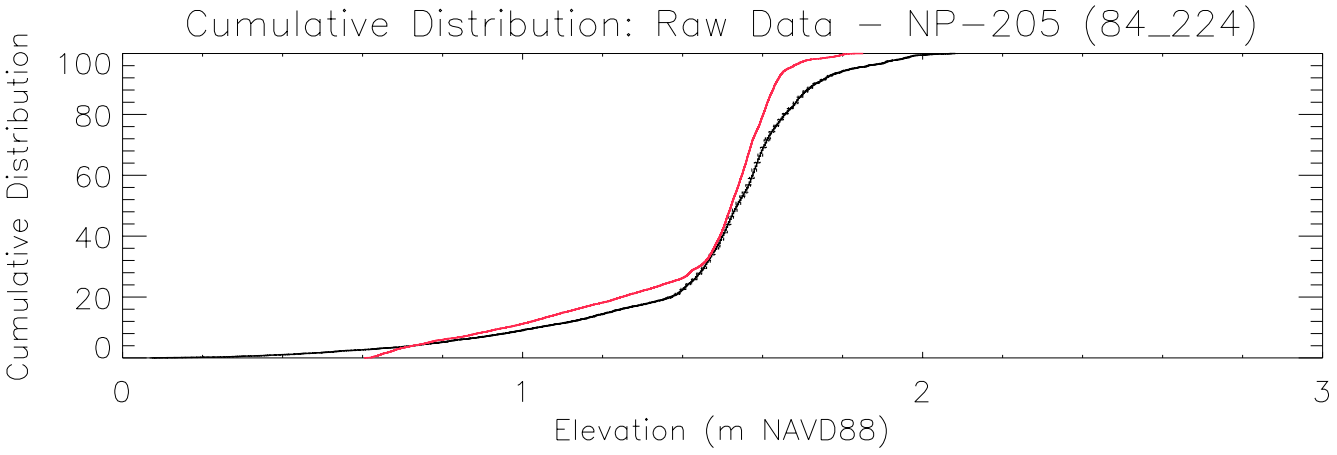

Mean: Water Year – 95% CI – NESRS1 (127_222)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

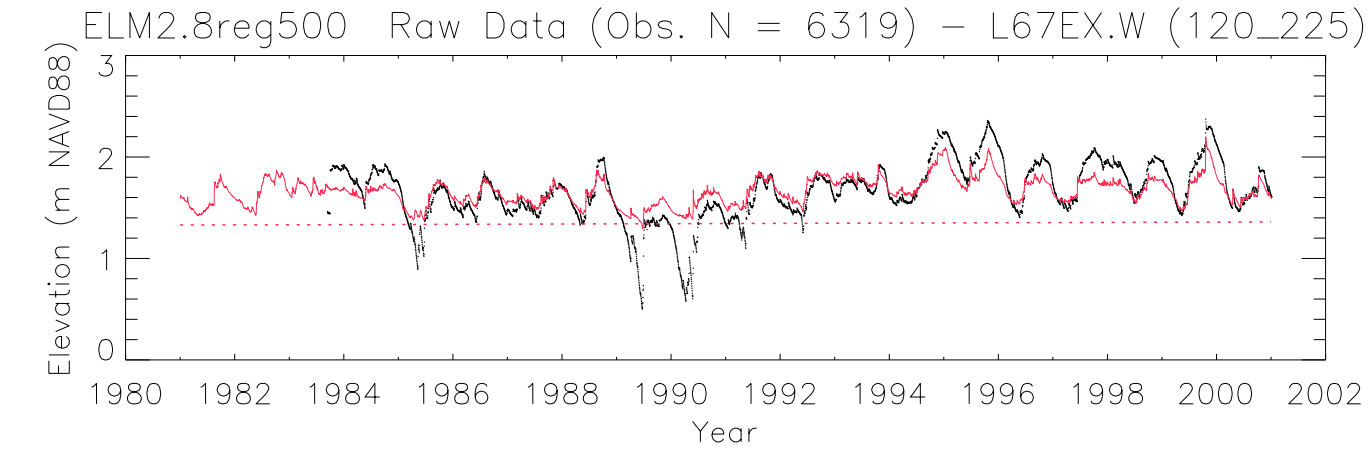

Cumulative Distribution: Raw Data – NESRS1 (127_222)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

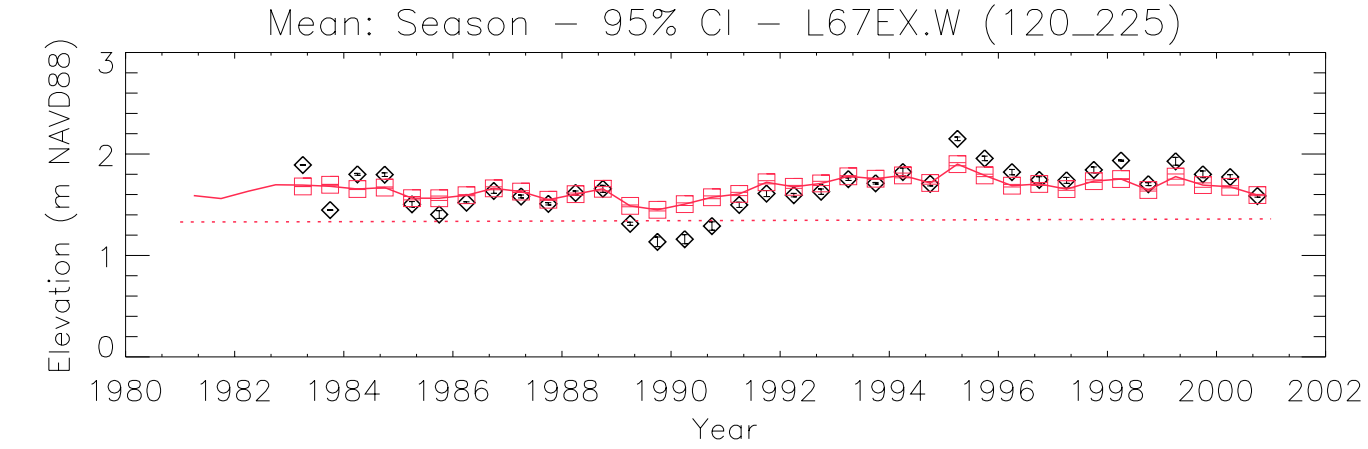

ELM2.8reg500 Raw Data (Obs. N = 7149) – NP–205 (84_224)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

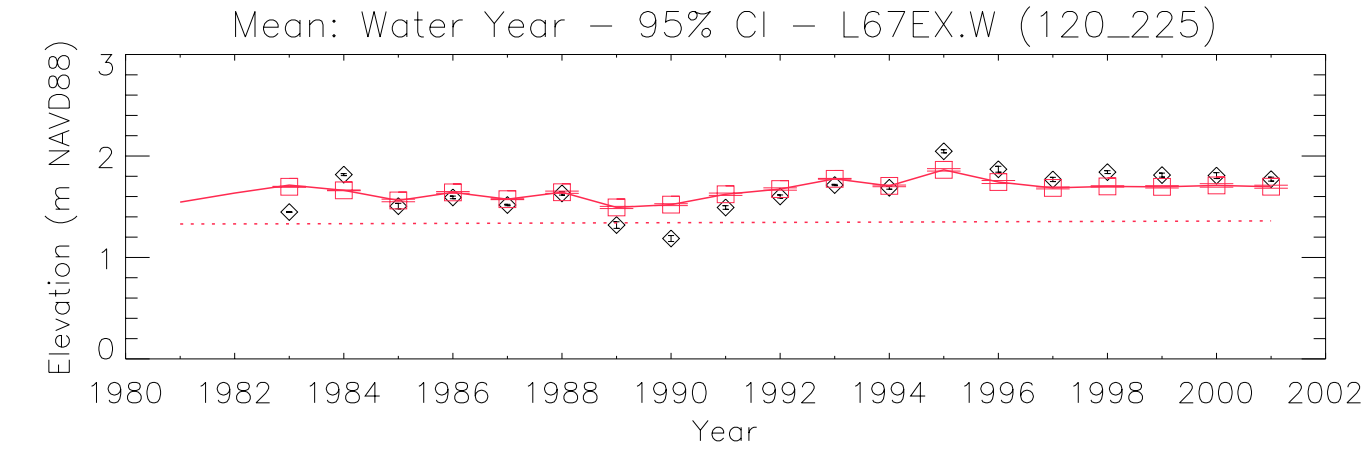

Mean: Season – 95% CI – NP–205 (84_224)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

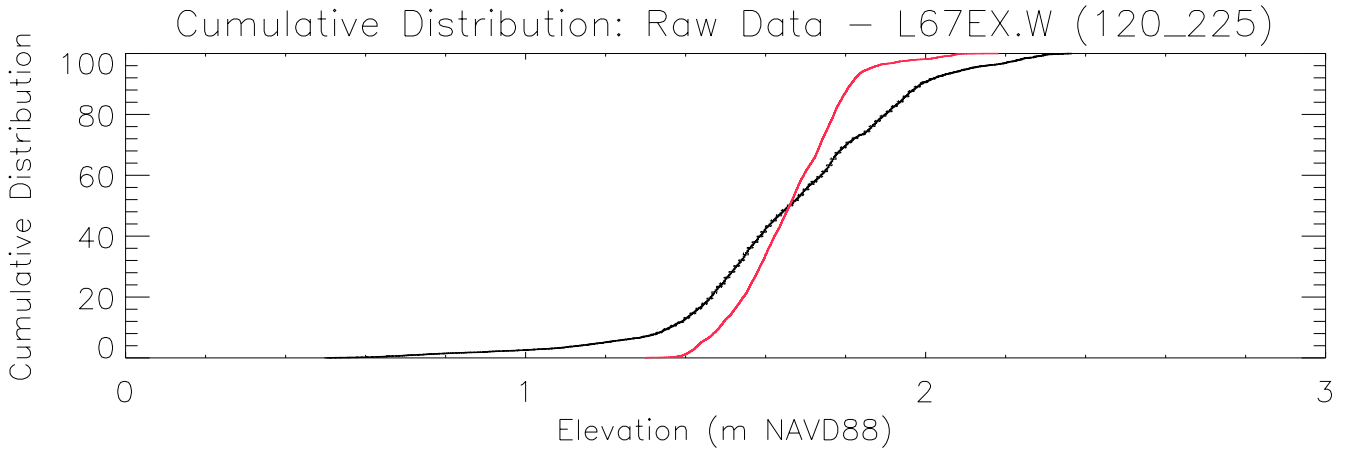

Mean: Water Year – 95% CI – NP–205 (84_224)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

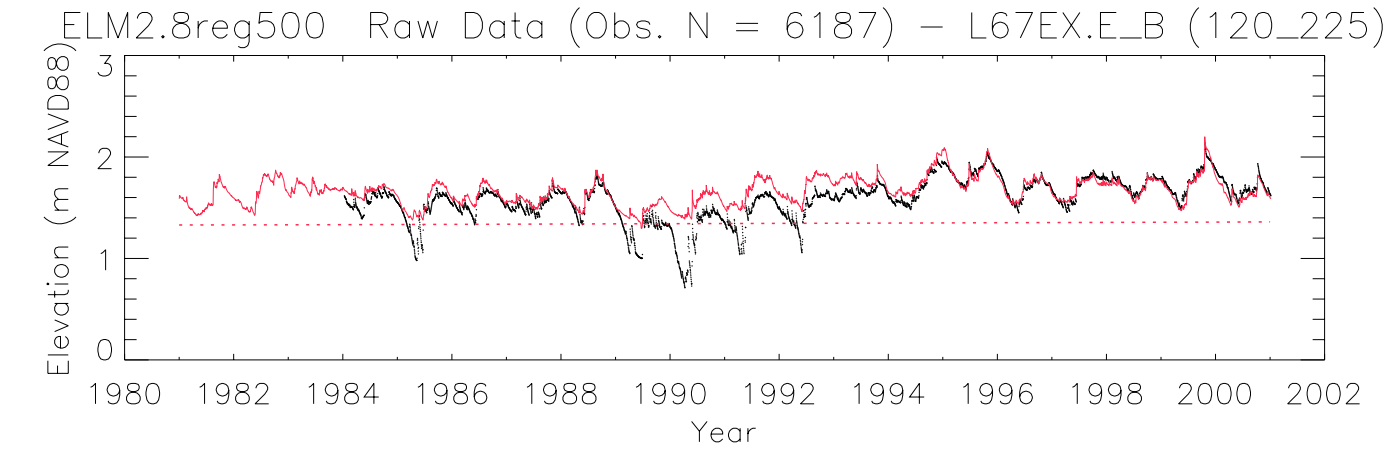

Cumulative Distribution: Raw Data – NP–205 (84_224)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

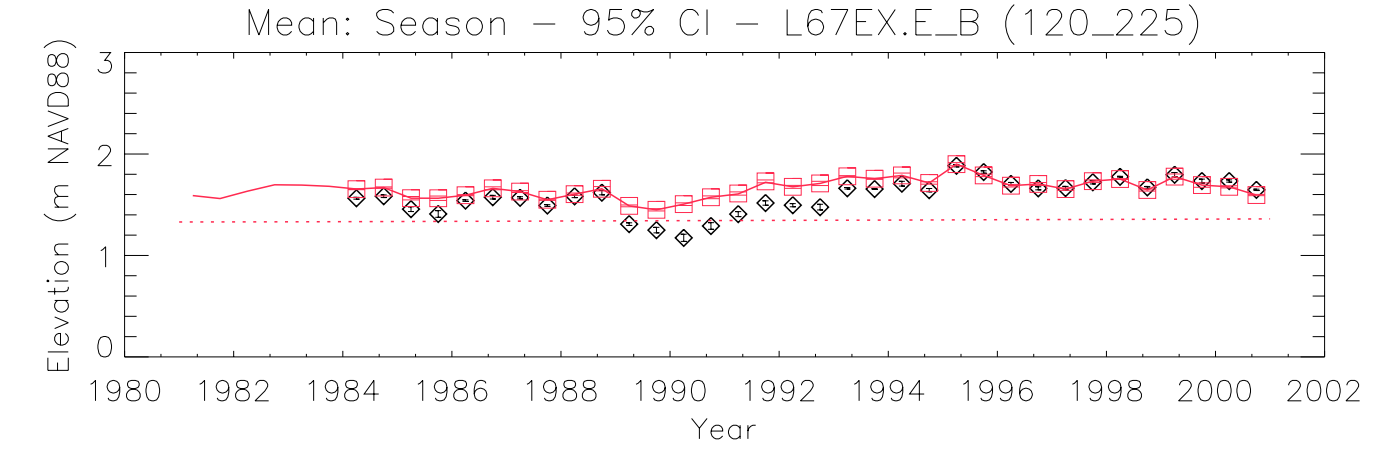

ELM2.8reg500 Raw Data (Obs. N = 6319) – L67EX.W (120_225)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

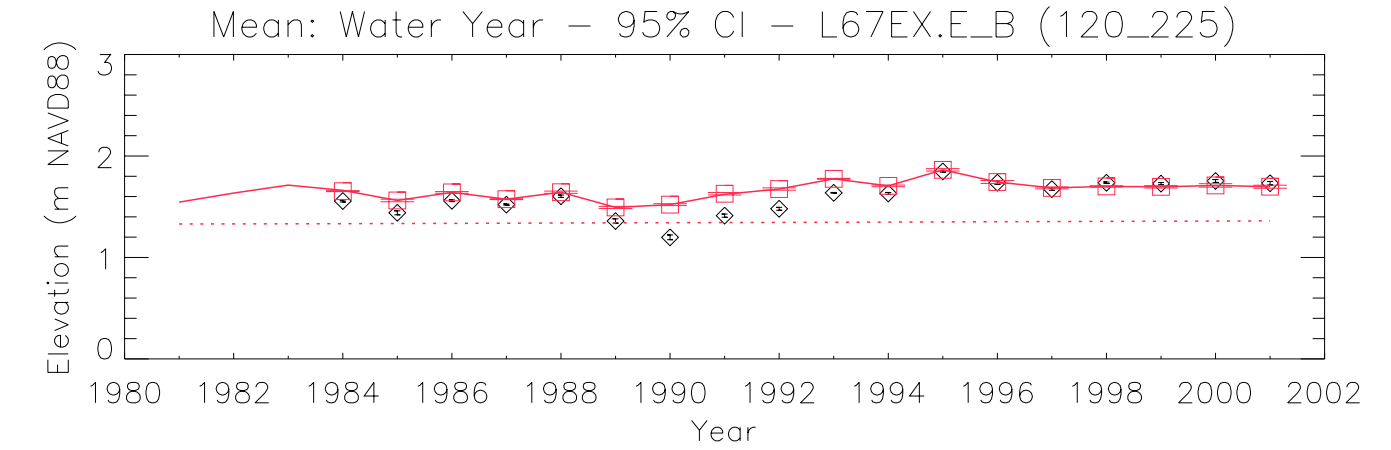

Mean: Season – 95% CI – L67EX.W (120_225)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

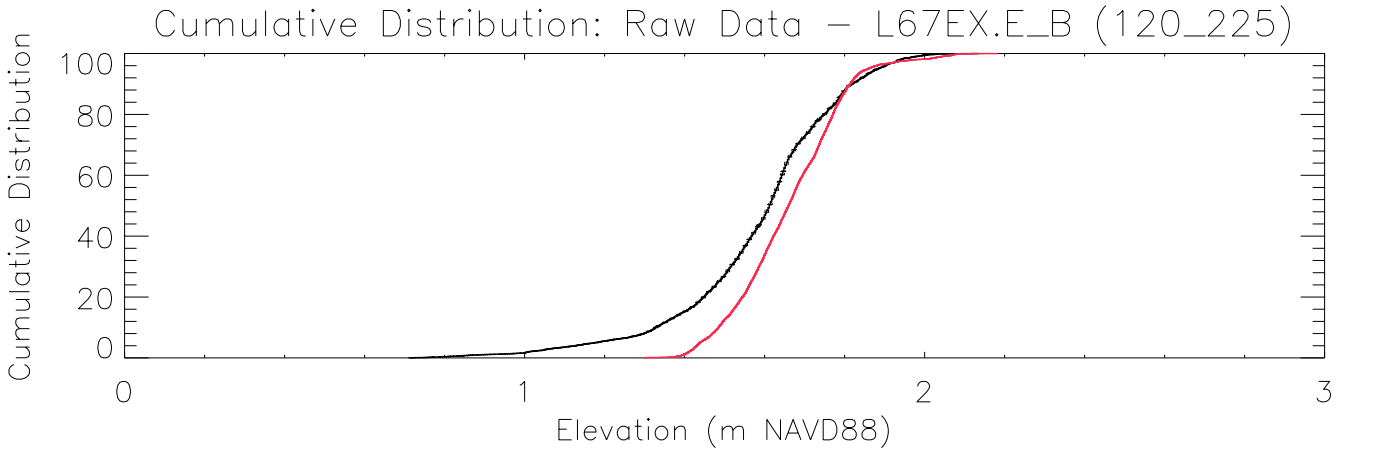

Mean: Water Year – 95% CI – L67EX.W (120_225)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

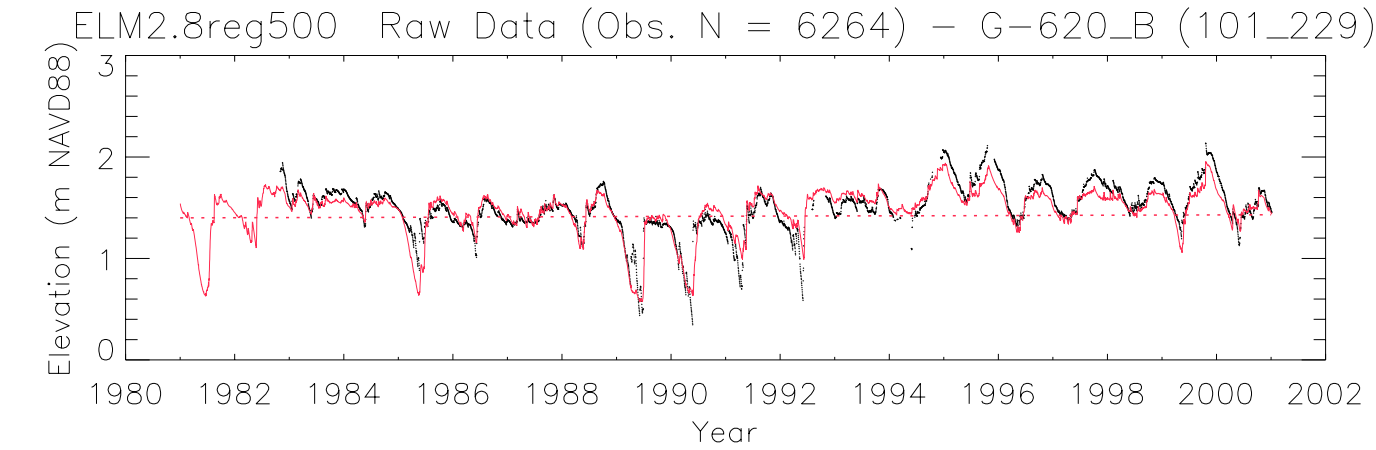

Cumulative Distribution: Raw Data – L67EX.W (120_225)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

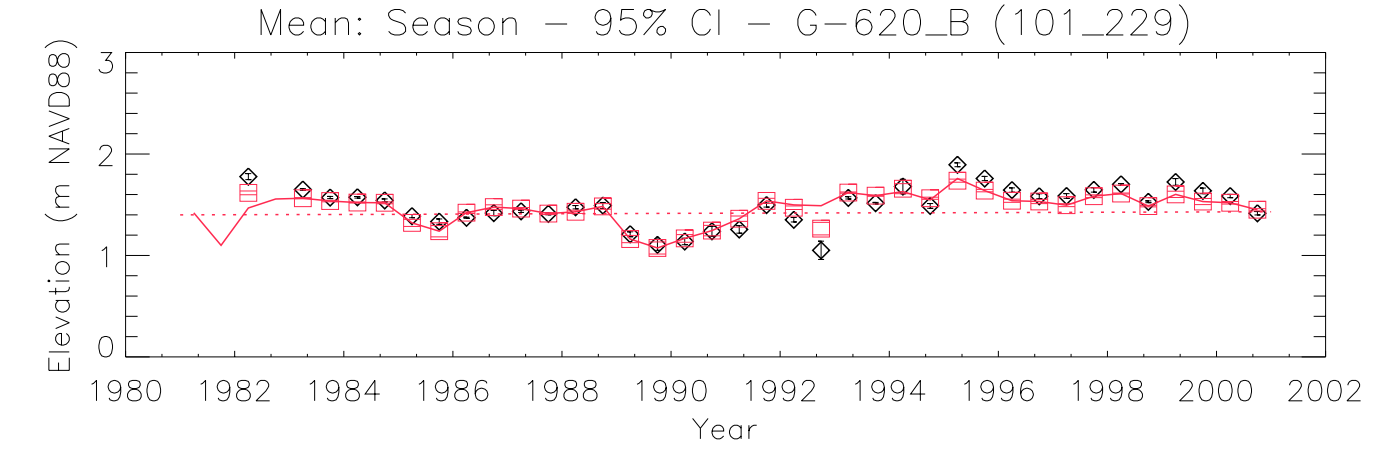

ELM2.8reg500 Raw Data (Obs. N = 6187) – L67EX.E_B (120_225)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

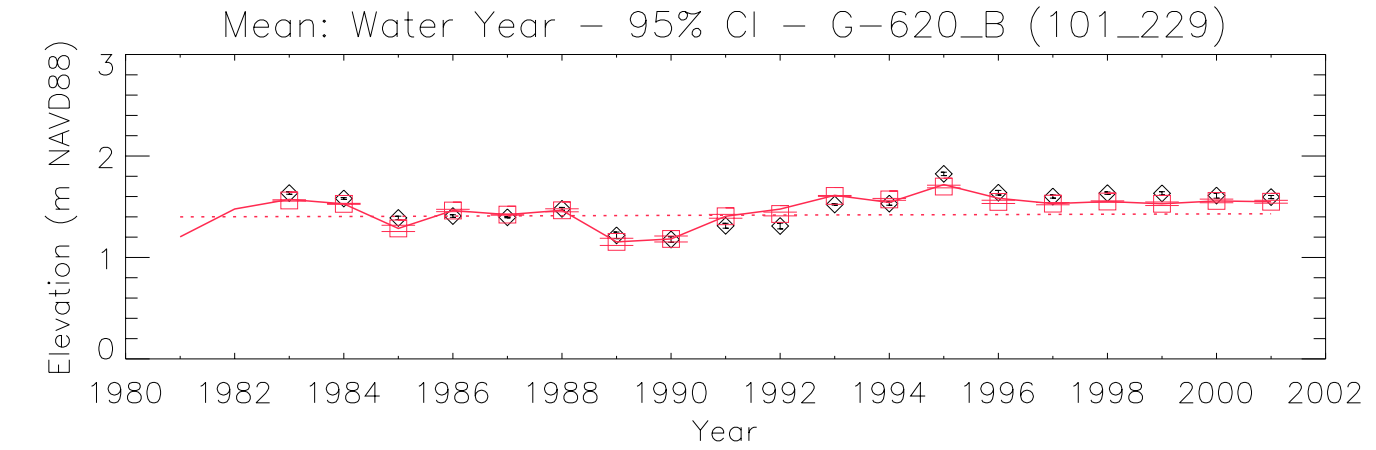

Mean: Season – 95% CI – L67EX.E_B (120_225)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

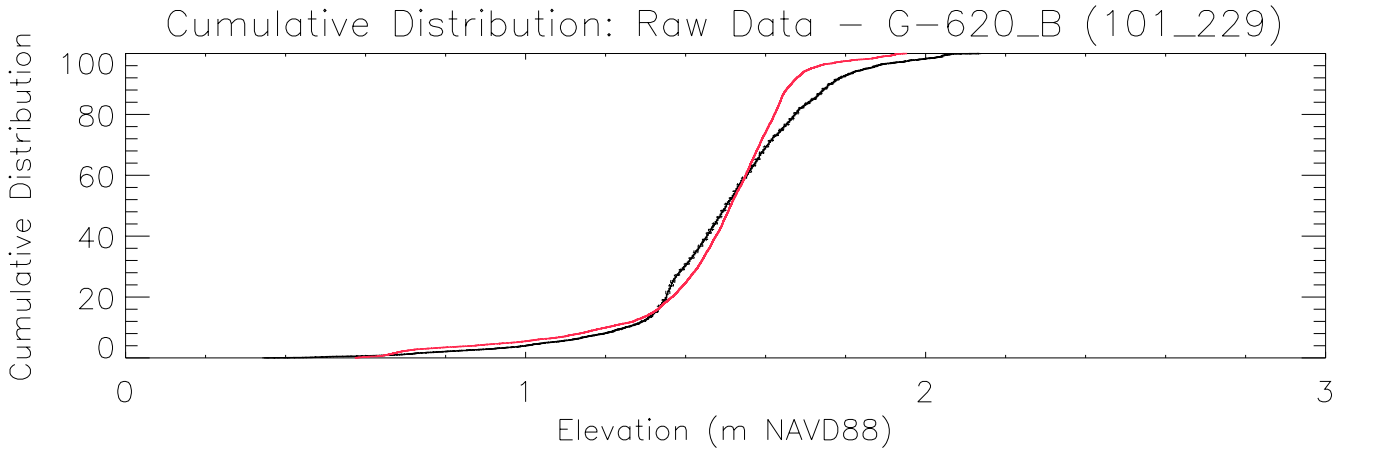

Mean: Water Year – 95% CI – L67EX.E_B (120_225)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

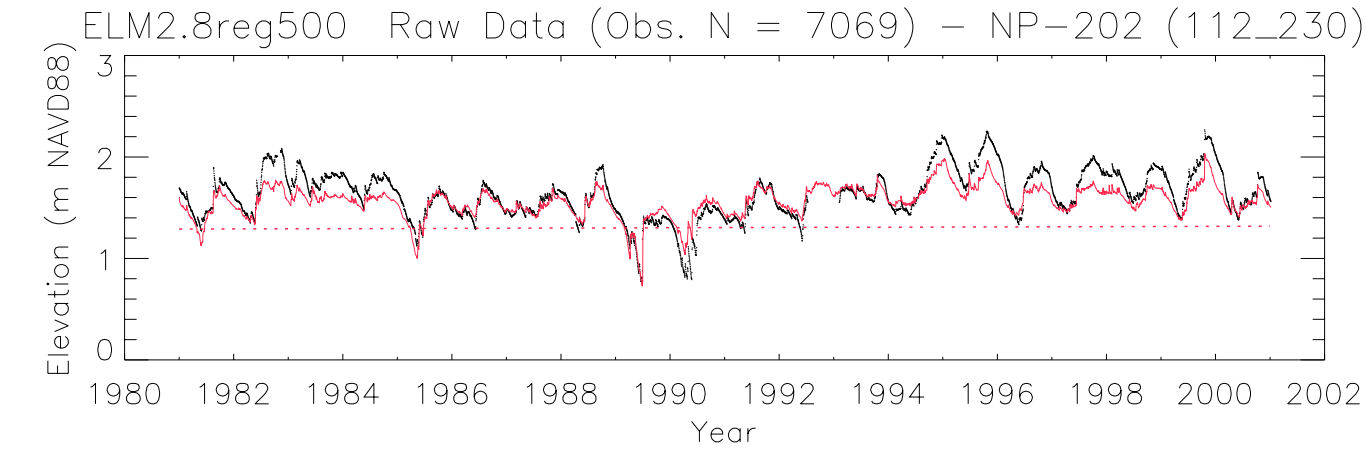

Cumulative Distribution: Raw Data – L67EX.E_B (120_225)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

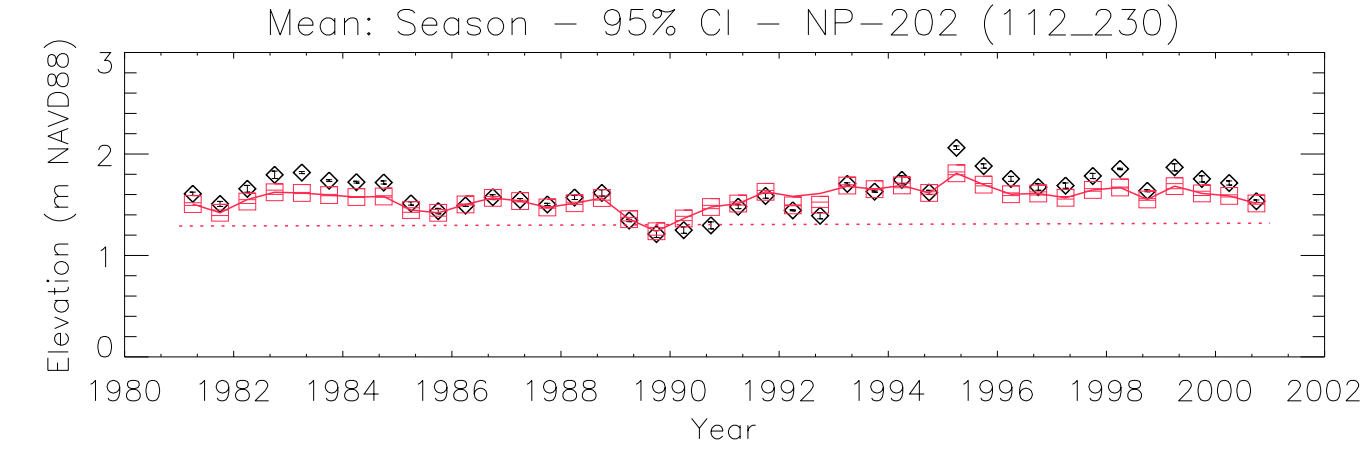

ELM2.8reg500 Raw Data (Obs. N = 6264) – G–620_B (101_229)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

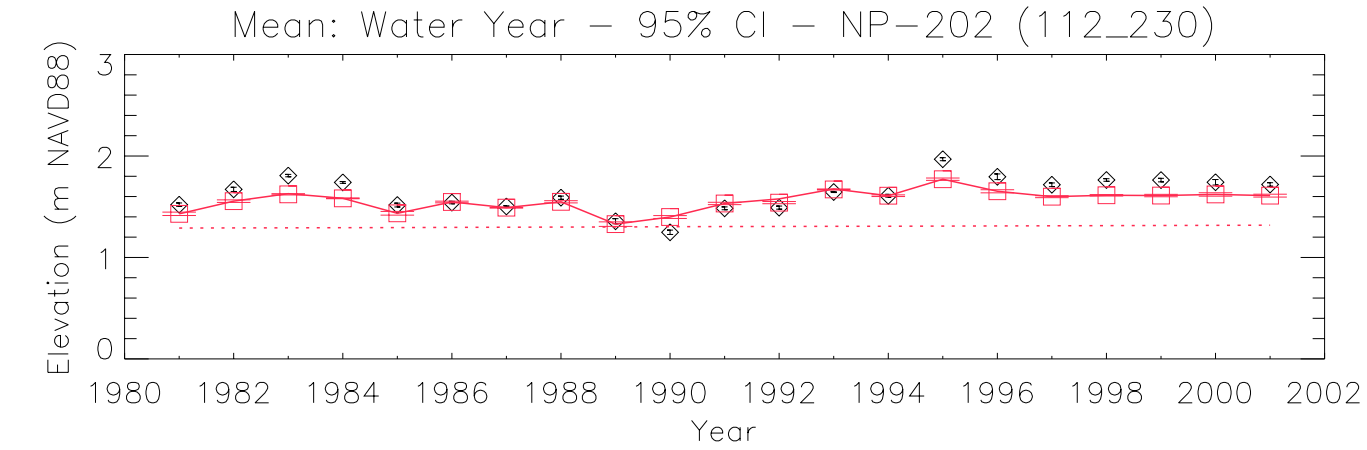

Mean: Season – 95% CI – G–620_B (101_229)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

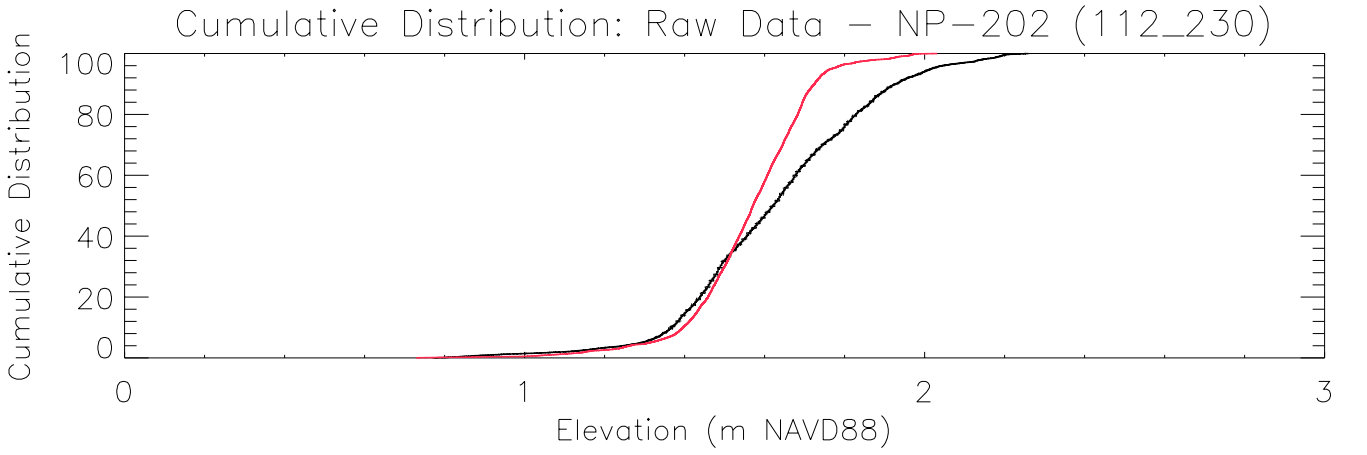

Mean: Water Year – 95% CI – G–620_B (101_229)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

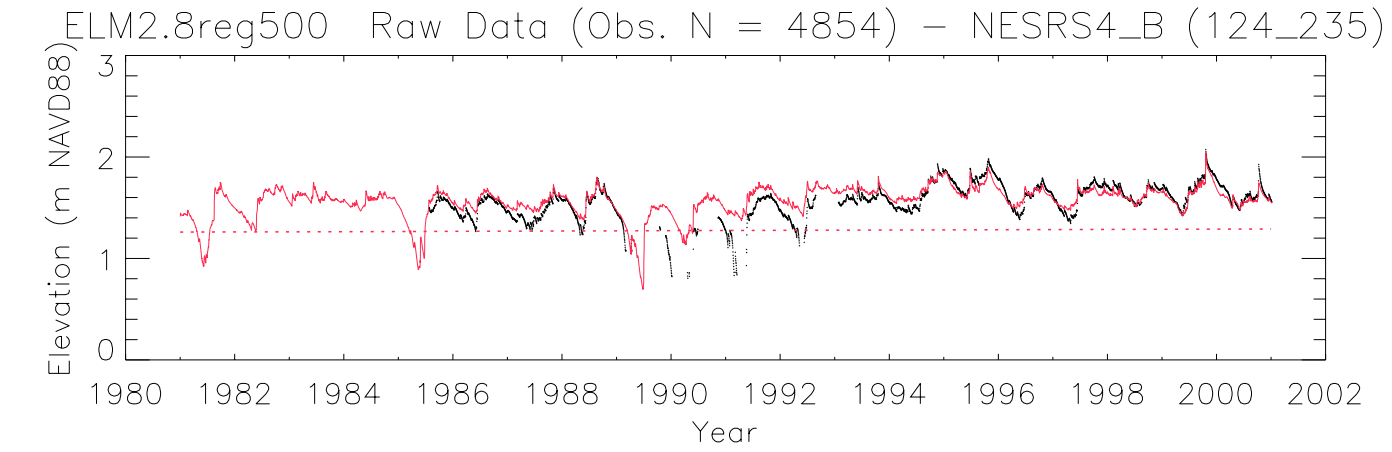

Cumulative Distribution: Raw Data – G–620_B (101_229)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

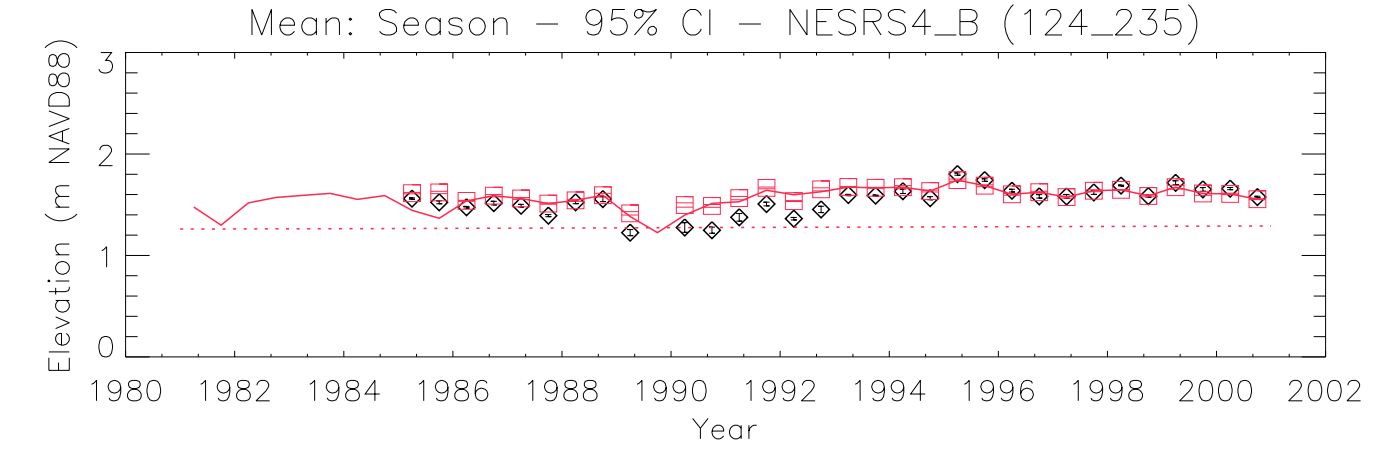

ELM2.8reg500 Raw Data (Obs. N = 7069) – NP–202 (112_230)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

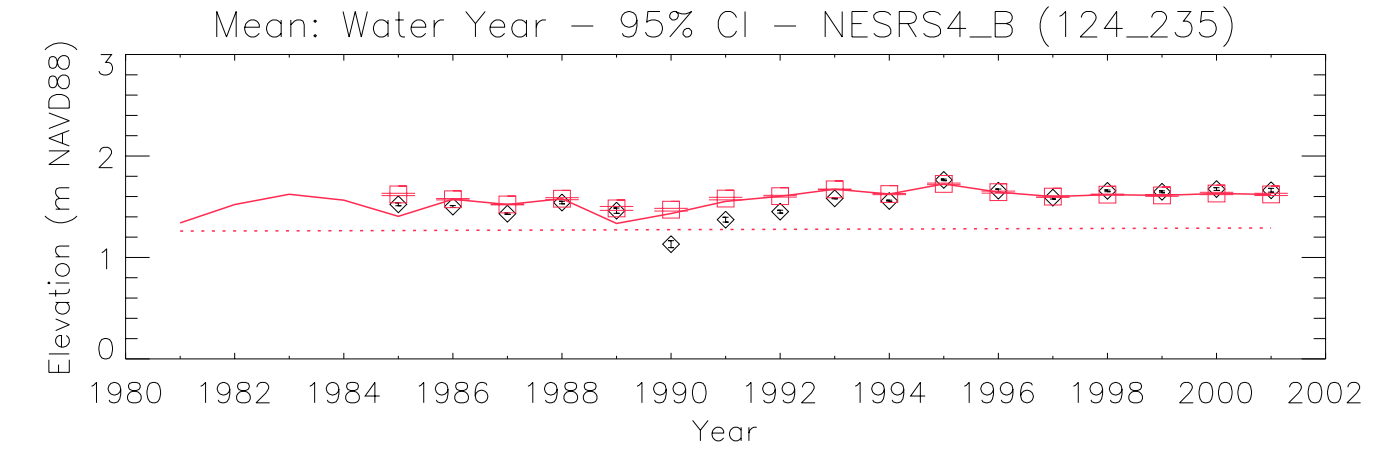

Mean: Season – 95% CI – NP–202 (112_230)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

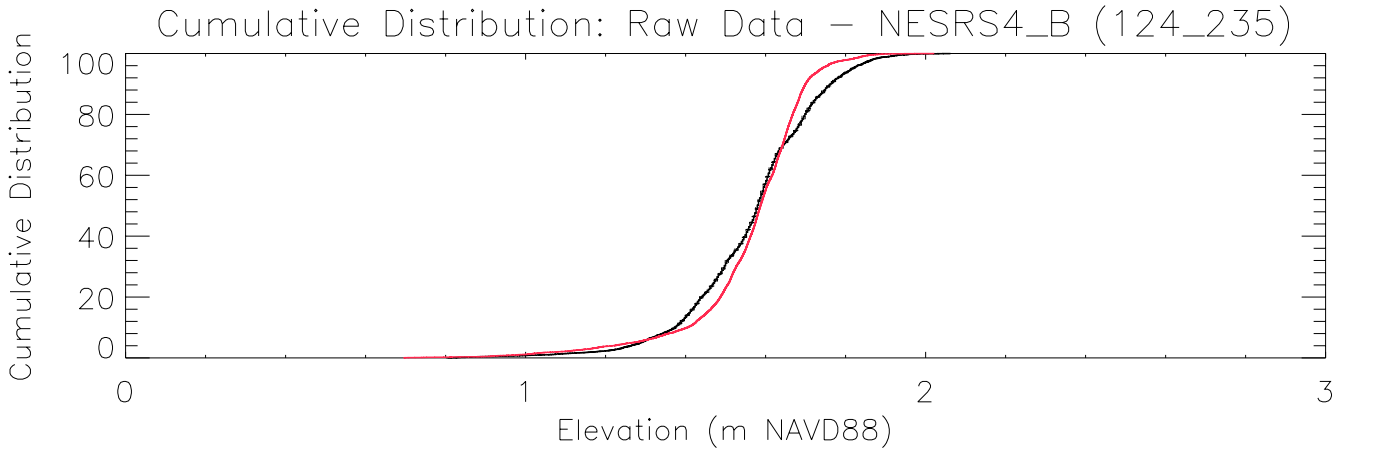

Mean: Water Year – 95% CI – NP–202 (112_230)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

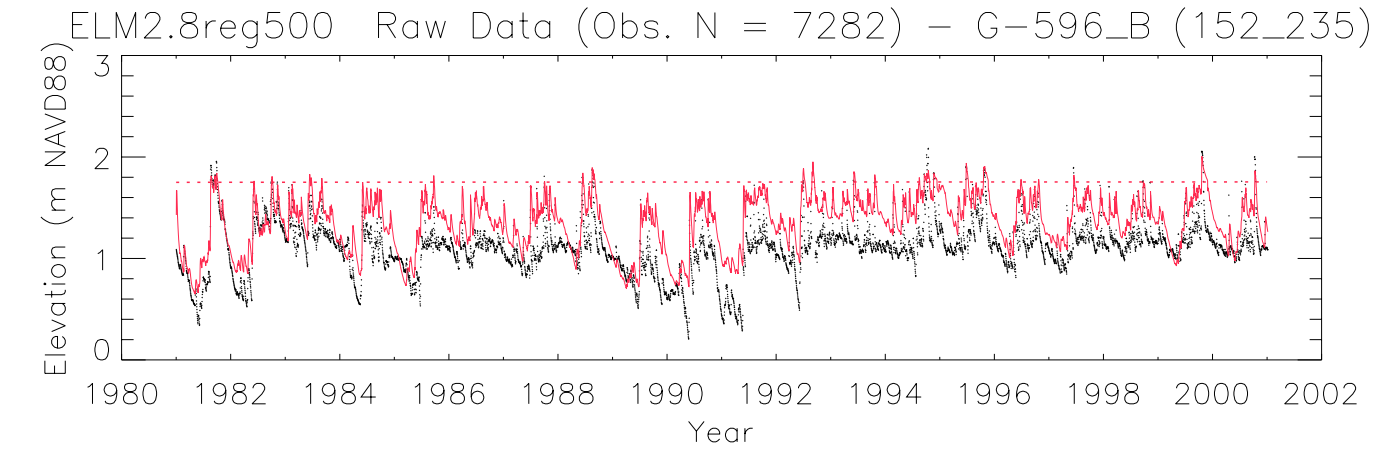

Cumulative Distribution: Raw Data – NP–202 (112_230)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

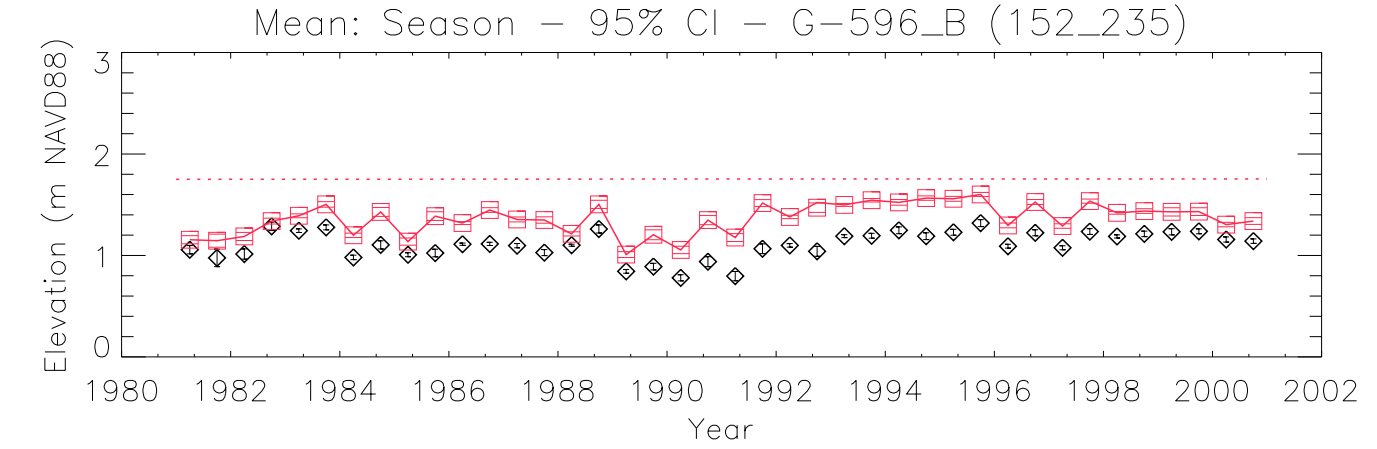

ELM2.8reg500 Raw Data (Obs. N = 4854) – NESRS4_B (124_235)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

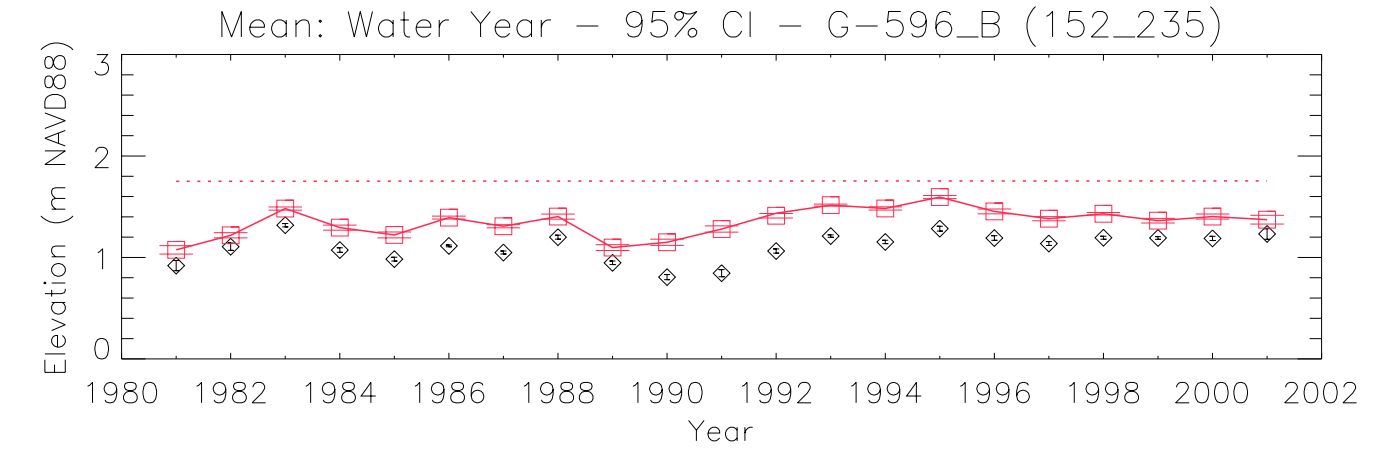

Mean: Season – 95% CI – NESRS4_B (124_235)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

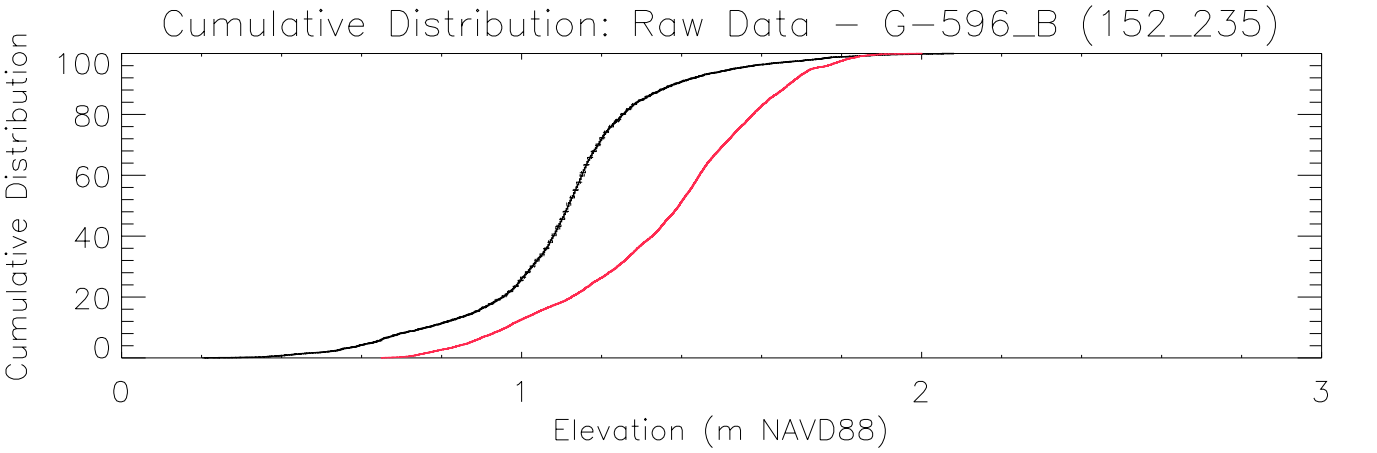

Mean: Water Year – 95% CI – NESRS4_B (124_235)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

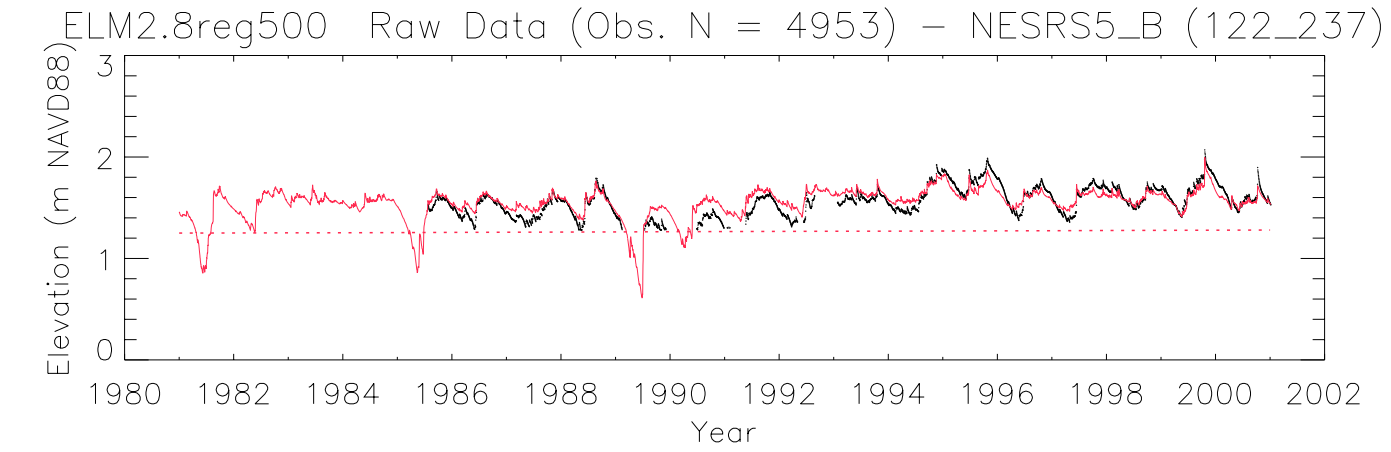

Cumulative Distribution: Raw Data – NESRS4_B (124_235)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

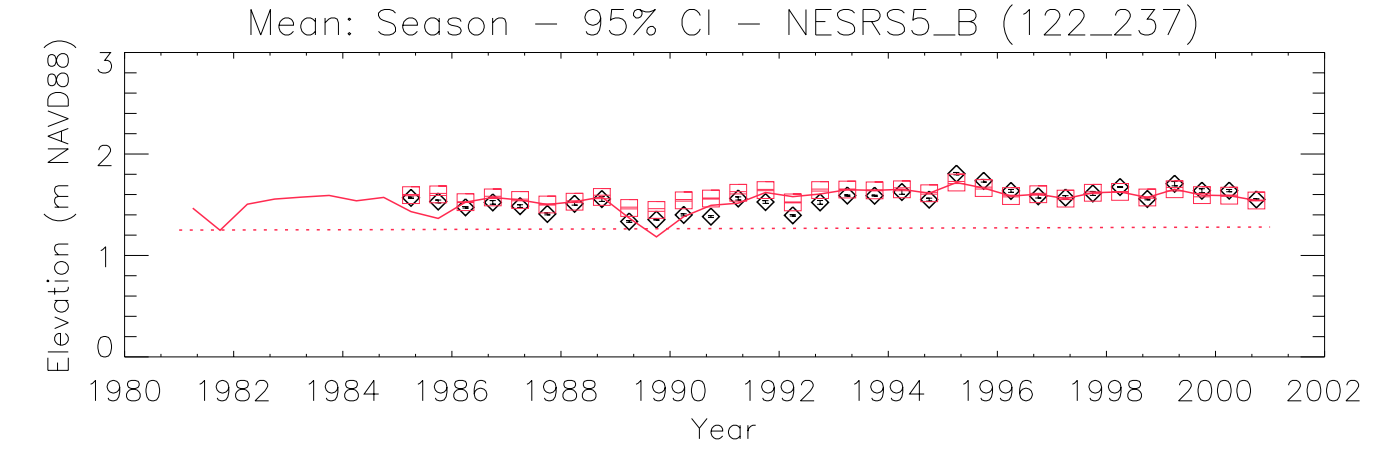

ELM2.8reg500 Raw Data (Obs. N = 7282) – G–596_B (152_235)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

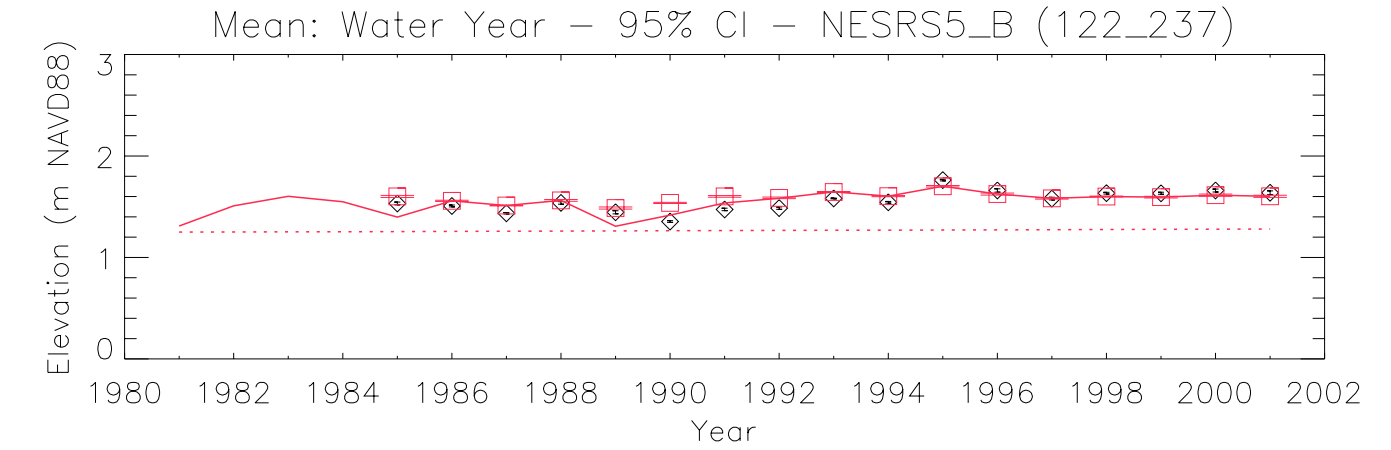

Mean: Season – 95% CI – G–596_B (152_235)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

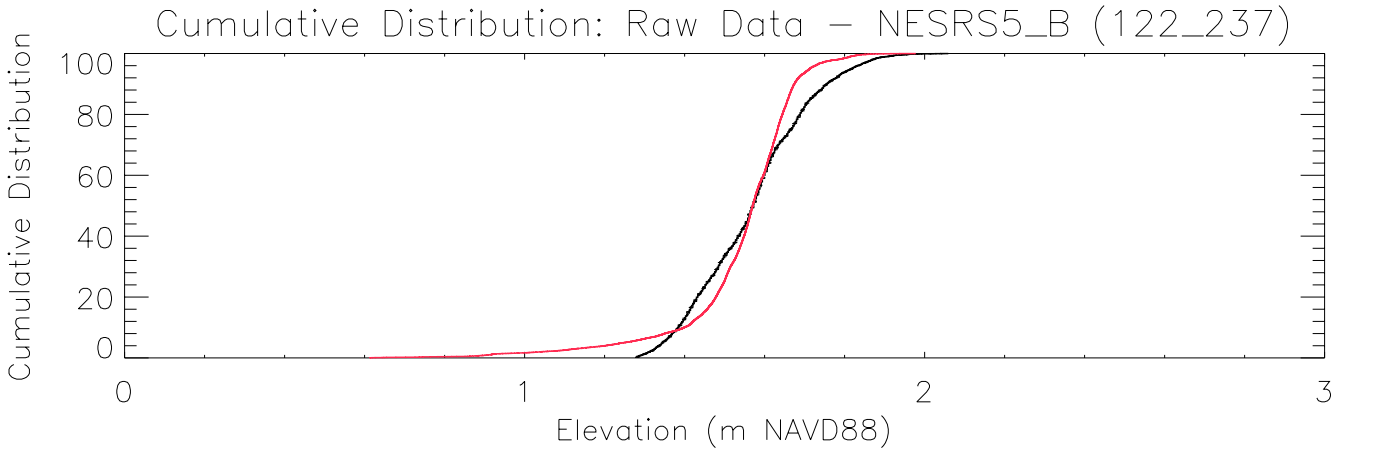

Mean: Water Year – 95% CI – G–596_B (152_235)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

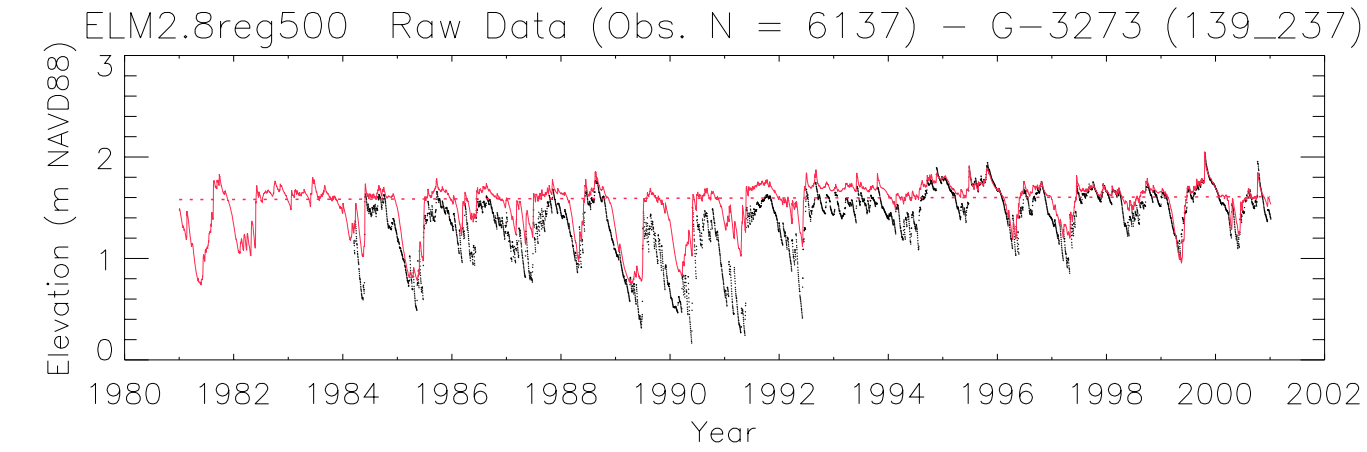

Cumulative Distribution: Raw Data – G–596_B (152_235)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

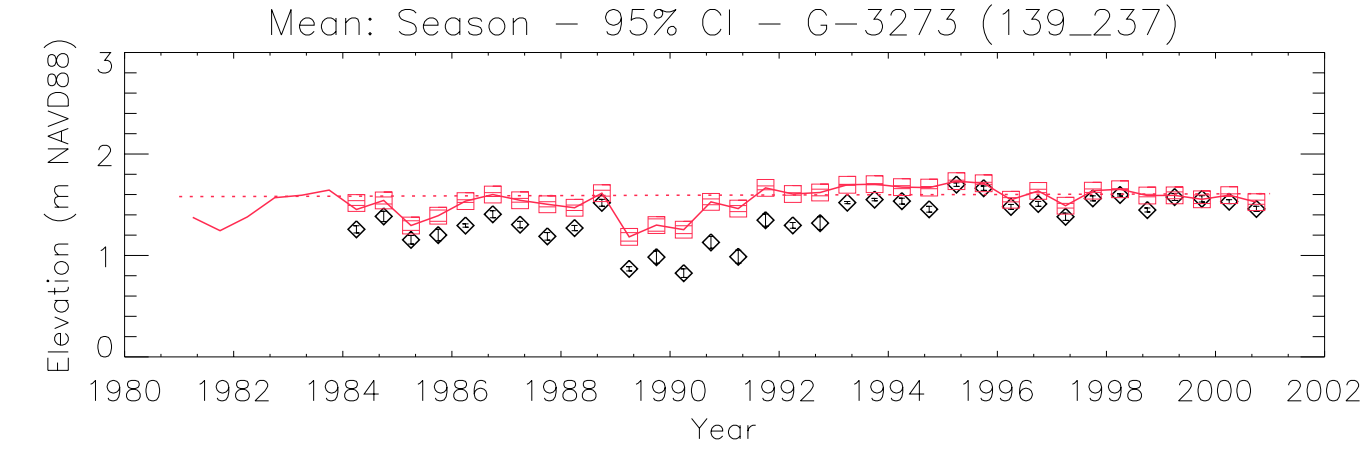

ELM2.8reg500 Raw Data (Obs. N = 4953) – NESRS5_B (122_237)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

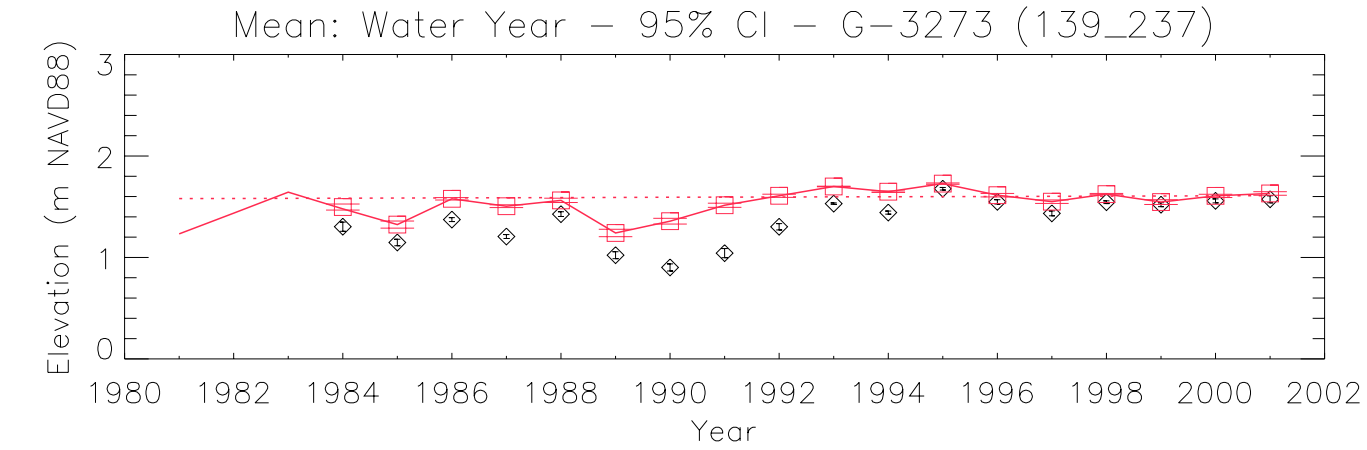

Mean: Season – 95% CI – NESRS5_B (122_237)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

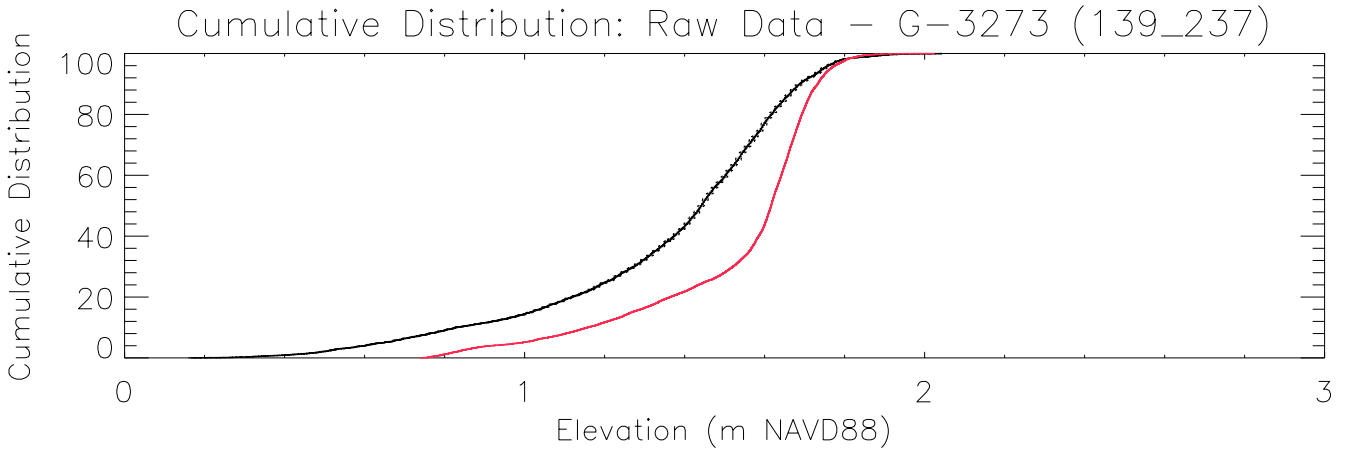

Mean: Water Year – 95% CI – NESRS5_B (122_237)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

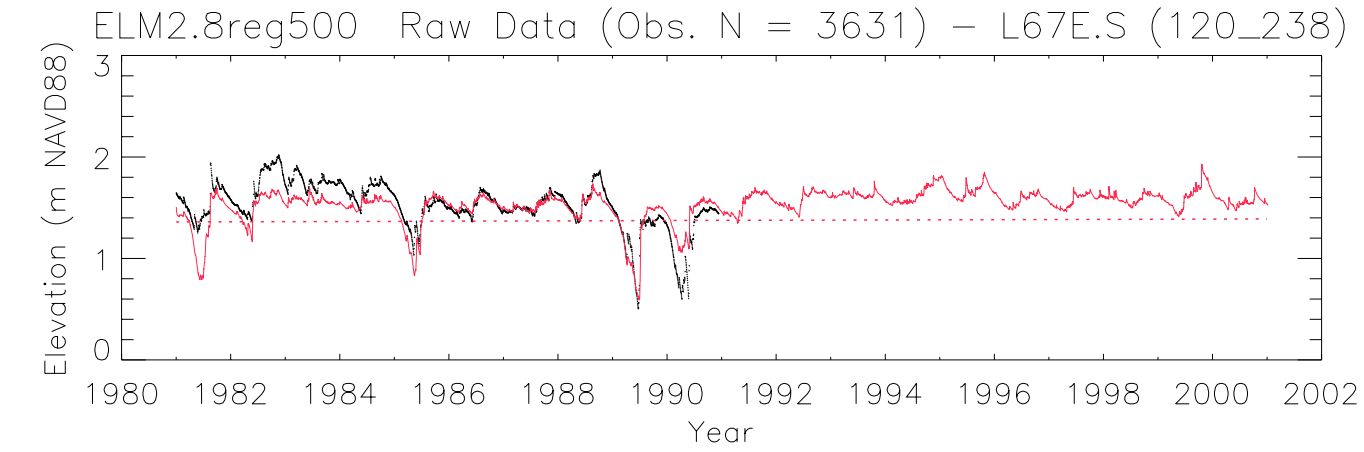

Cumulative Distribution: Raw Data – NESRS5_B (122_237)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

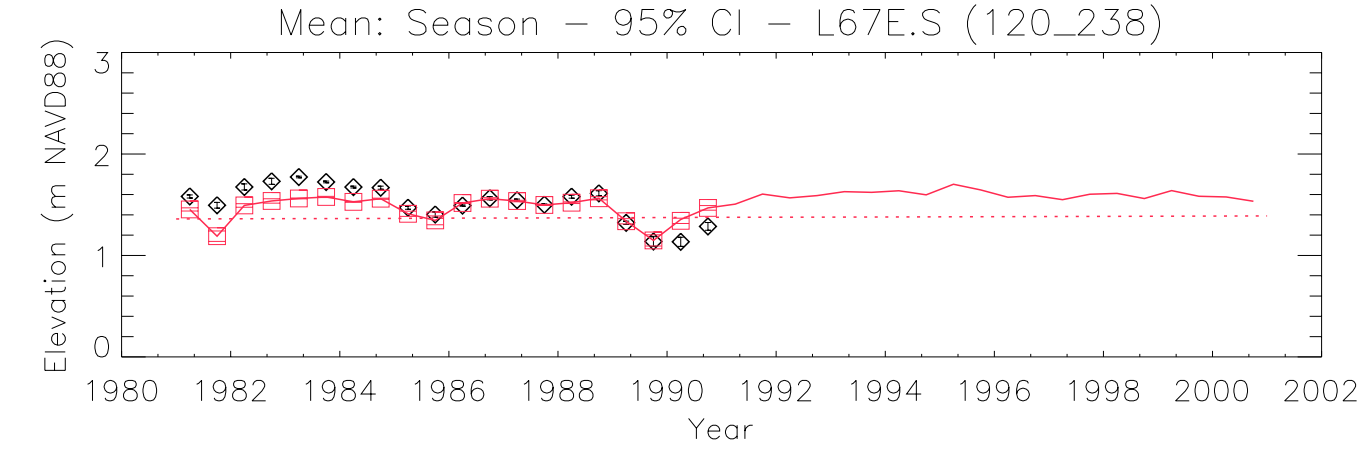

ELM2.8reg500 Raw Data (Obs. N = 6137) – G–3273 (139_237)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

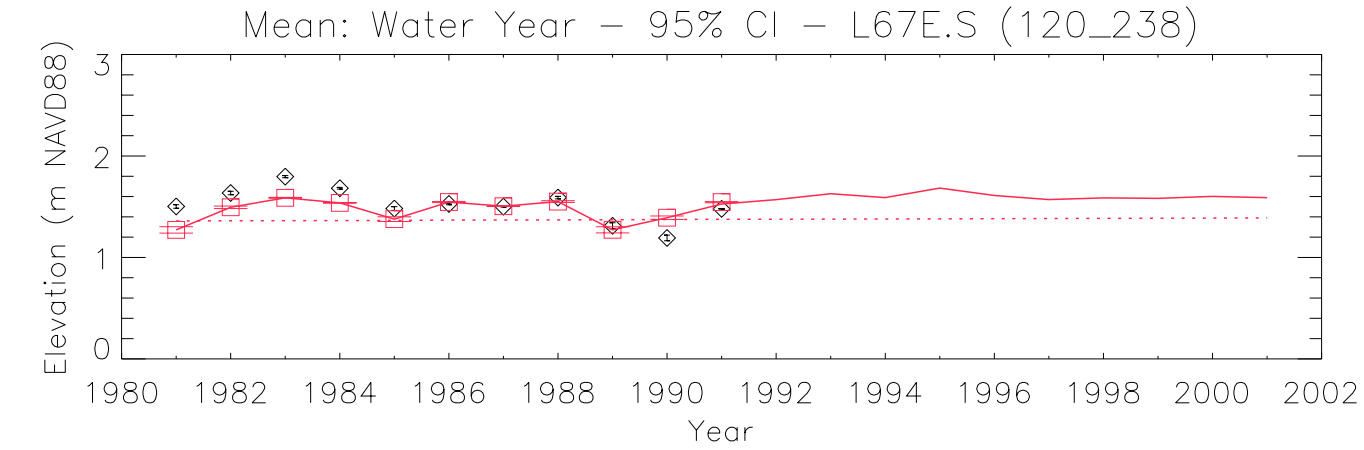

Mean: Season – 95% CI – G–3273 (139_237)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

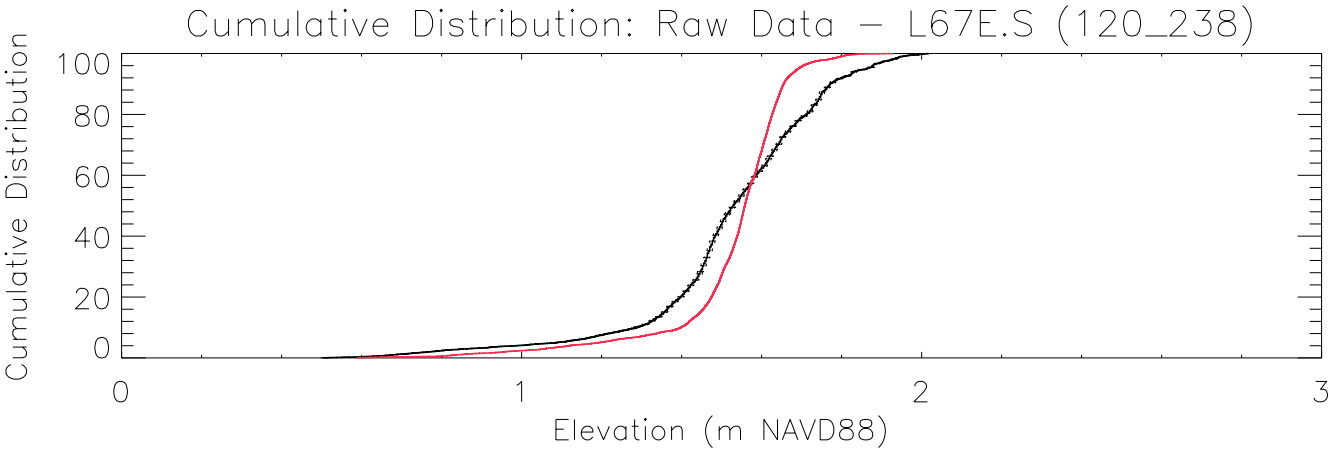

Mean: Water Year – 95% CI – G–3273 (139_237)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

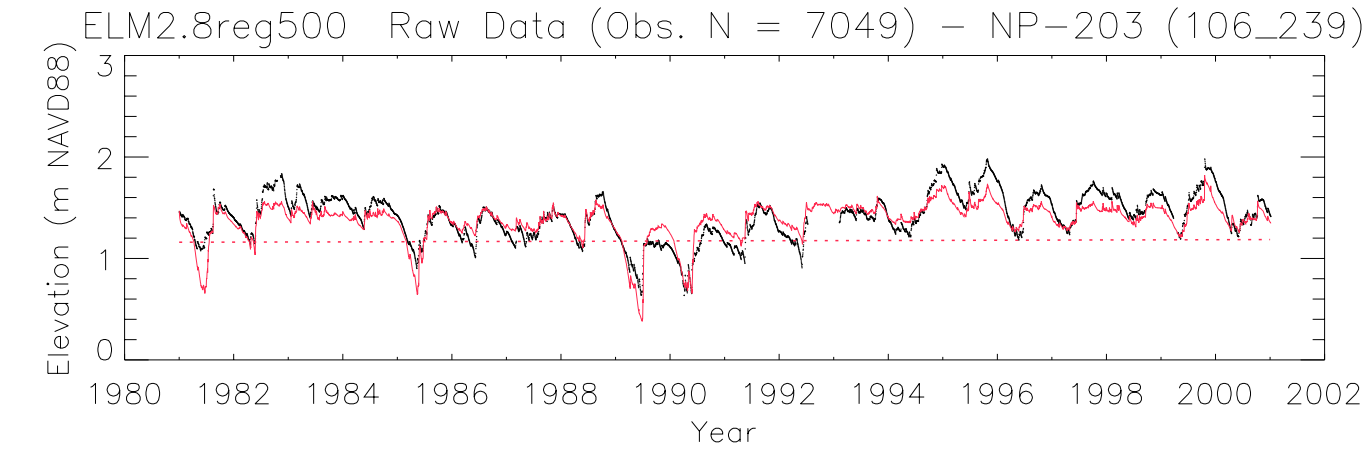

Cumulative Distribution: Raw Data – G–3273 (139_237)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

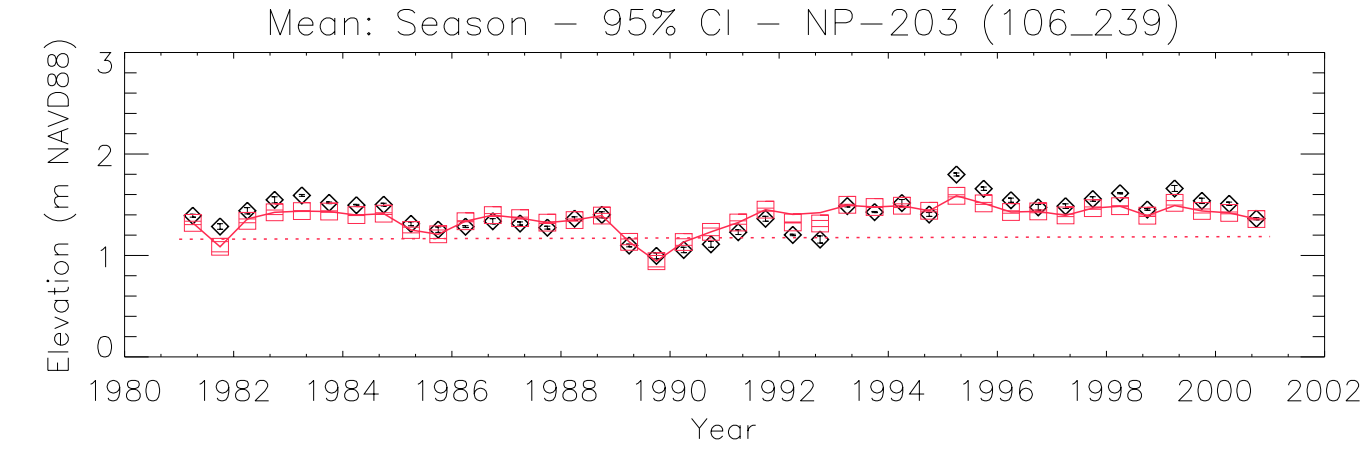

ELM2.8reg500 Raw Data (Obs. N = 3631) – L67E.S (120_238)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

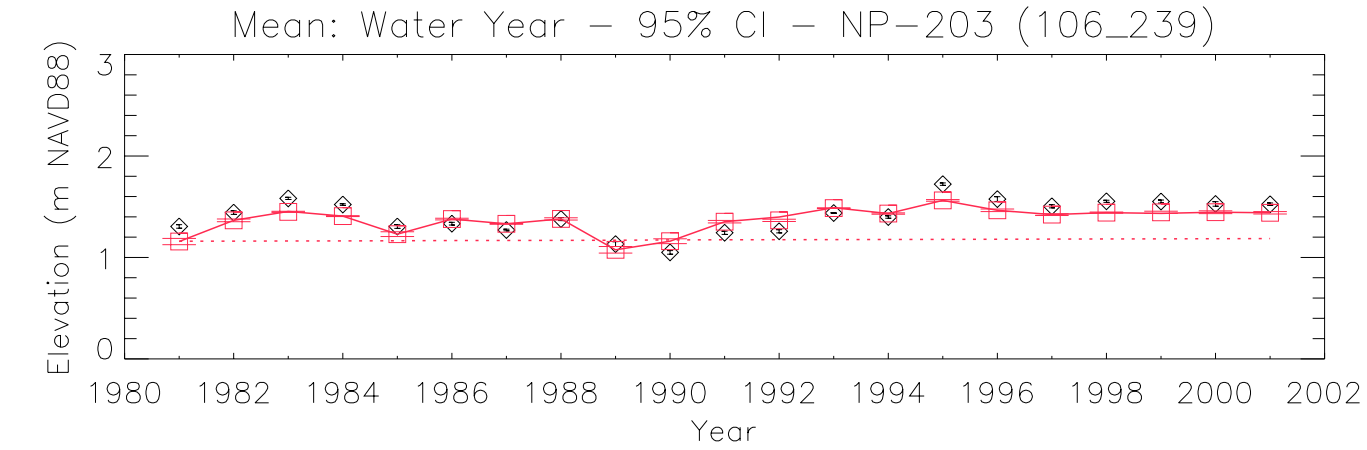

Mean: Season – 95% CI – L67E.S (120_238)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

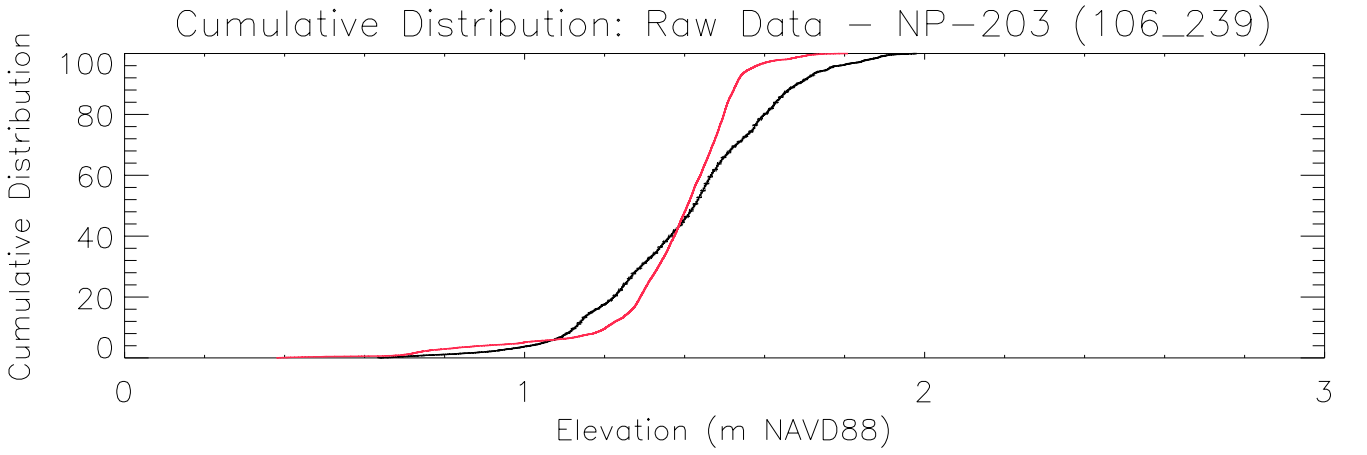

Mean: Water Year – 95% CI – L67E.S (120_238)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

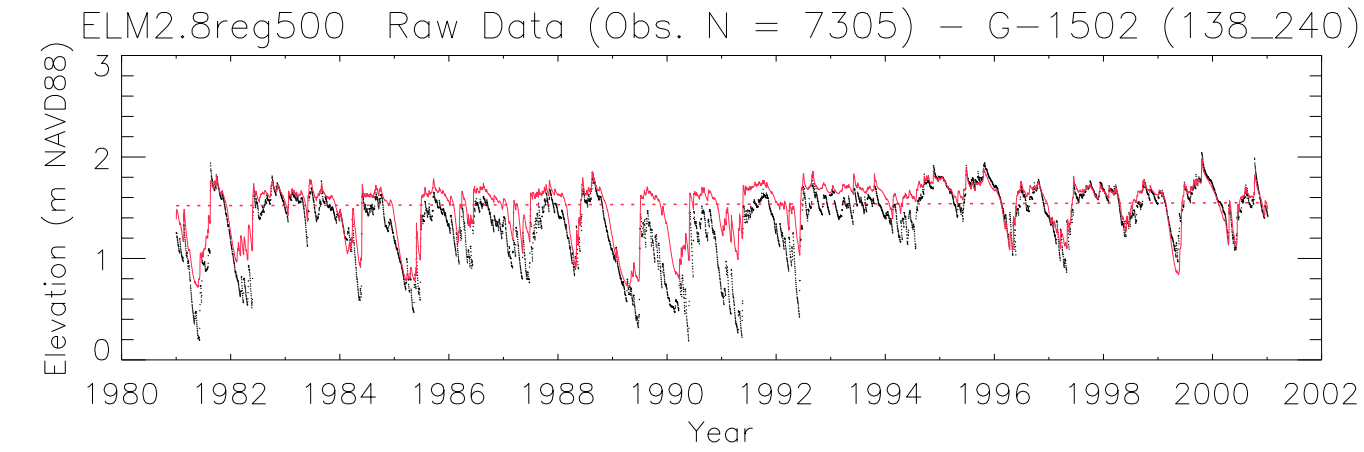

Cumulative Distribution: Raw Data – L67E.S (120_238)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

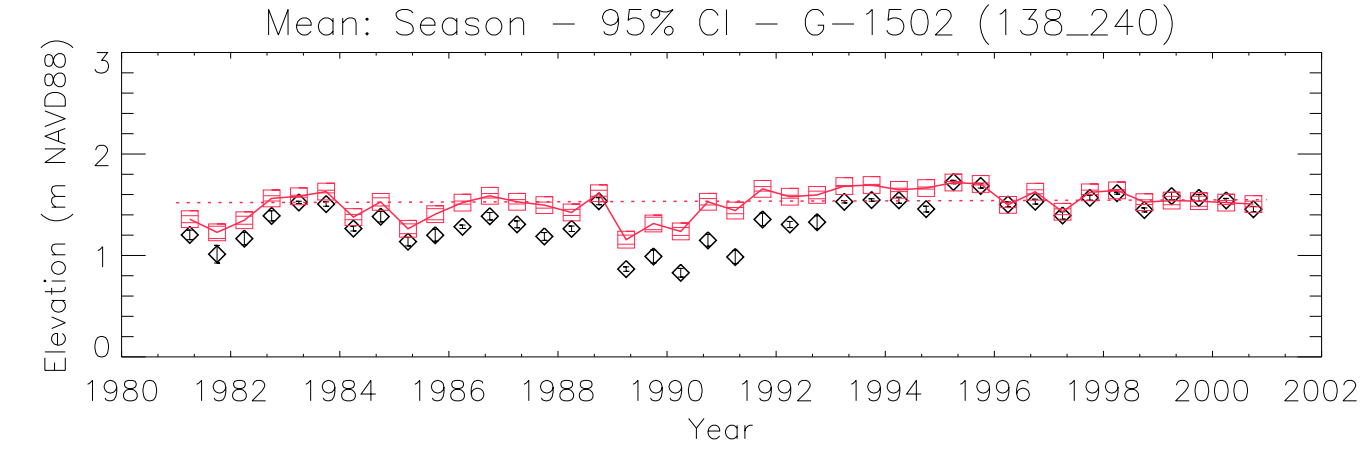

ELM2.8reg500 Raw Data (Obs. N = 7049) – NP–203 (106_239)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

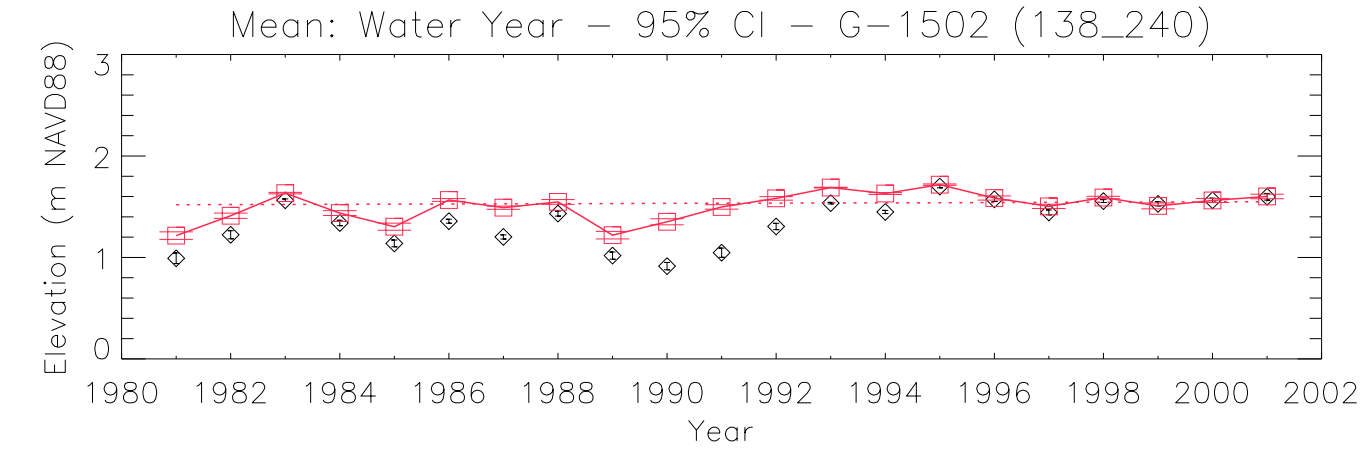

Mean: Season – 95% CI – NP–203 (106_239)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

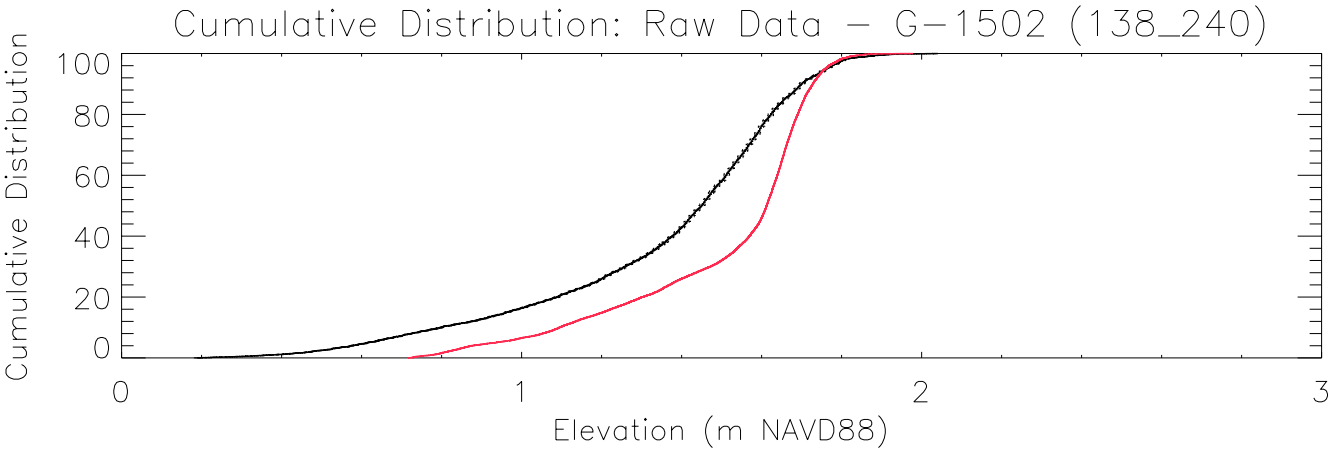

Mean: Water Year – 95% CI – NP–203 (106_239)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

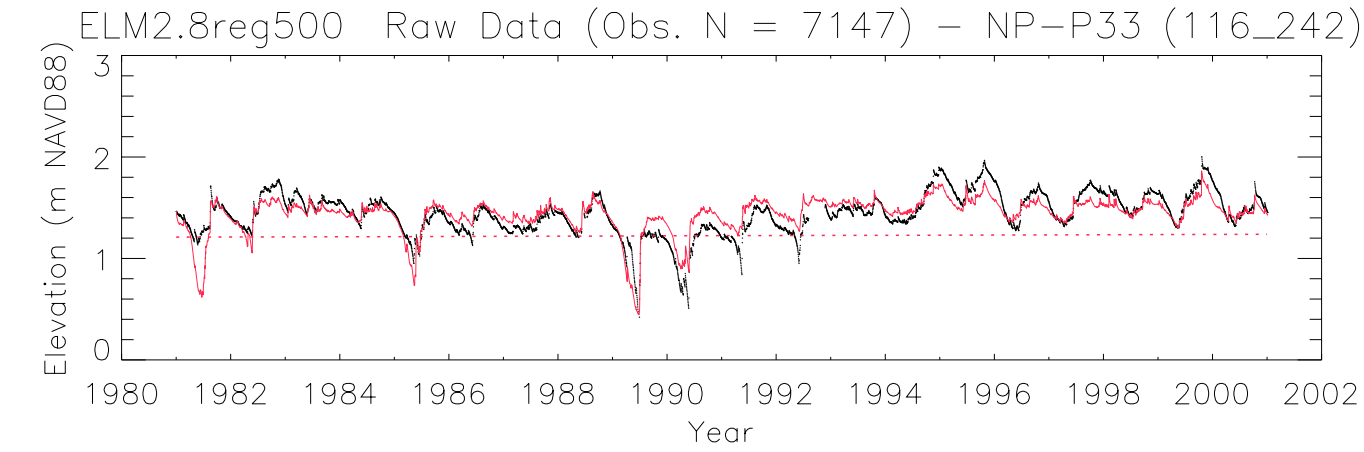

Cumulative Distribution: Raw Data – NP–203 (106_239)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

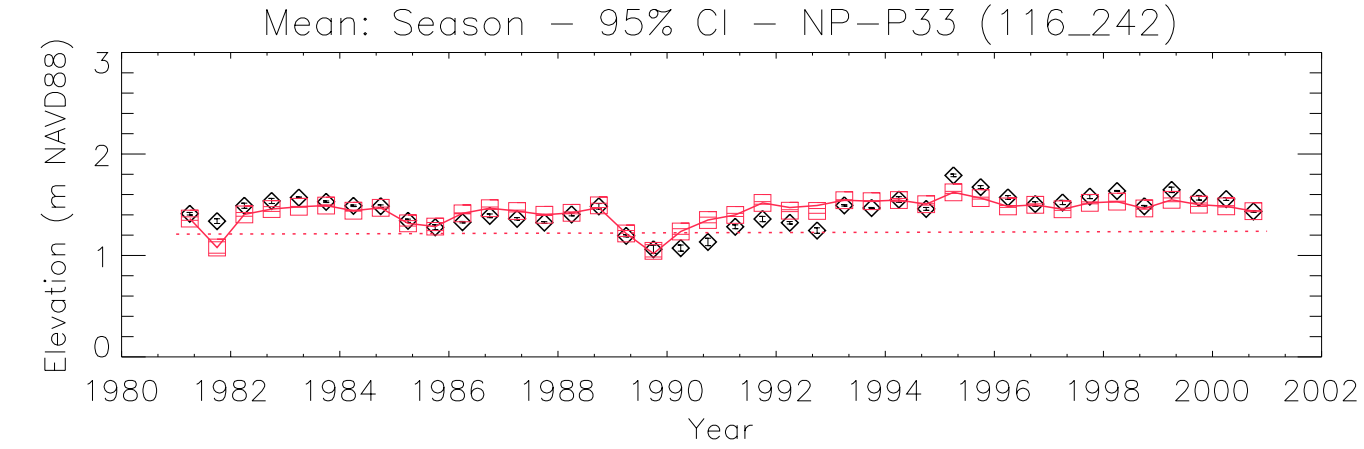

ELM2.8reg500 Raw Data (Obs. N = 7305) – G–1502 (138_240)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

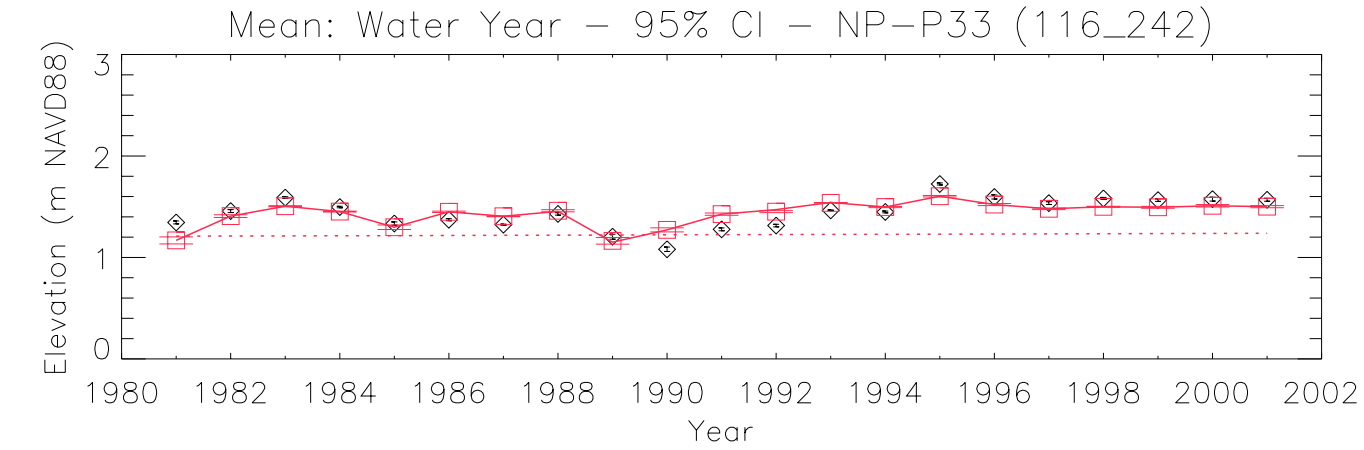

Mean: Season – 95% CI – G–1502 (138_240)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

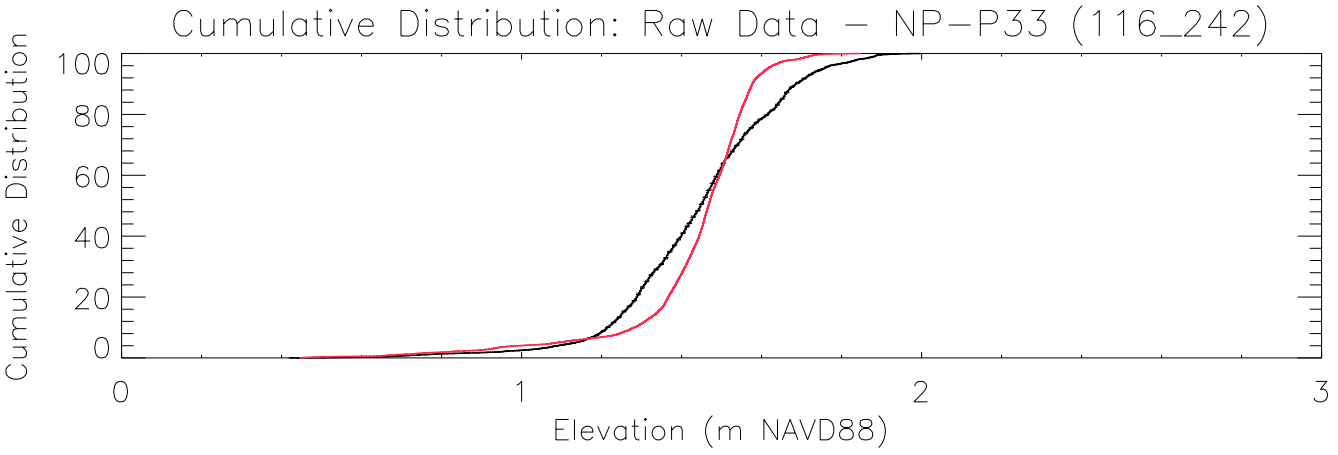

Mean: Water Year – 95% CI – G–1502 (138_240)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

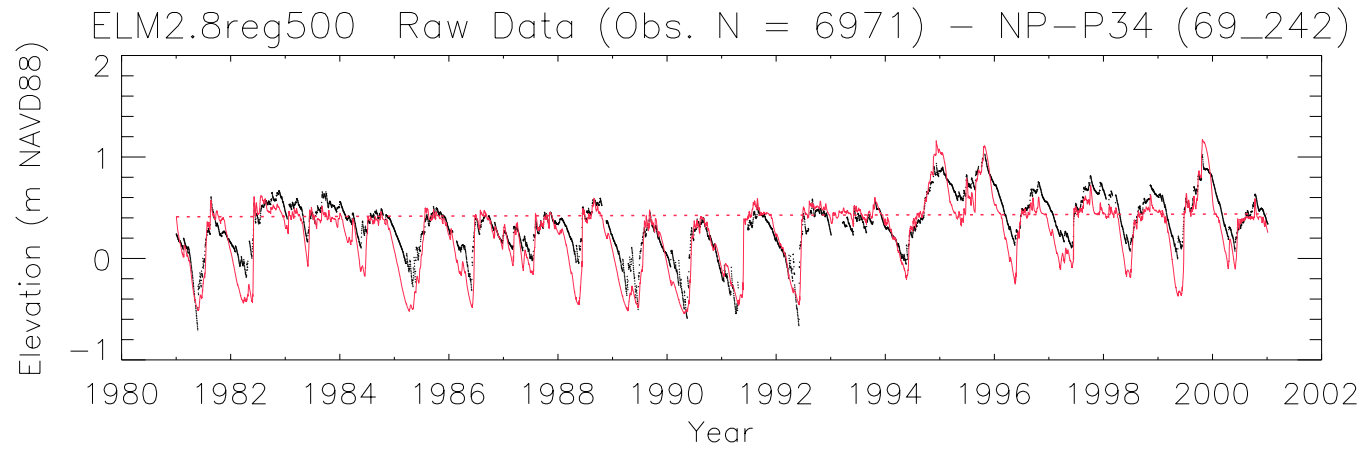

Cumulative Distribution: Raw Data – G–1502 (138_240)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

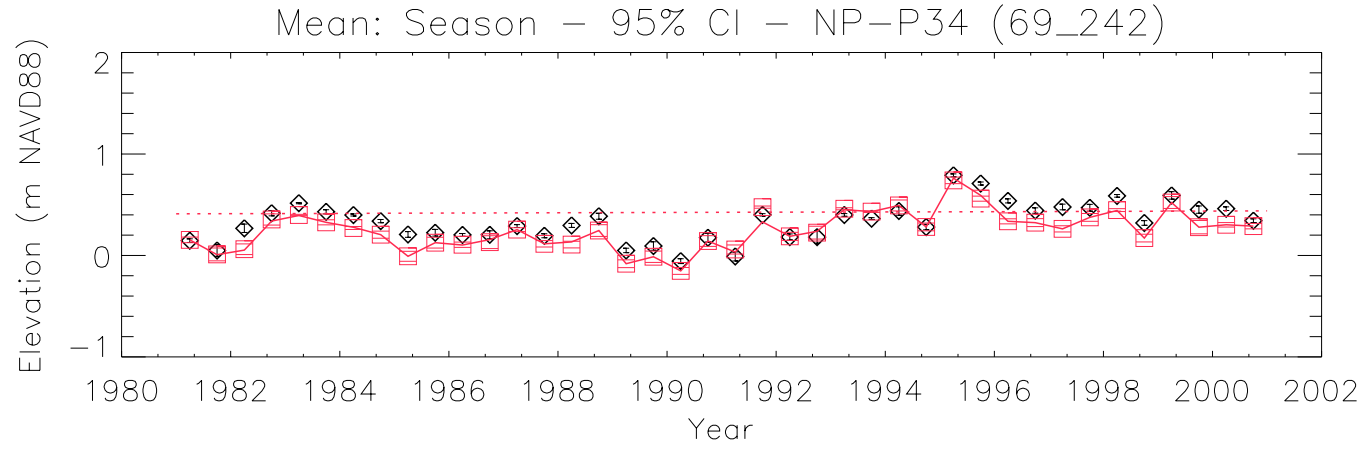

ELM2.8reg500 Raw Data (Obs. N = 7147) – NP–P33 (116_242)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

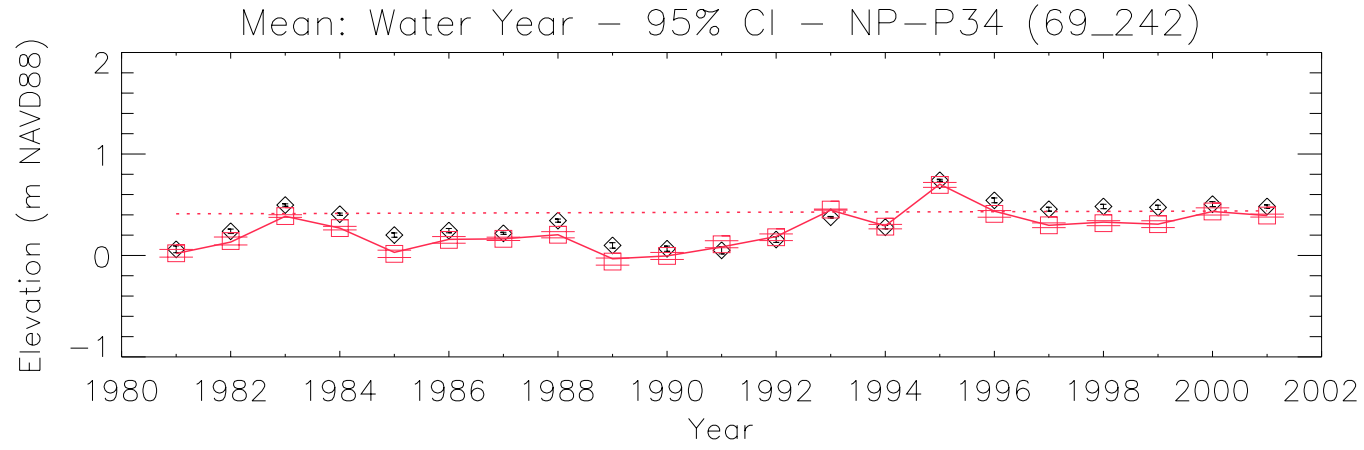

Mean: Season – 95% CI – NP–P33 (116_242)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

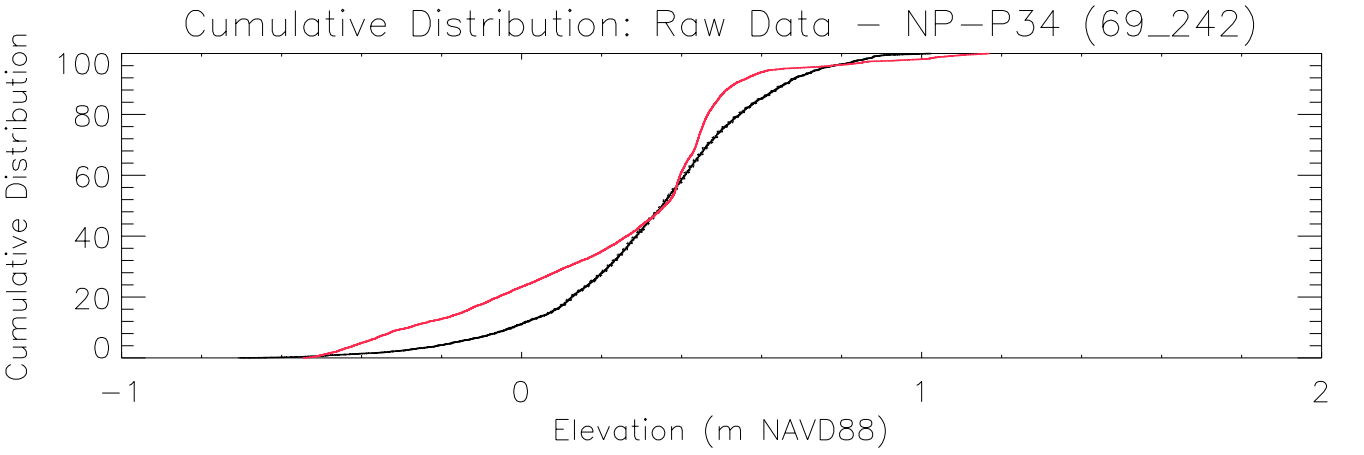

Mean: Water Year – 95% CI – NP–P33 (116_242)
Elevation (m NAVD88)
0
1
2
3
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

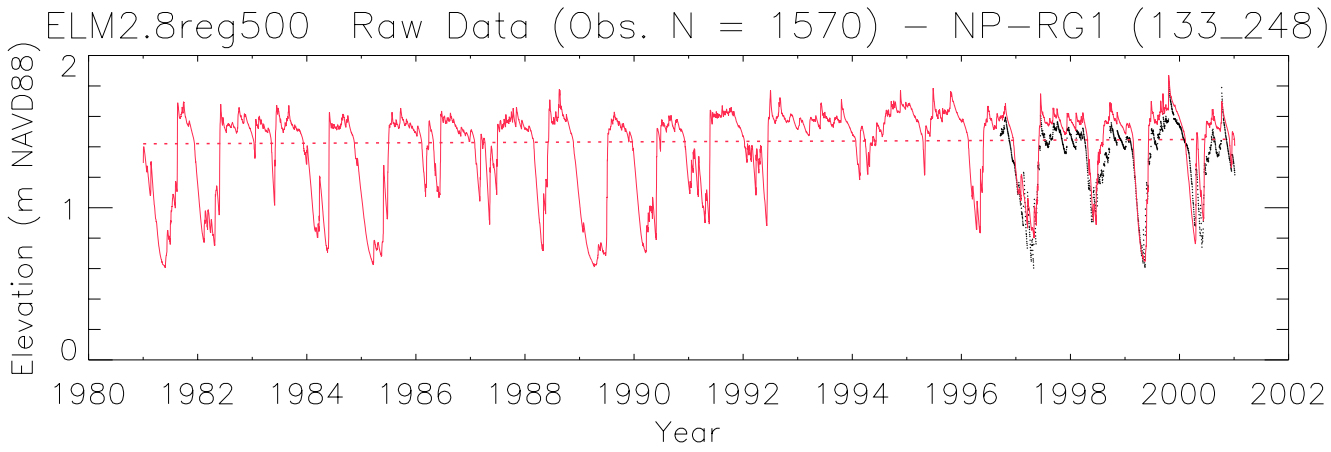

Cumulative Distribution: Raw Data – NP–P33 (116_242)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
3
Elevation (m NAVD88)

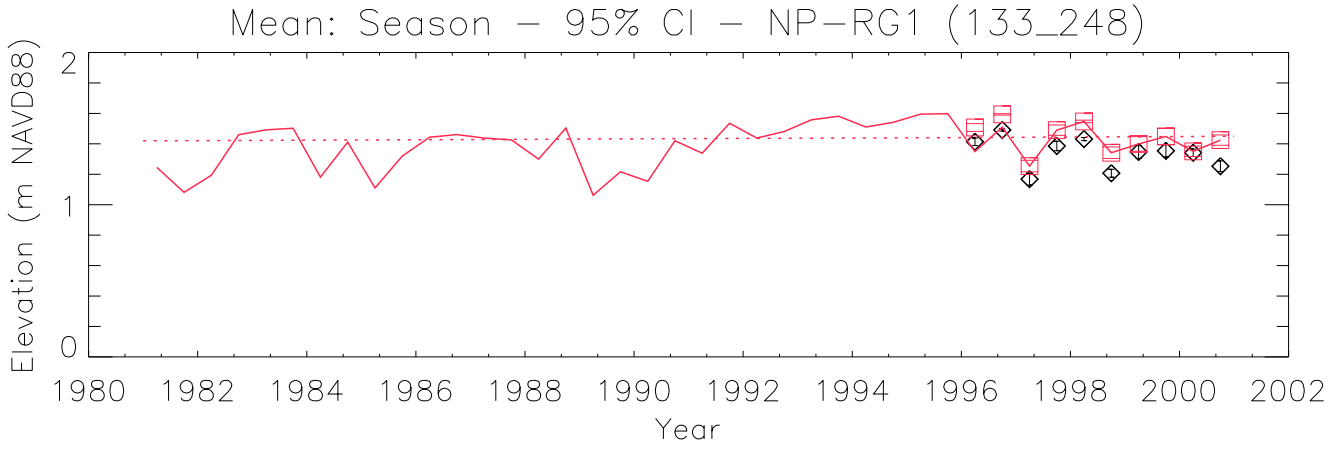

ELM2.8reg500 Raw Data (Obs. N = 6971) – NP–P34 (69_242)
Elevation (m NAVD88)
2
1
0
–1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

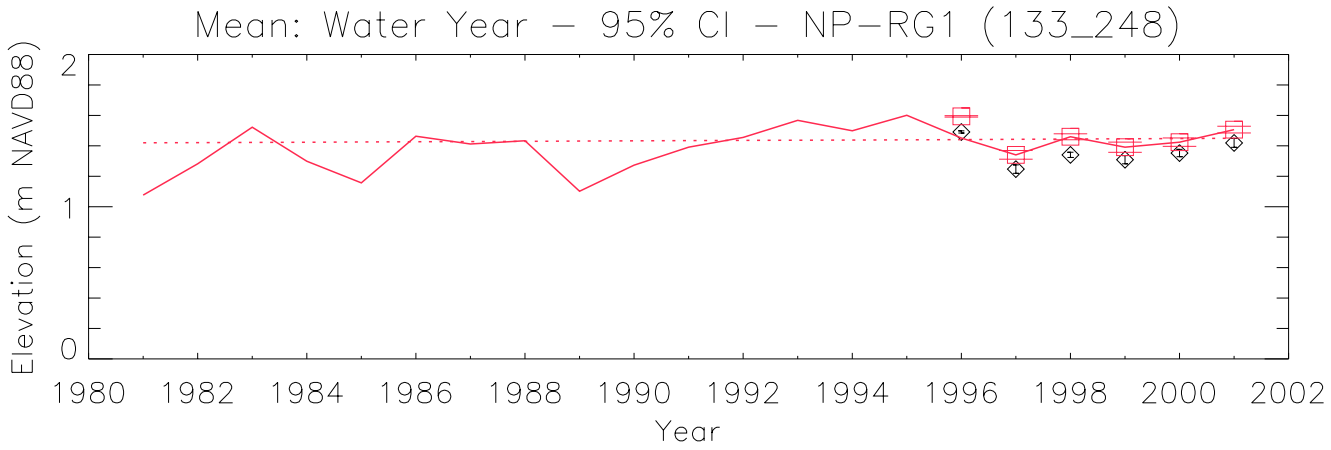

Mean: Season – 95% CI – NP–P34 (69_242)
Elevation (m NAVD88)
2
1
0
–1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

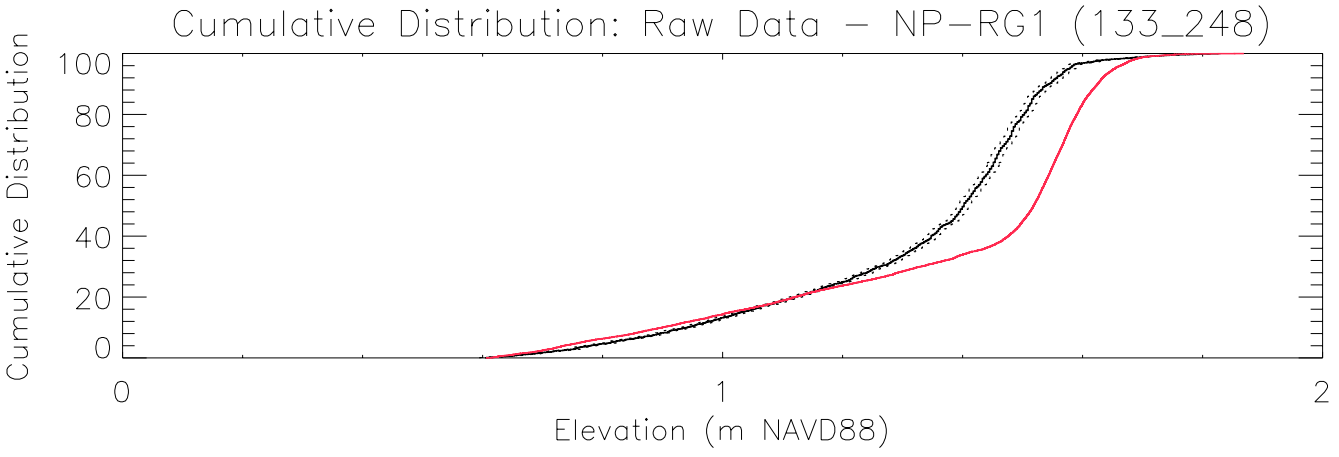

Mean: Water Year – 95% CI – NP–P34 (69_242)
Elevation (m NAVD88)
2
1
0
–1
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

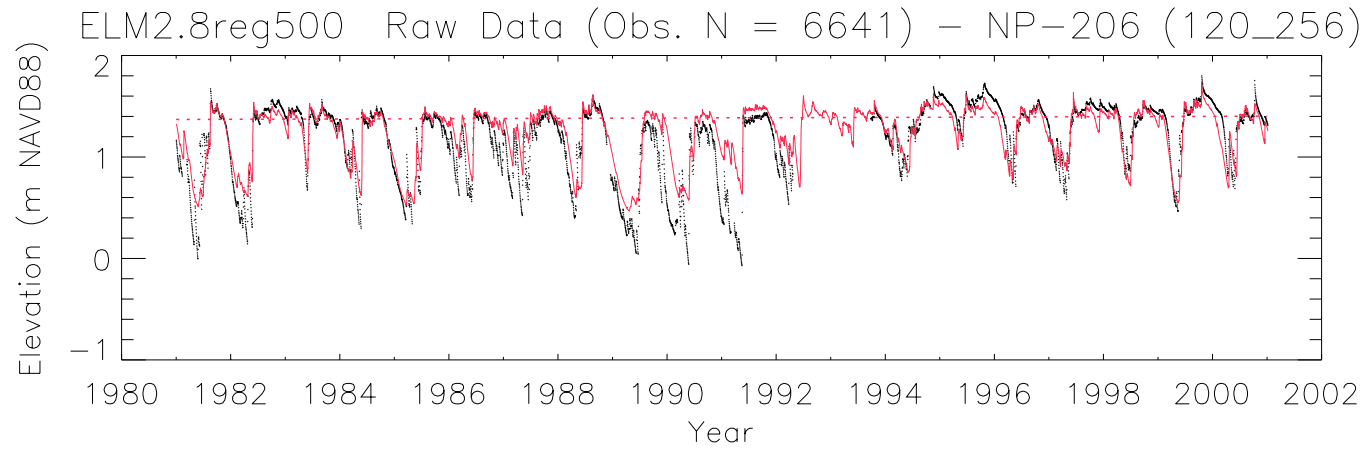

Cumulative Distribution: Raw Data – NP–P34 (69_242)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
2
Elevation (m NAVD88)

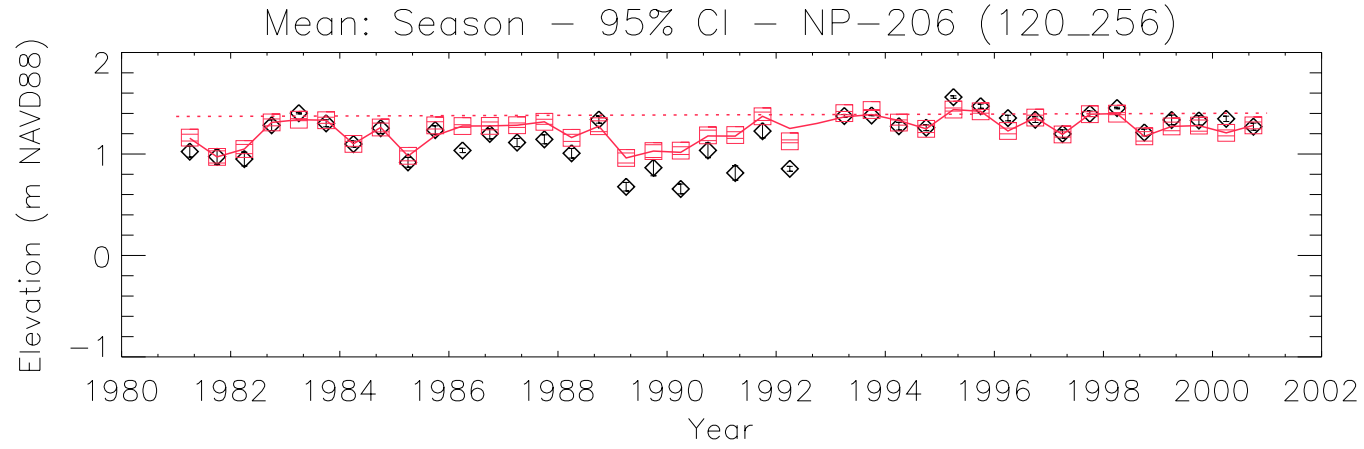

ELM2.8reg500 Raw Data (Obs. N = 1570) – NP–RG1 (133_248)
Elevation (m NAVD88)
0
1
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

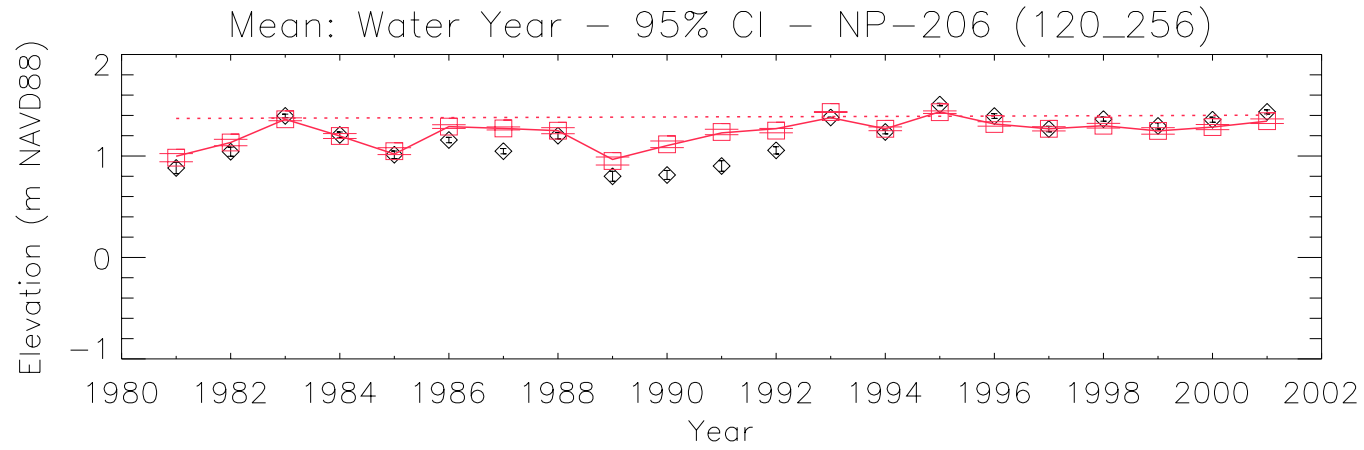

Mean: Season – 95% CI – NP–RG1 (133_248)
Elevation (m NAVD88)
0
1
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

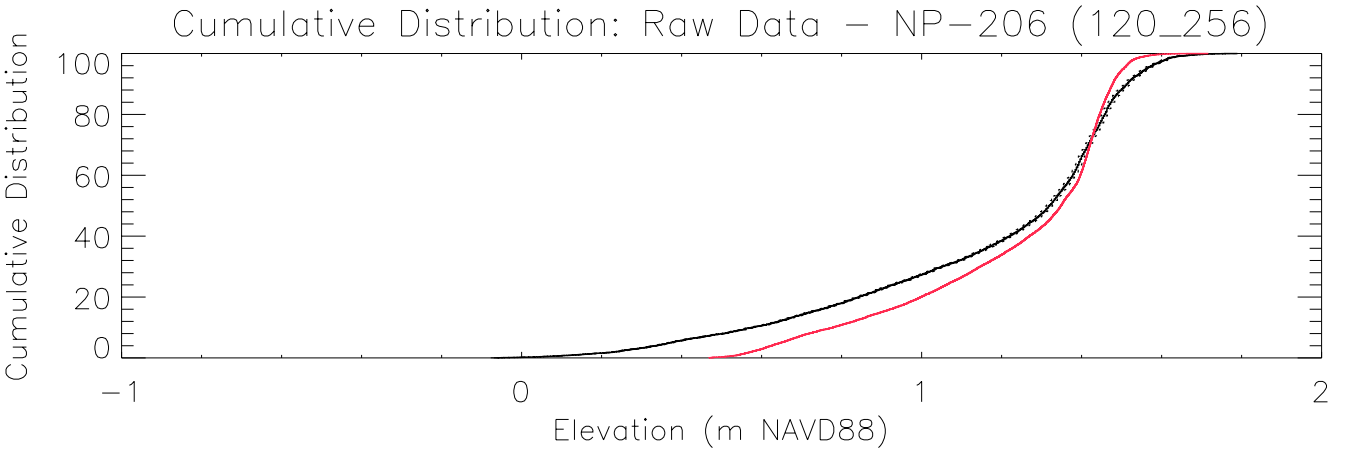

Mean: Water Year – 95% CI – NP–RG1 (133_248)
Elevation (m NAVD88)
0
1
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

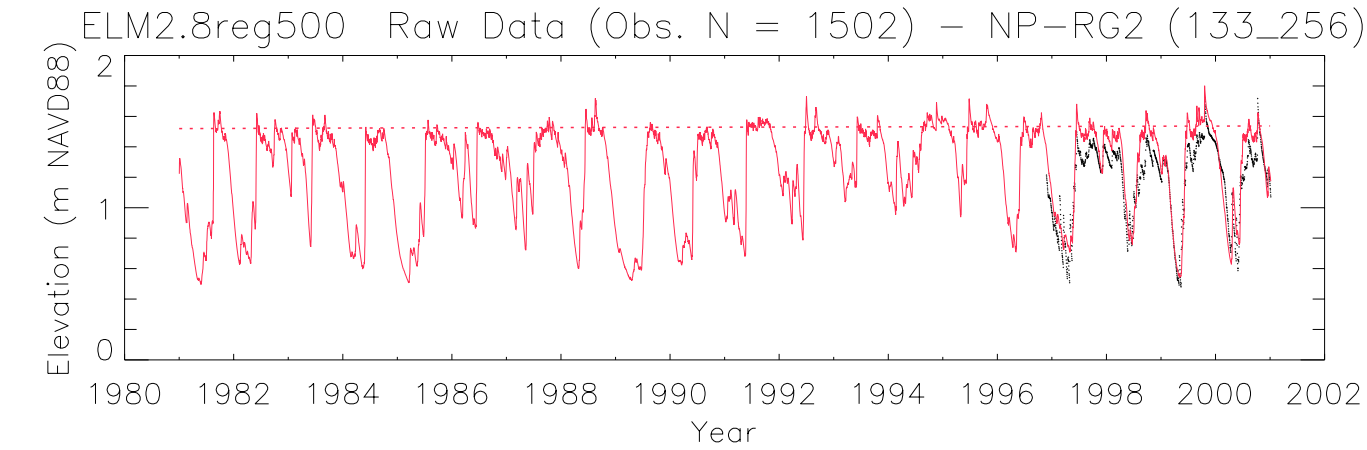

Cumulative Distribution: Raw Data – NP–RG1 (133_248)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
Elevation (m NAVD88)

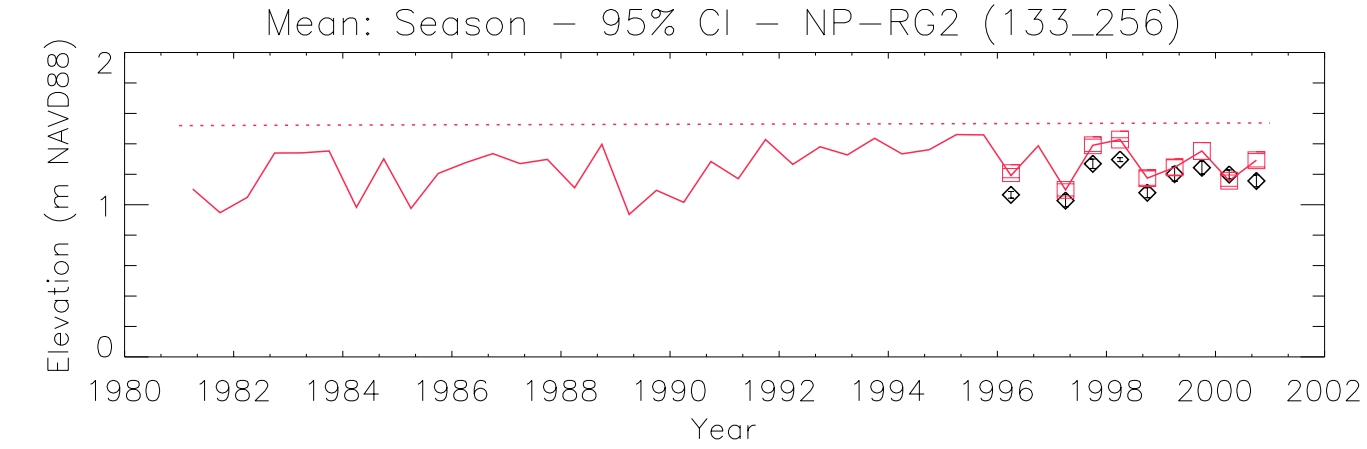

ELM2.8reg500 Raw Data (Obs. N = 6641) – NP–206 (120_256)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

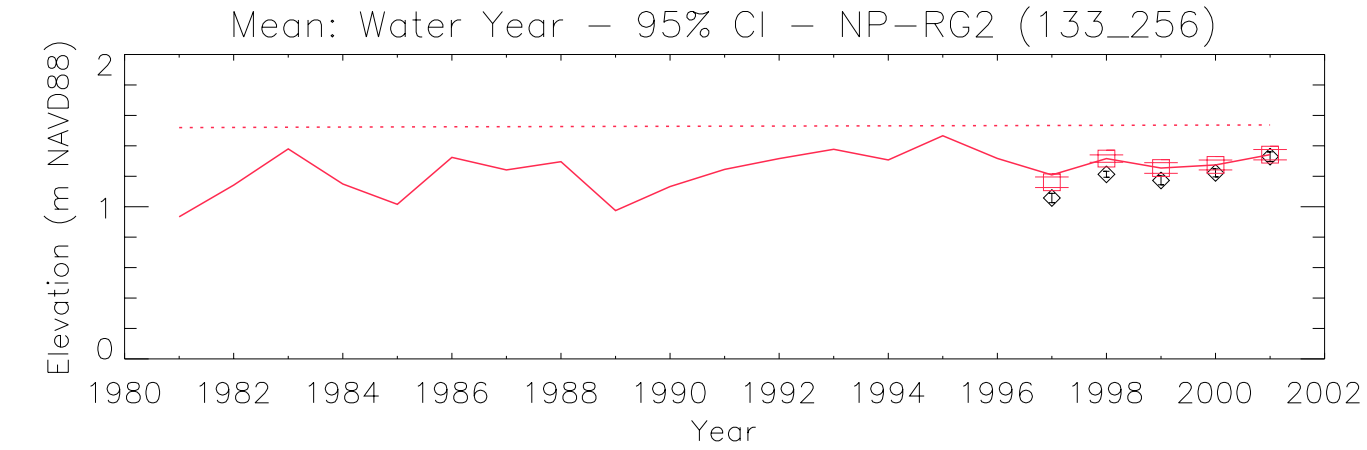

Mean: Season – 95% CI – NP–206 (120_256)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

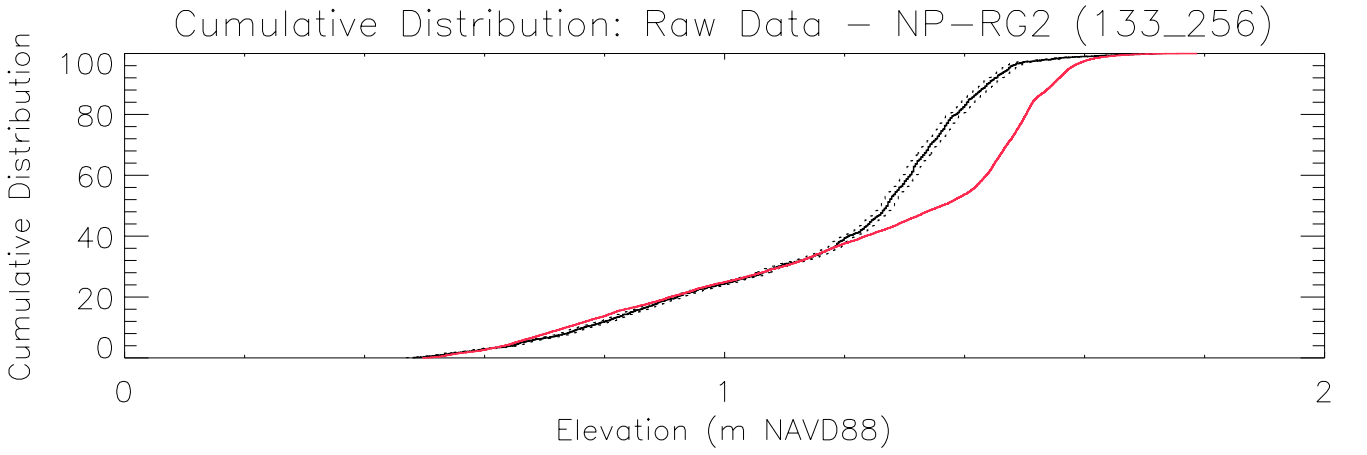

Mean: Water Year – 95% CI – NP–206 (120_256)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

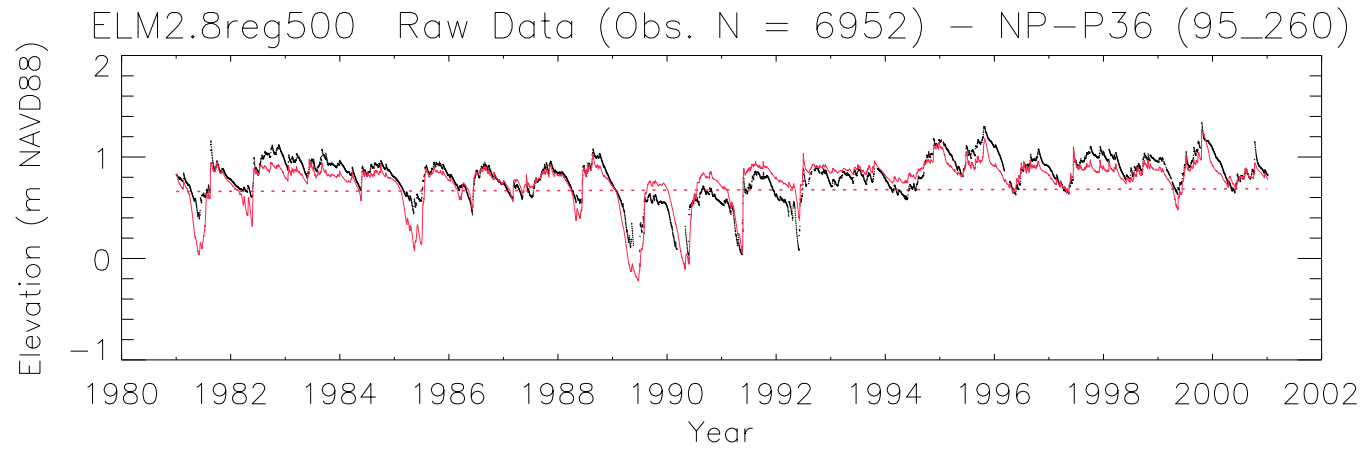

Cumulative Distribution: Raw Data – NP–206 (120_256)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
2
Elevation (m NAVD88)

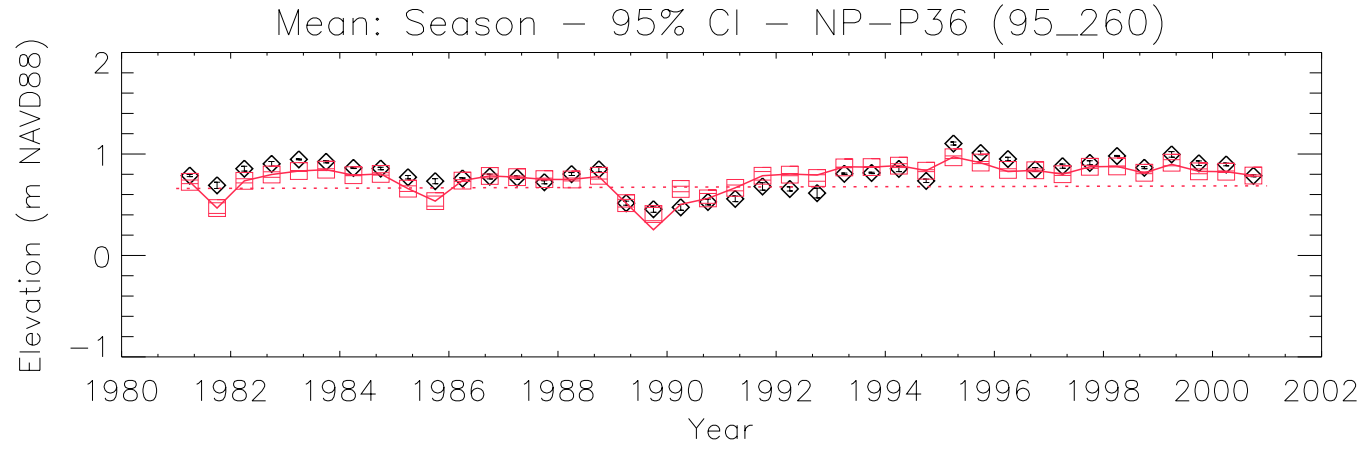

ELM2.8reg500 Raw Data (Obs. N = 1502) – NP–RG2 (133_256)
Elevation (m NAVD88)
0
1
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

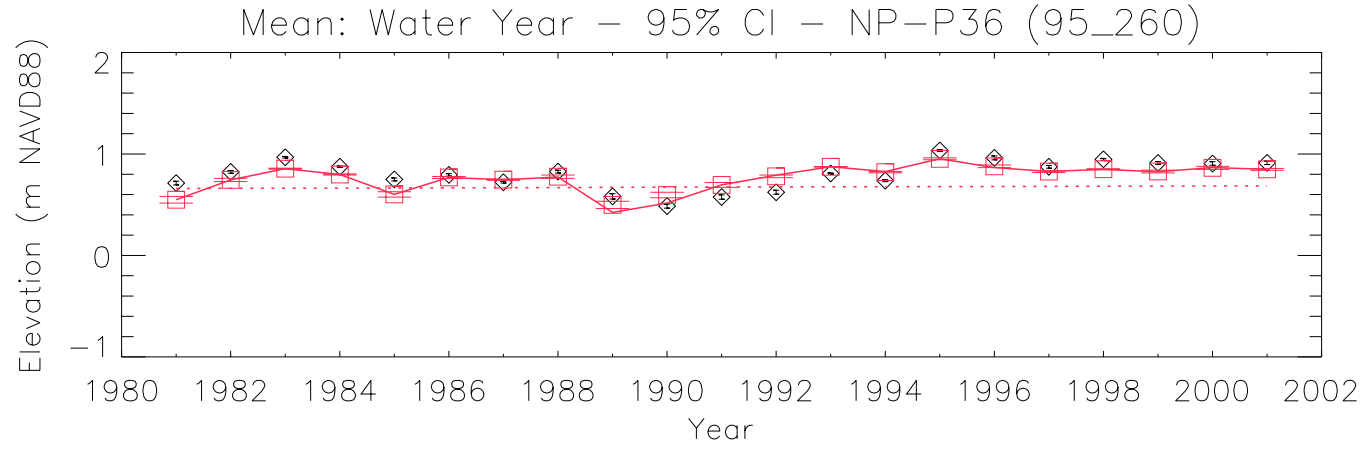

Mean: Season – 95% CI – NP–RG2 (133_256)
Elevation (m NAVD88)
0
1
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

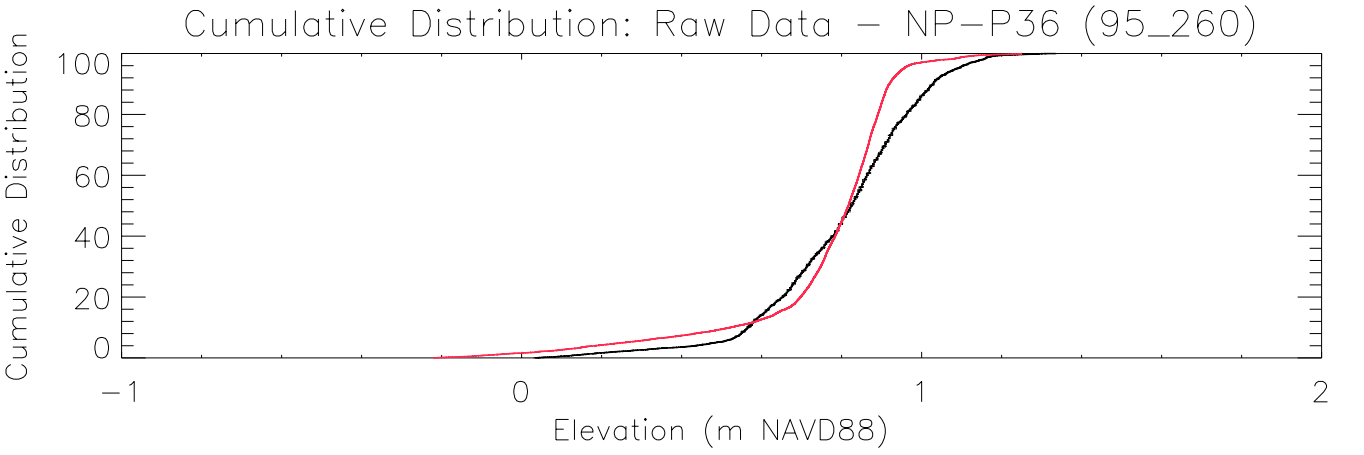

Mean: Water Year – 95% CI – NP–RG2 (133_256)
Elevation (m NAVD88)
0
1
2
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

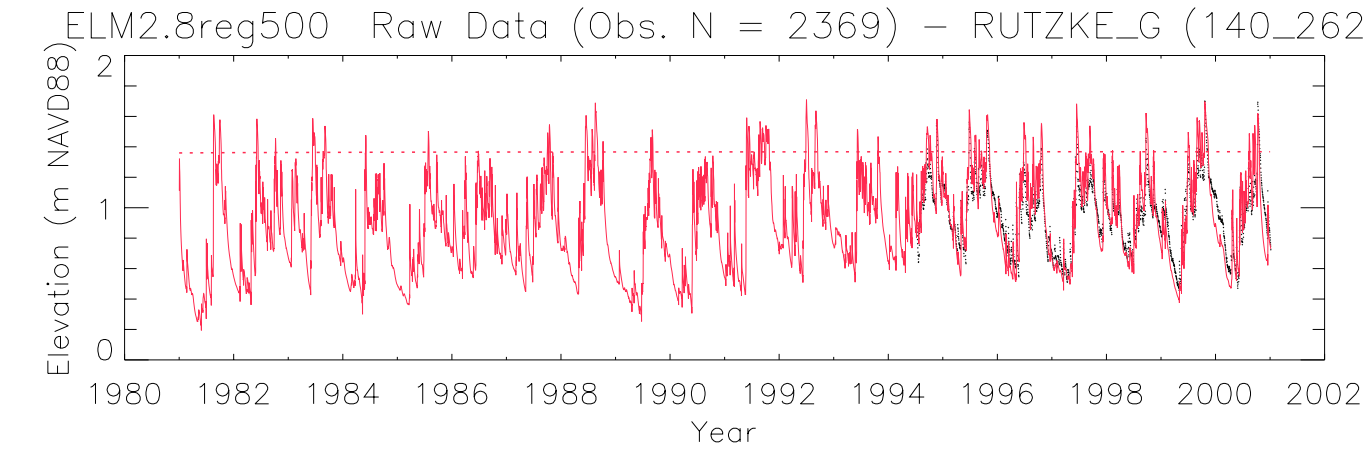

Cumulative Distribution: Raw Data – NP–RG2 (133_256)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
Elevation (m NAVD88)

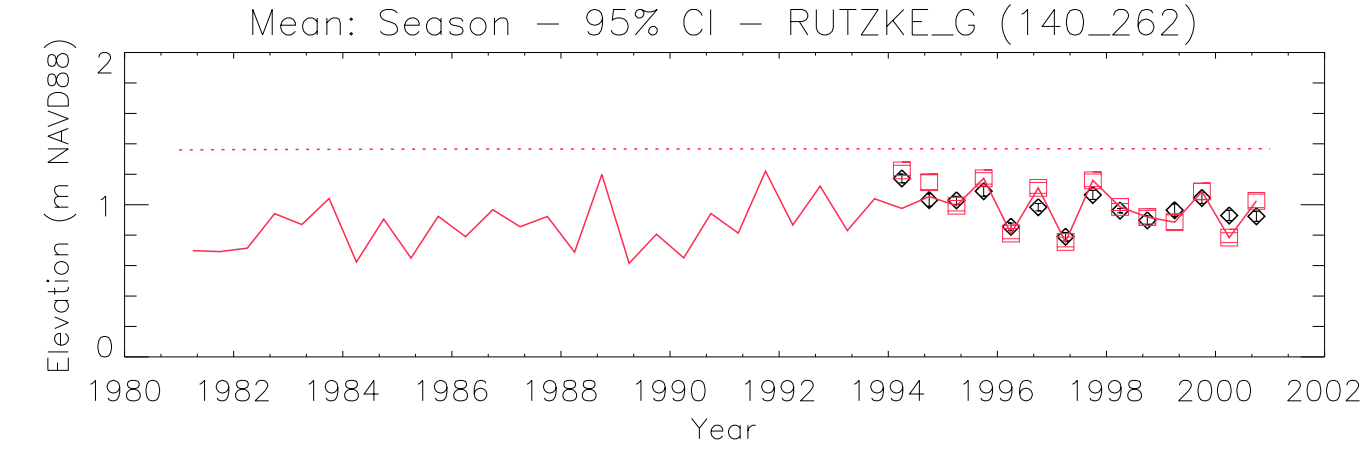

ELM2.8reg500 Raw Data (Obs. N = 6952) – NP–P36 (95_260)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

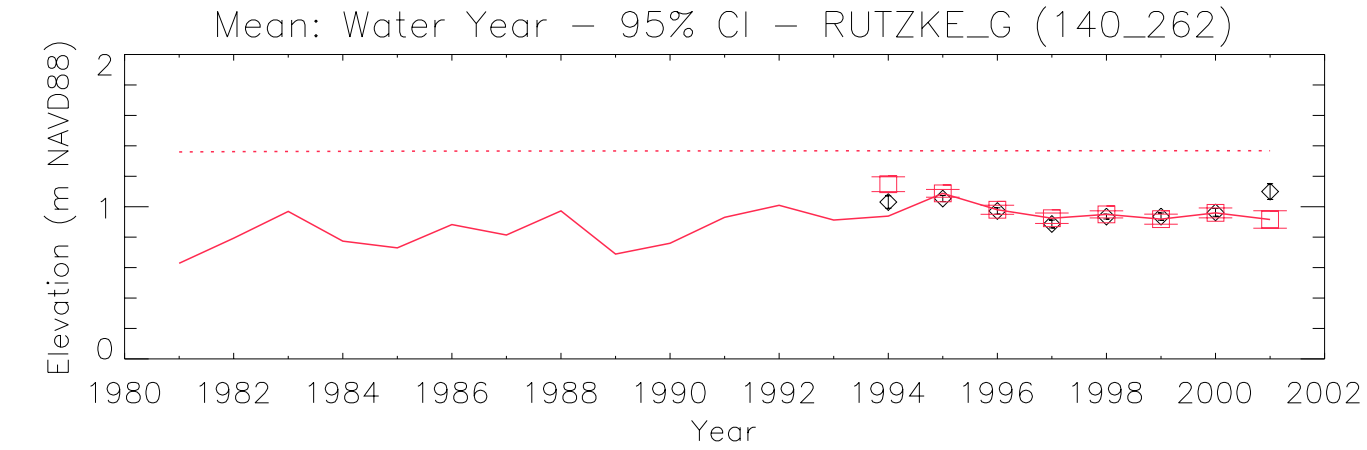

Mean: Season – 95% CI – NP–P36 (95_260)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

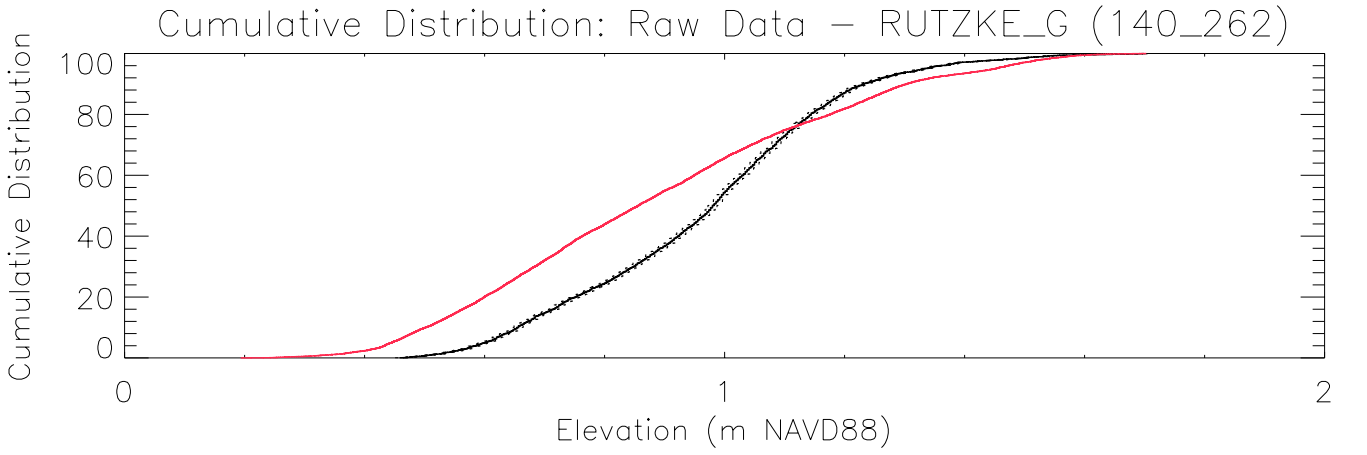

Mean: Water Year – 95% CI – NP–P36 (95_260)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

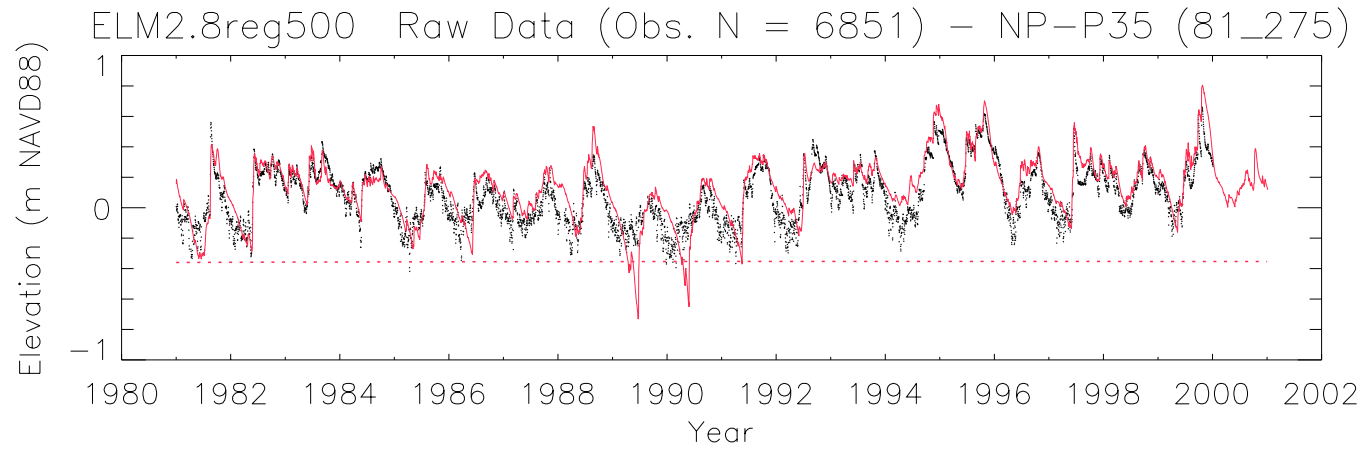

Cumulative Distribution: Raw Data – NP–P36 (95_260)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
2
Elevation (m NAVD88)

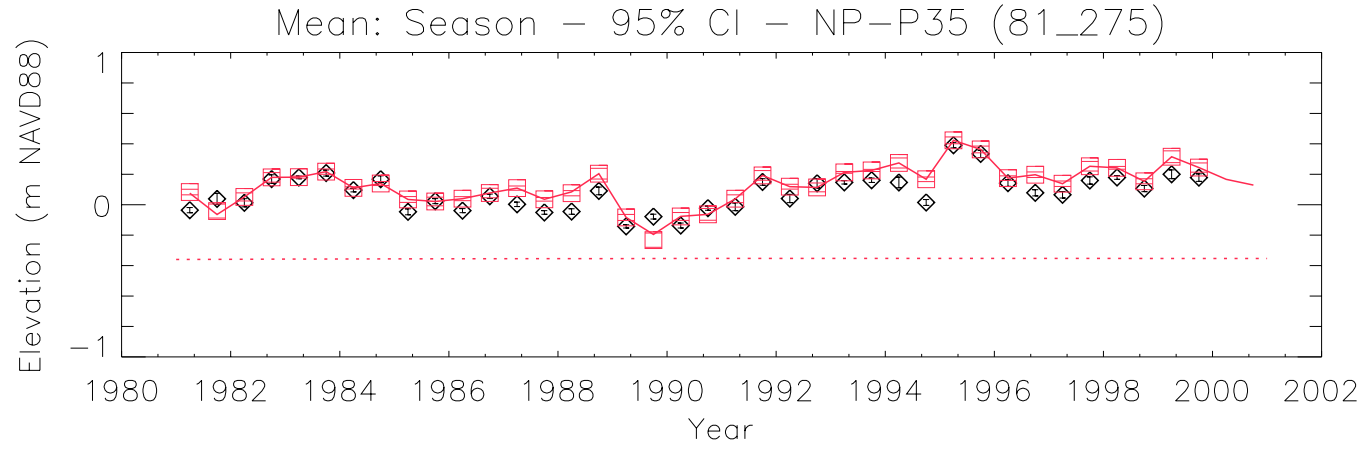

ELM2.8reg500 Raw Data (Obs. N = 2369) – RUTZKE_G (140_262)
Elevation (m NAVD88)
0
1
2
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

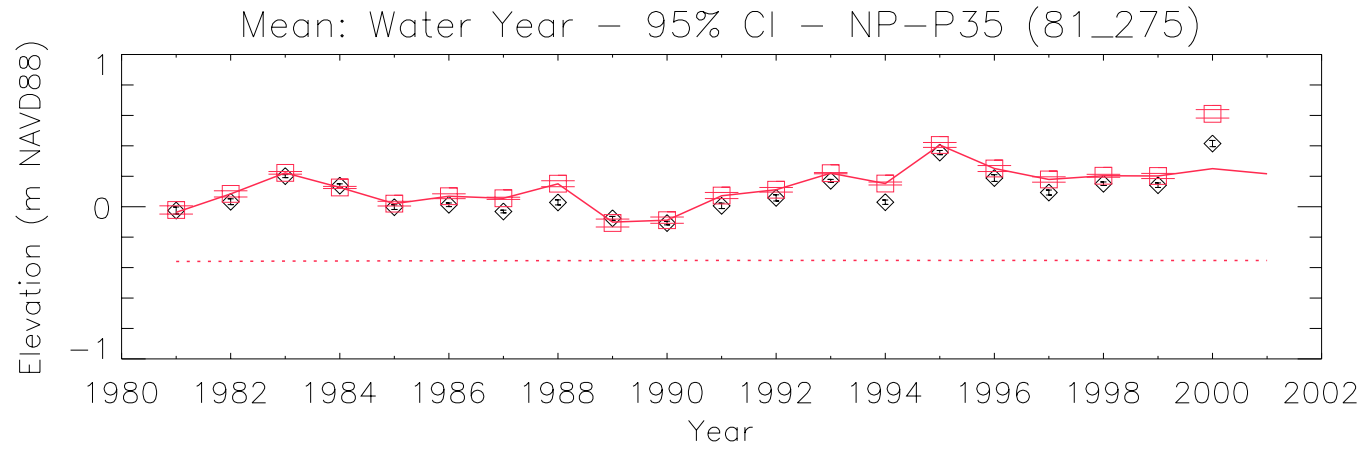

Mean: Season – 95% CI – RUTZKE_G (140_262)
Elevation (m NAVD88)
0
1
2
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

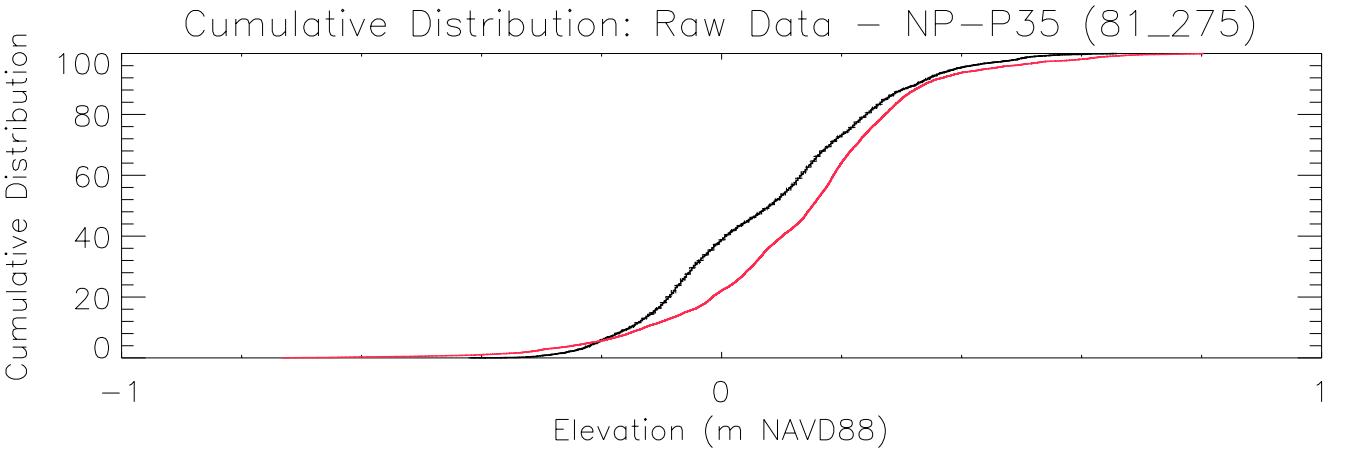

Mean: Water Year – 95% CI – RUTZKE_G (140_262)
Elevation (m NAVD88)
0
1
2
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
Year

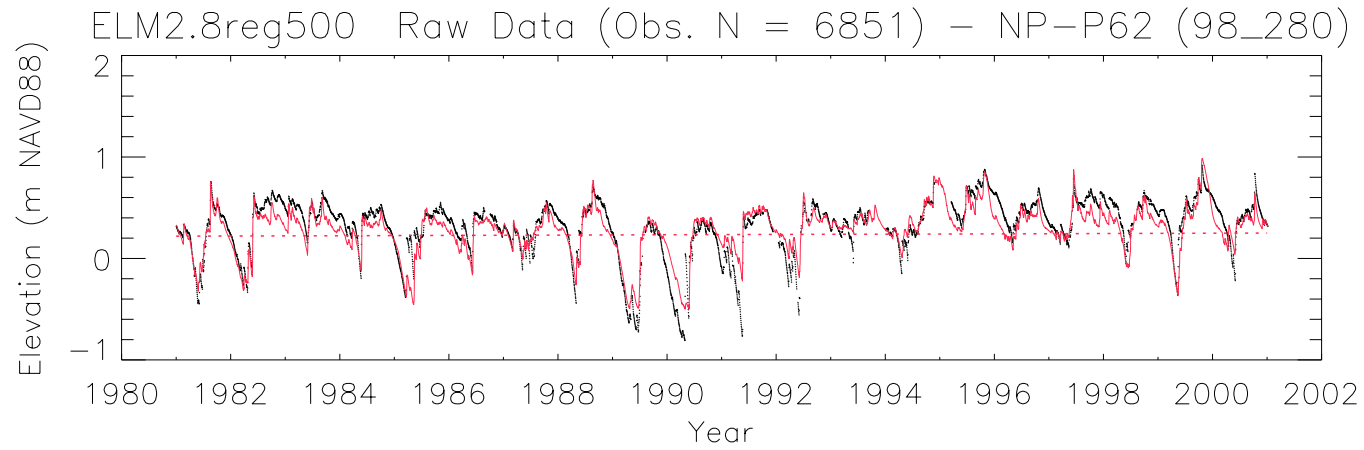

Cumulative Distribution: Raw Data – RUTZKE_G (140_262)
Cumulative Distribution
0
20
40
60
80
100
0
1
2
Elevation (m NAVD88)

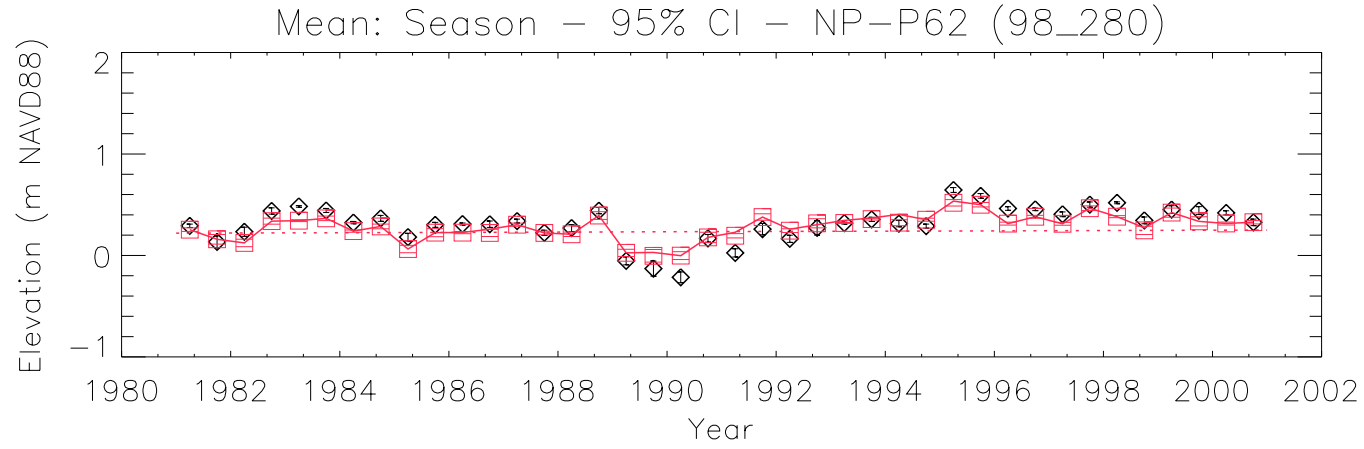

ELM2.8reg500 Raw Data (Obs. N = 6851) – NP–P35 (81_275)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

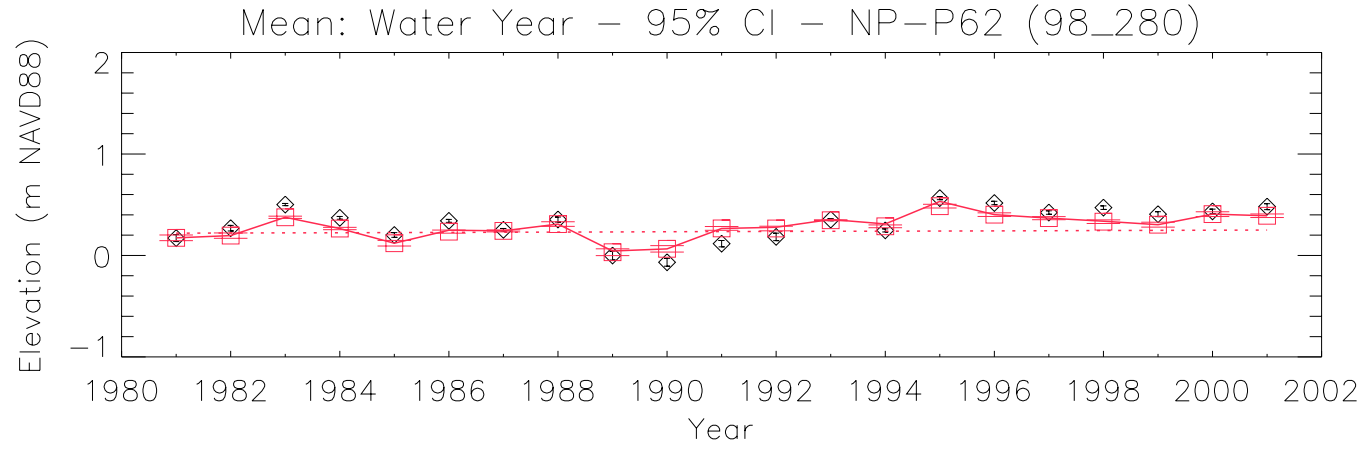

Mean: Season – 95% CI – NP–P35 (81_275)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

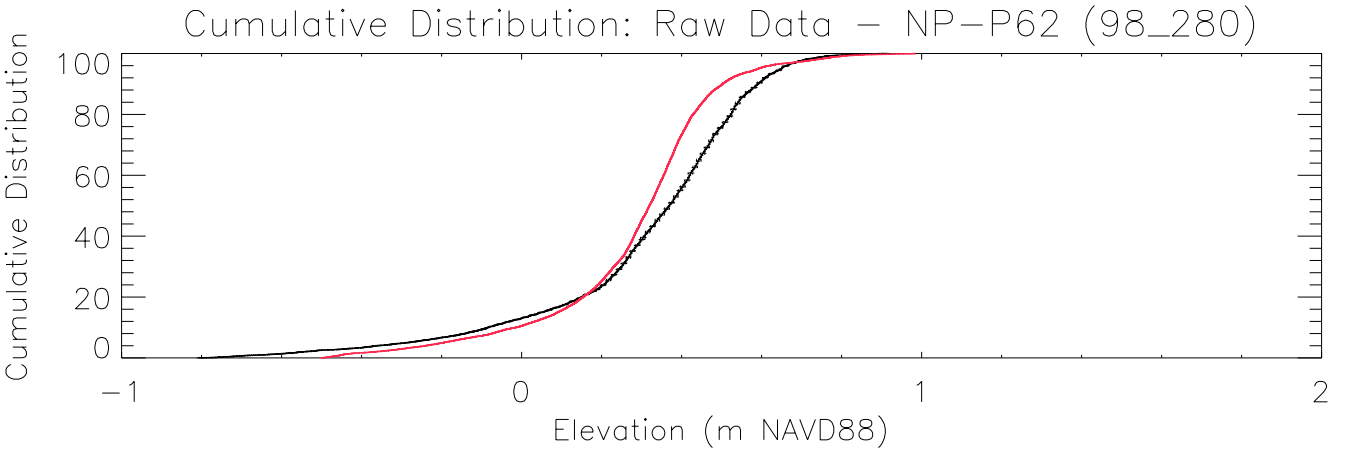

Mean: Water Year – 95% CI – NP–P35 (81_275)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

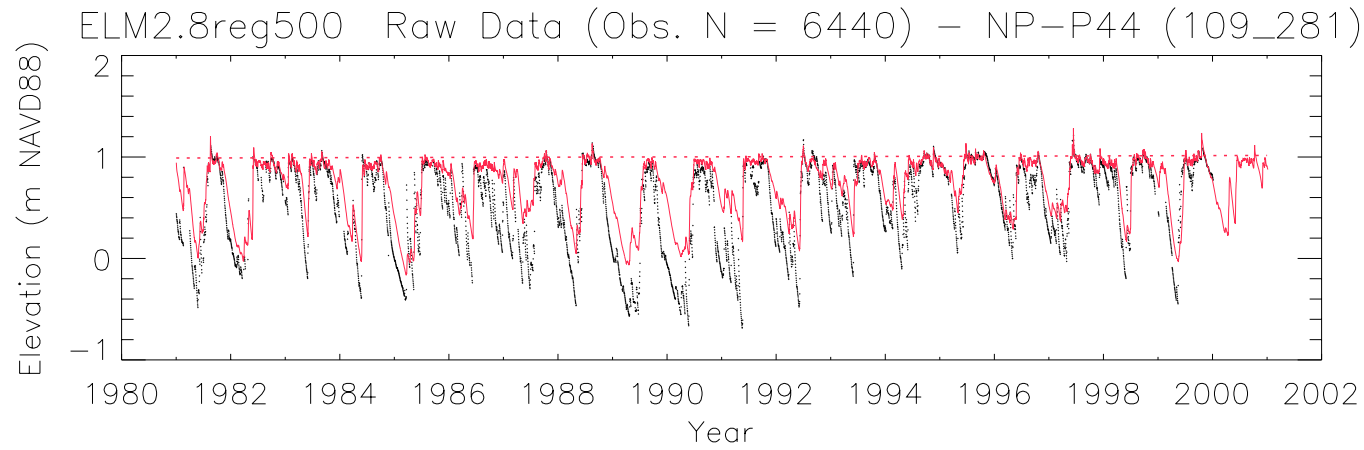

Cumulative Distribution: Raw Data – NP–P35 (81_275)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
Elevation (m NAVD88)

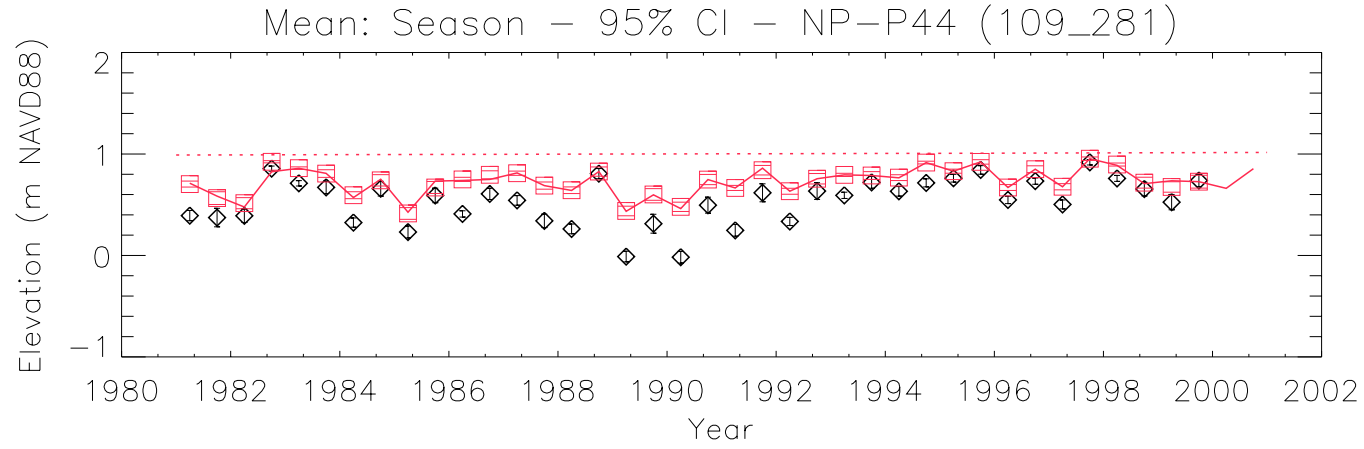

ELM2.8reg500 Raw Data (Obs. N = 6851) – NP–P62 (98_280)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

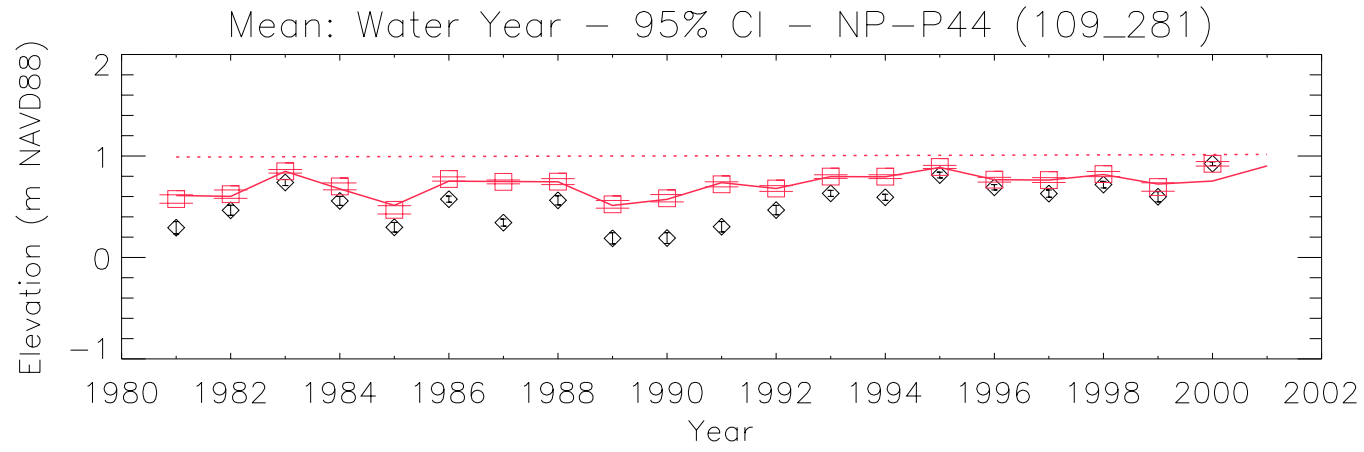

Mean: Season – 95% CI – NP–P62 (98_280)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

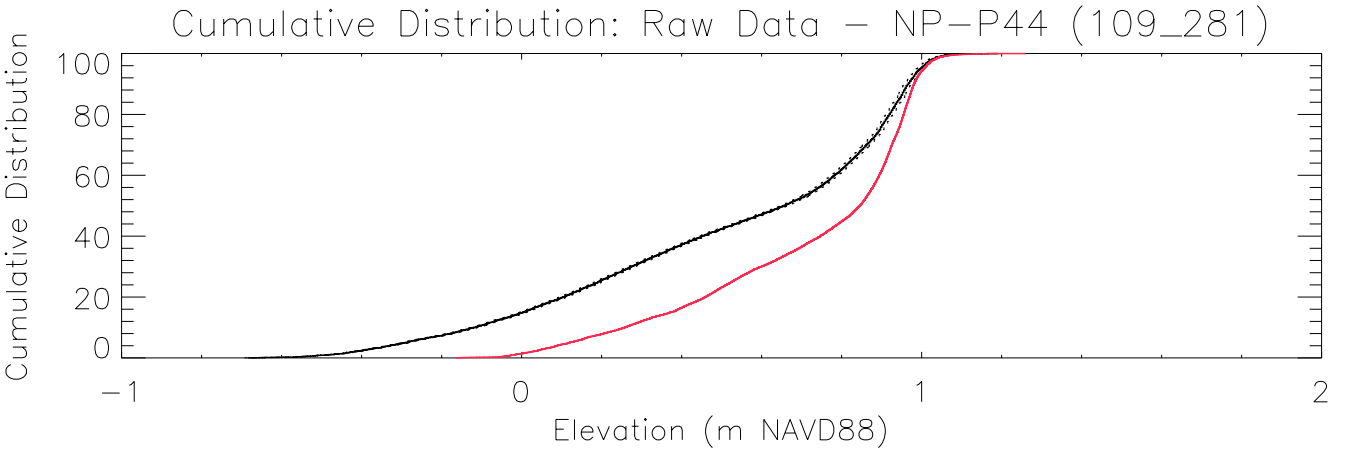

Mean: Water Year – 95% CI – NP–P62 (98_280)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

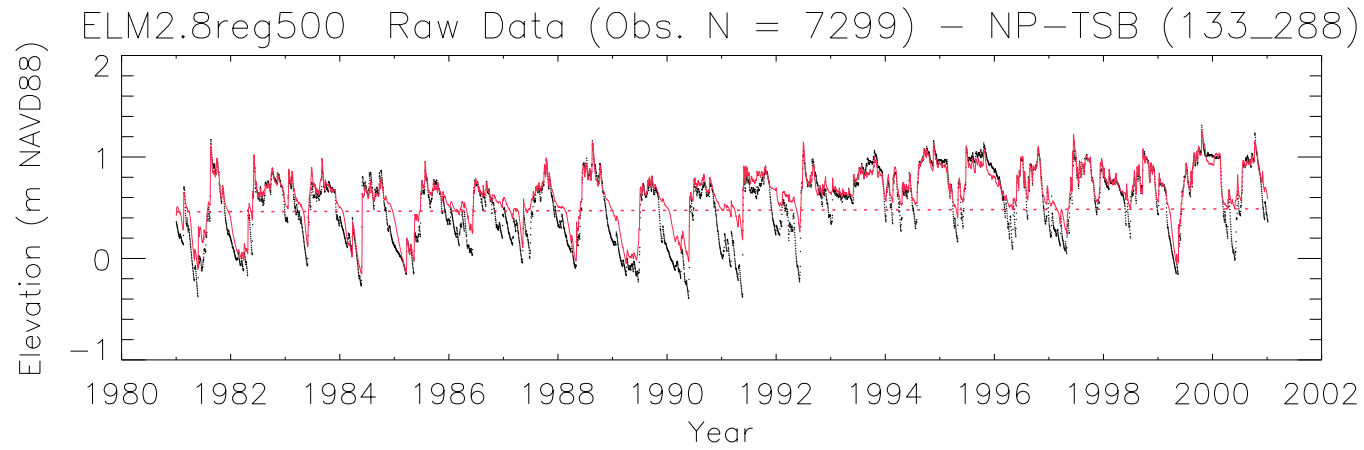

Cumulative Distribution: Raw Data – NP–P62 (98_280)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
2
Elevation (m NAVD88)

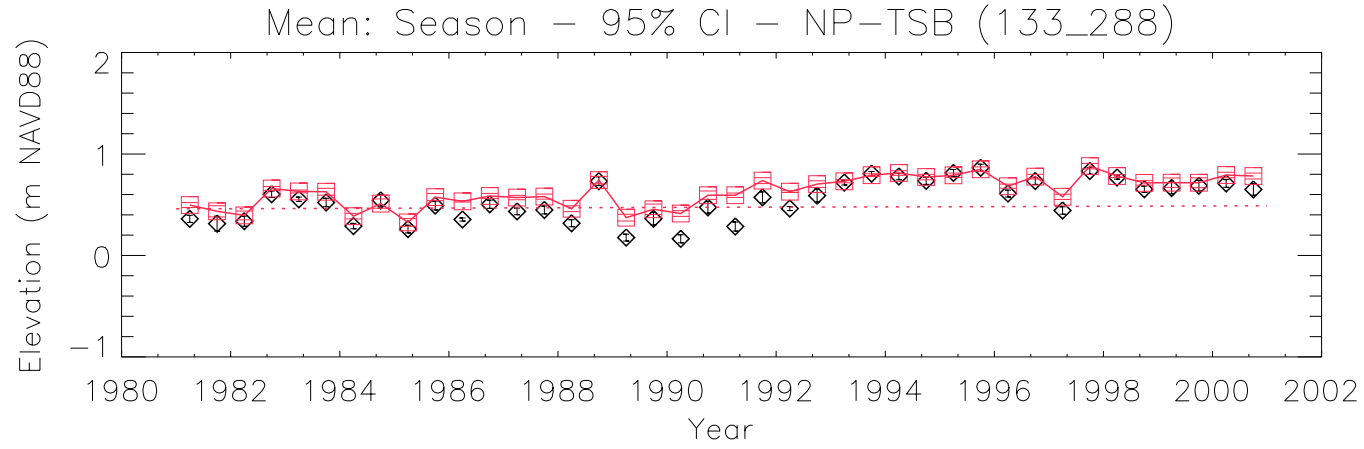

ELM2.8reg500 Raw Data (Obs. N = 6440) – NP–P44 (109_281)
Elevation (m NAVD88)
2
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

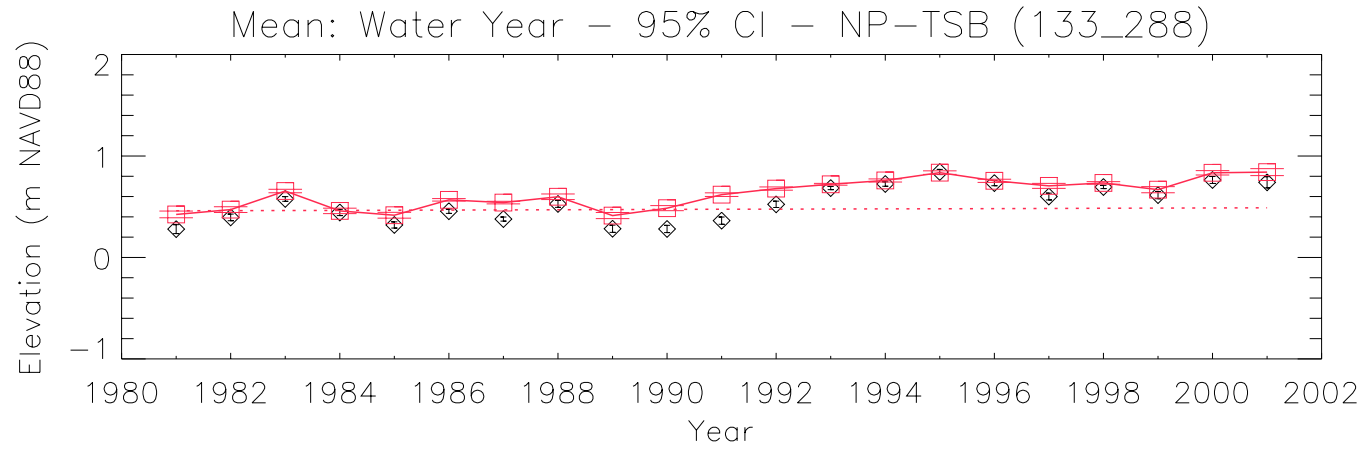

Mean: Season – 95% CI – NP–P44 (109_281)
Elevation (m NAVD88)
2
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

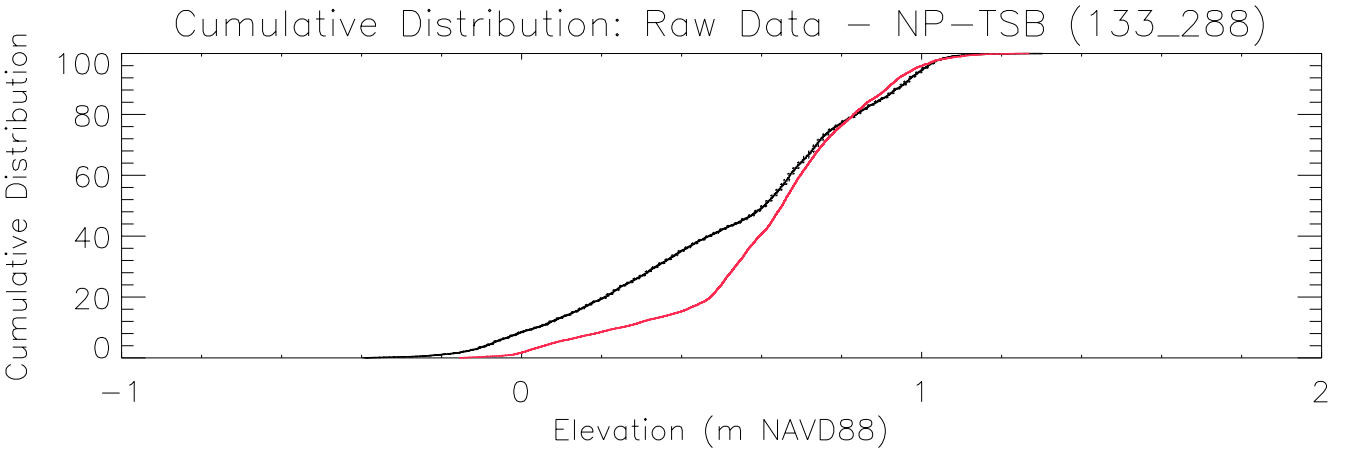

Mean: Water Year – 95% CI – NP–P44 (109_281)
Elevation (m NAVD88)
2
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

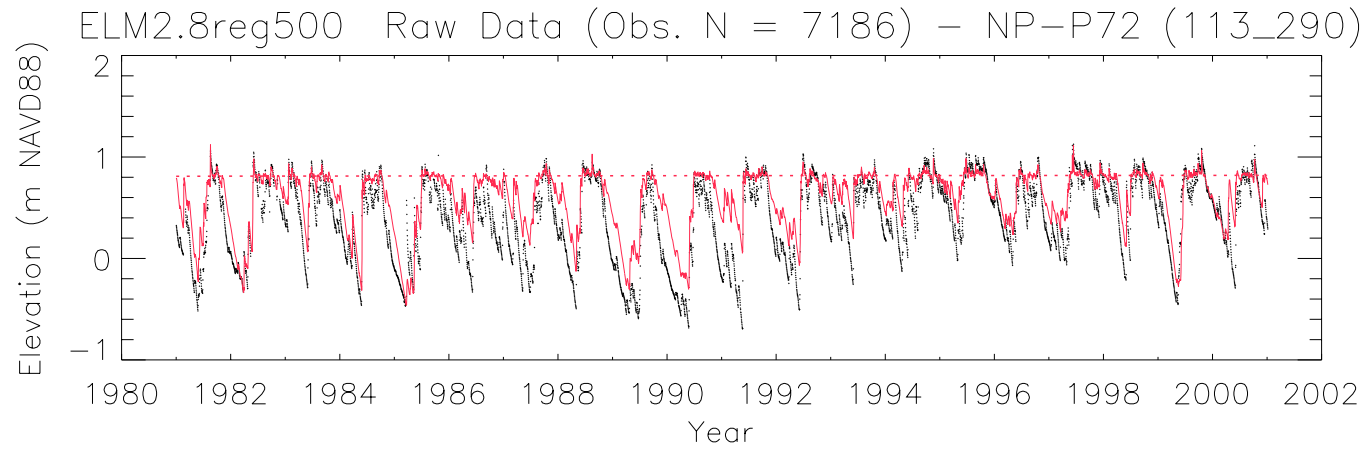

Cumulative Distribution: Raw Data – NP–P44 (109_281)
Cumulative Distribution
100
80
60
40
20
0
−1
0
1
2
Elevation (m NAVD88)

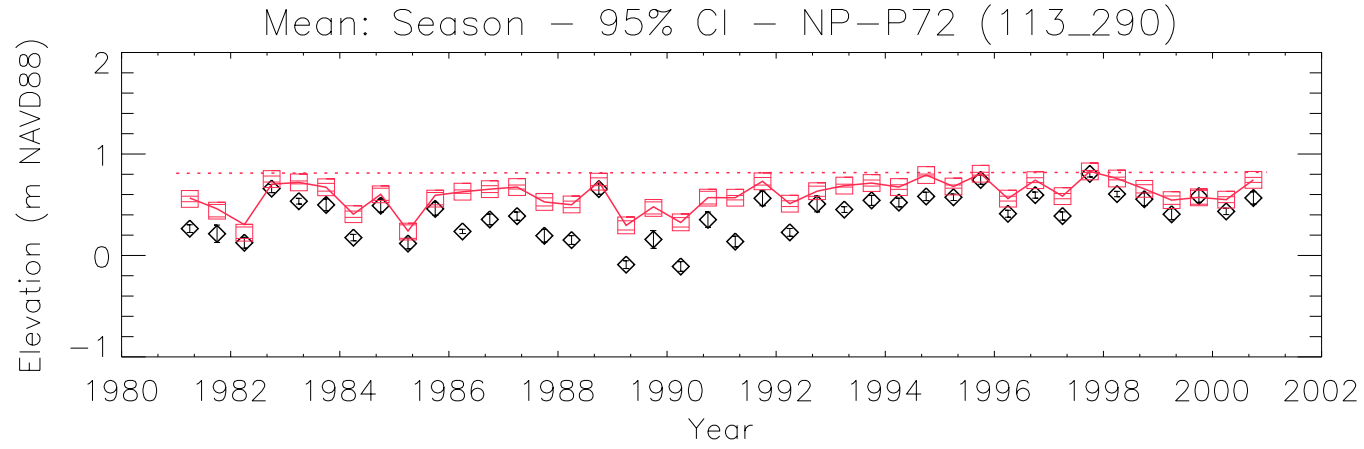

ELM2.8reg500 Raw Data (Obs. N = 7299) – NP–TSB (133_288)
Elevation (m NAVD88)
2
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

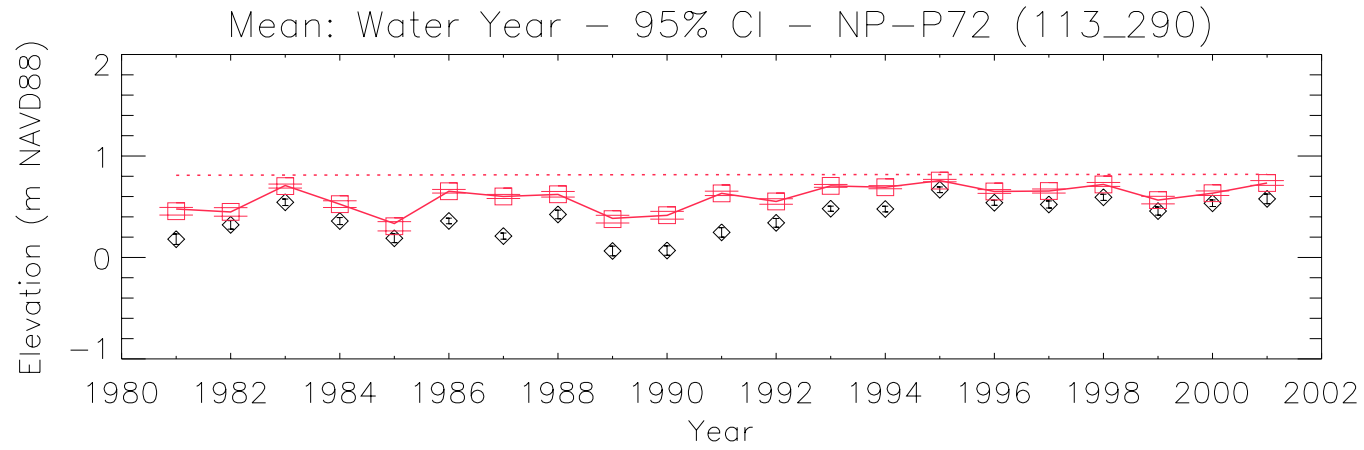

Mean: Season – 95% CI – NP–TSB (133_288)
Elevation (m NAVD88)
2
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

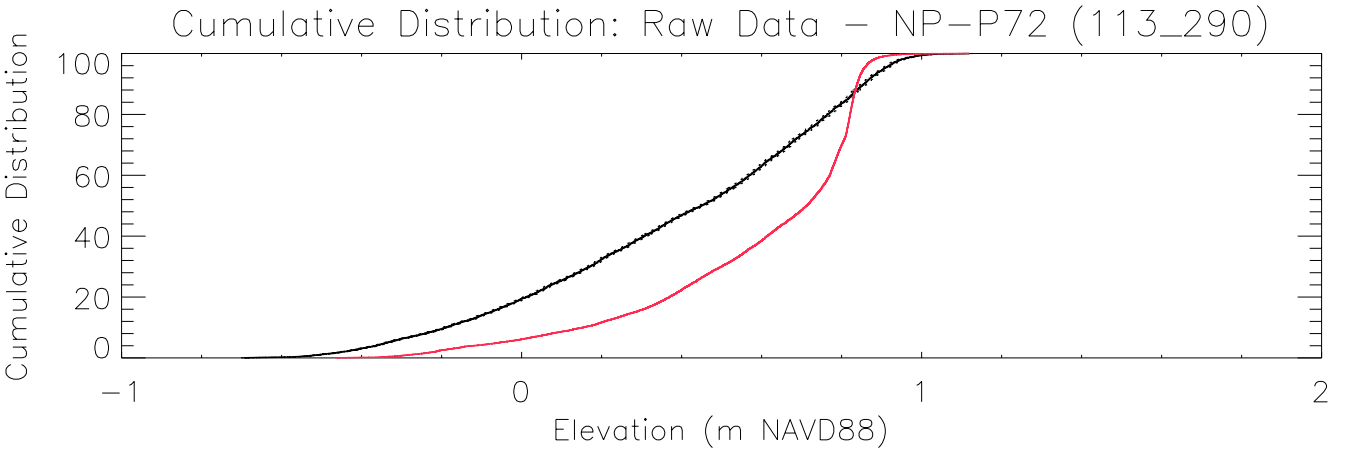

Mean: Water Year – 95% CI – NP–TSB (133_288)
Elevation (m NAVD88)
2
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

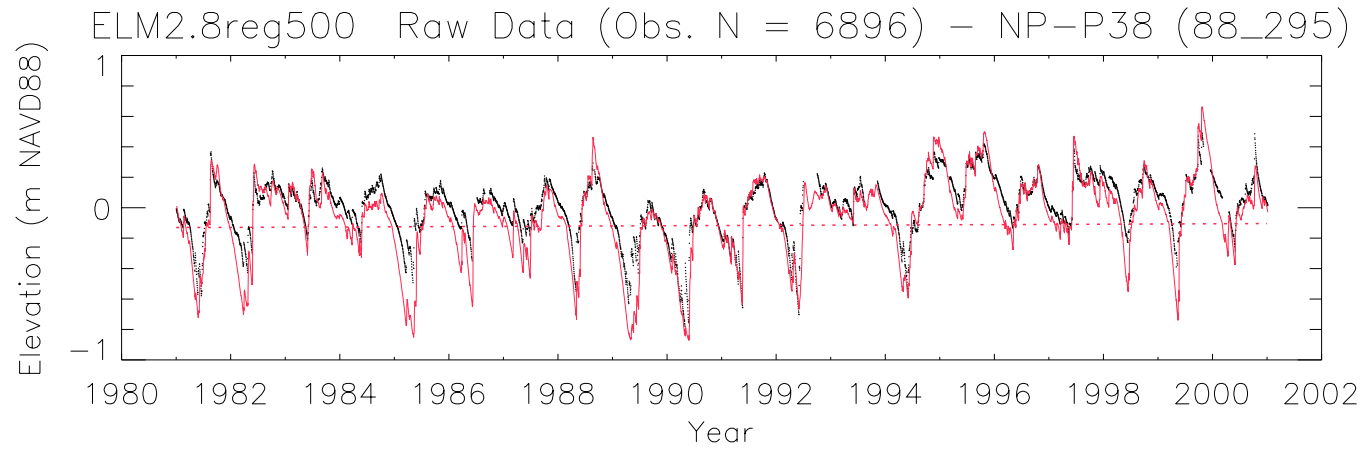

Cumulative Distribution: Raw Data – NP–TSB (133_288)
Cumulative Distribution
100
80
60
40
20
0
−1
0
1
2
Elevation (m NAVD88)

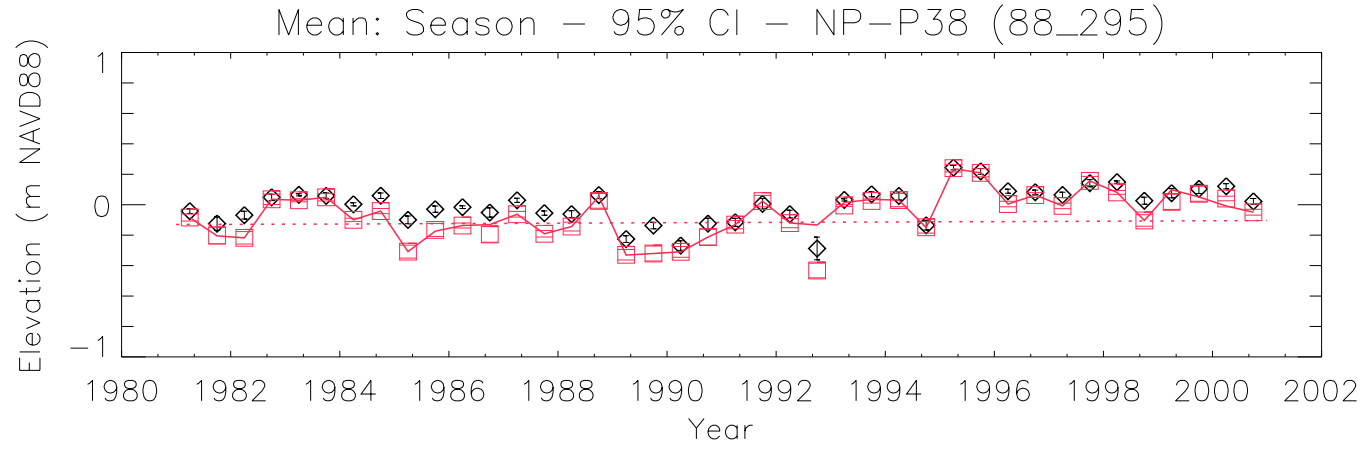

ELM2.8reg500 Raw Data (Obs. N = 7186) – NP–P72 (113_290)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

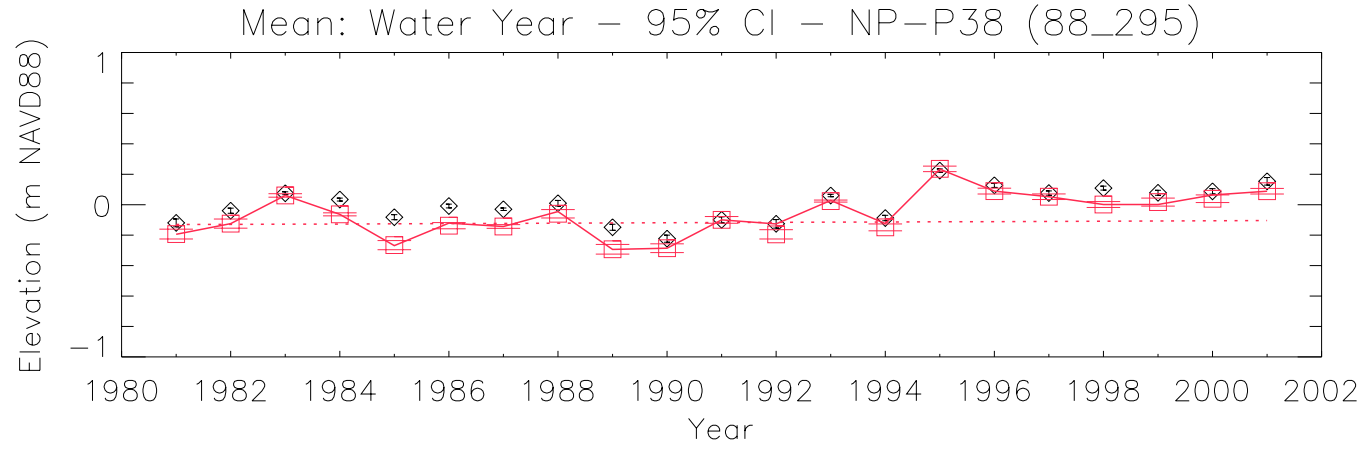

Mean: Season – 95% CI – NP–P72 (113_290)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

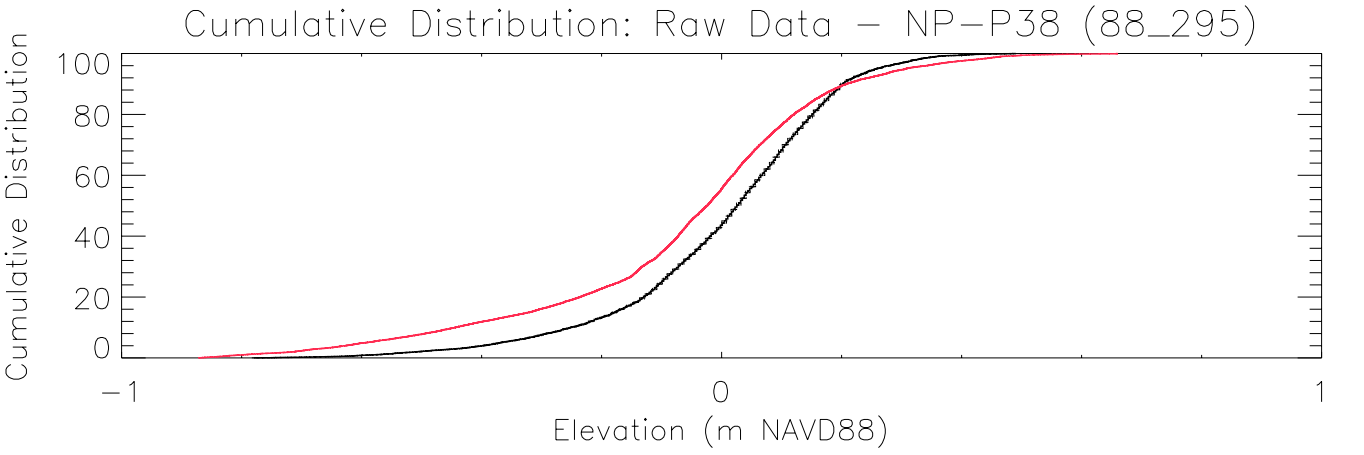

Mean: Water Year – 95% CI – NP–P72 (113_290)
Elevation (m NAVD88)
2
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

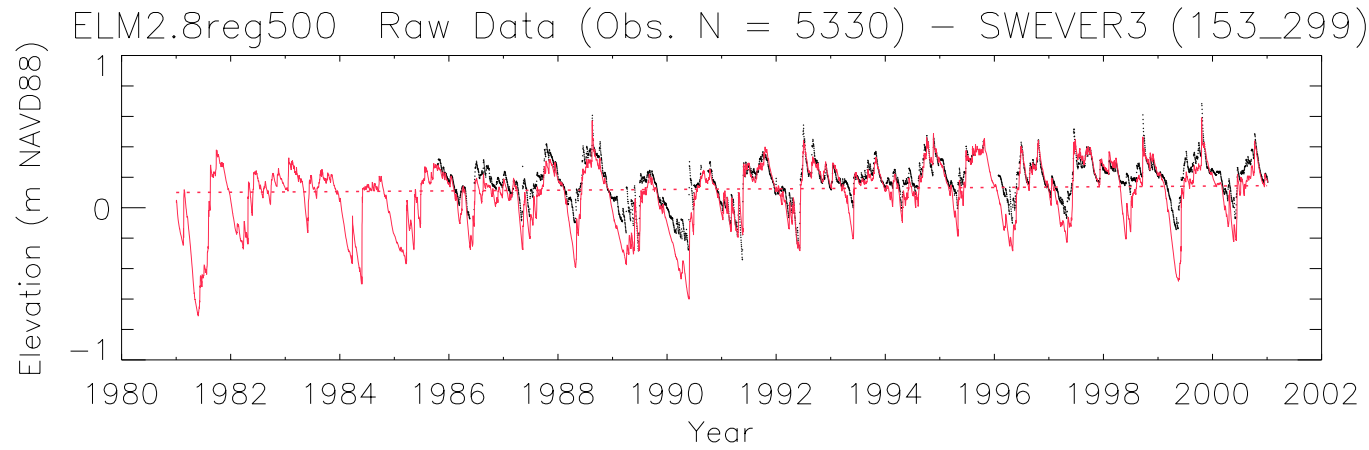

Cumulative Distribution: Raw Data – NP–P72 (113_290)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
2
Elevation (m NAVD88)

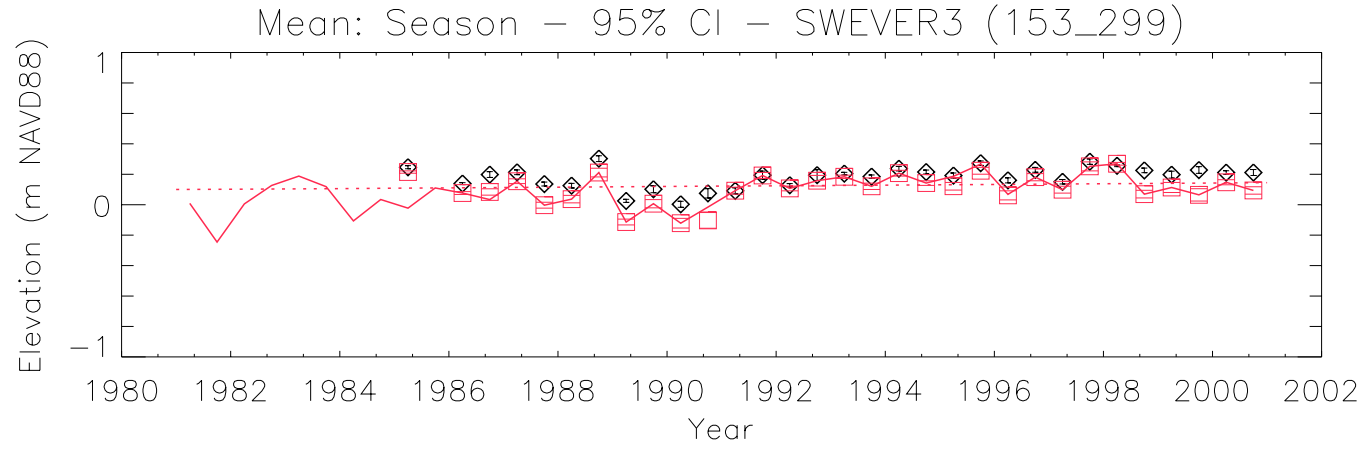

ELM2.8reg500 Raw Data (Obs. N = 6896) – NP–P38 (88_295)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

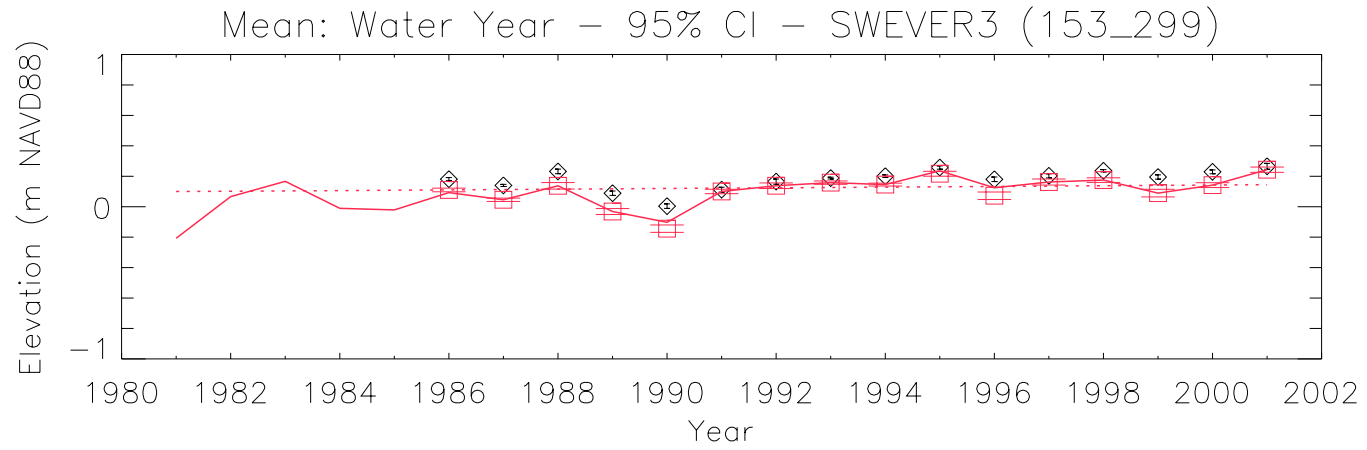

Mean: Season – 95% CI – NP–P38 (88_295)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

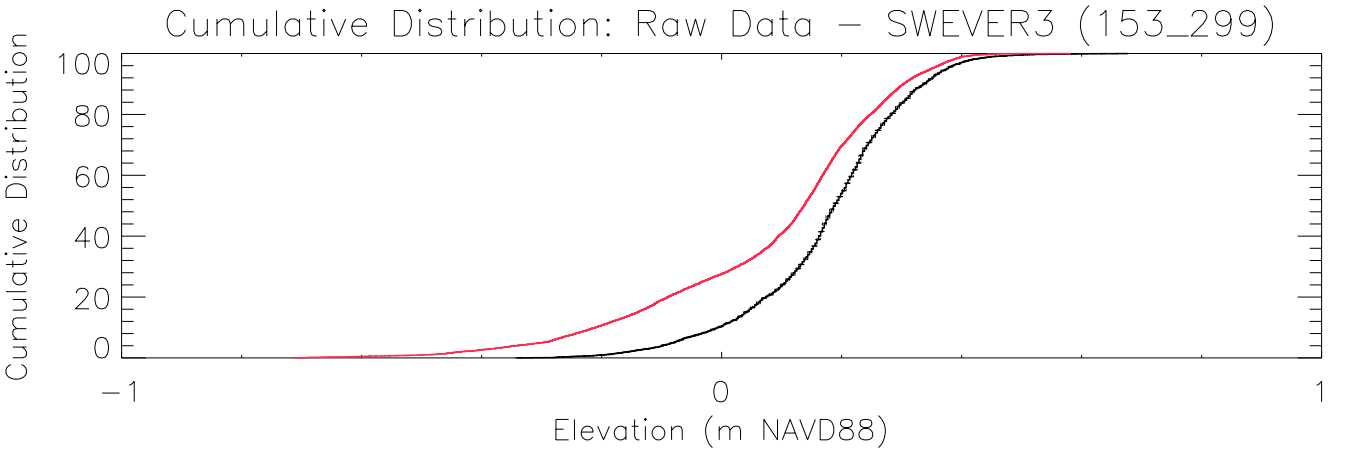

Mean: Water Year – 95% CI – NP–P38 (88_295)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

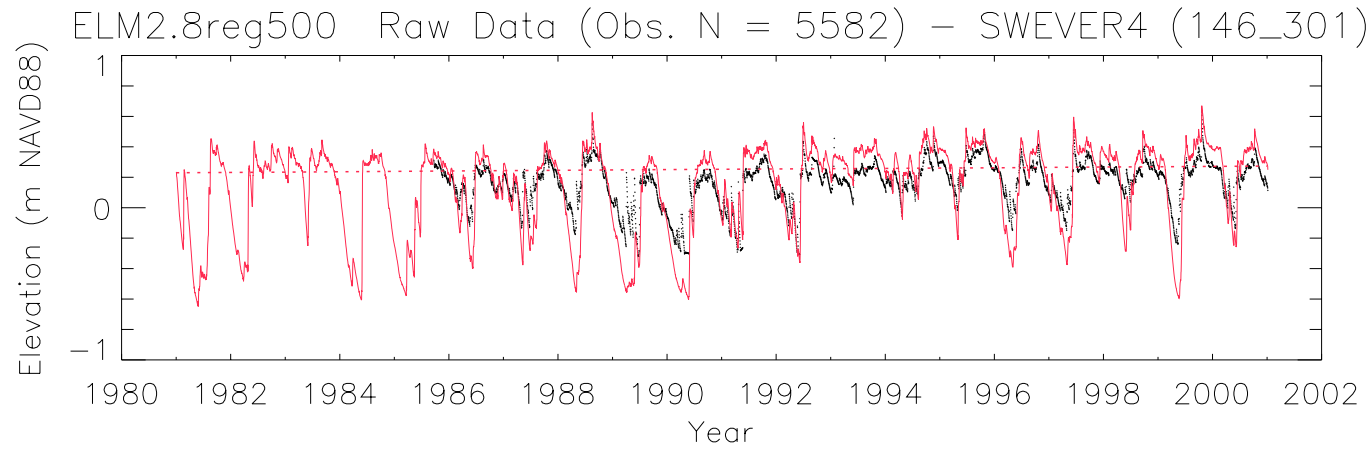

Cumulative Distribution: Raw Data – NP–P38 (88_295)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
Elevation (m NAVD88)

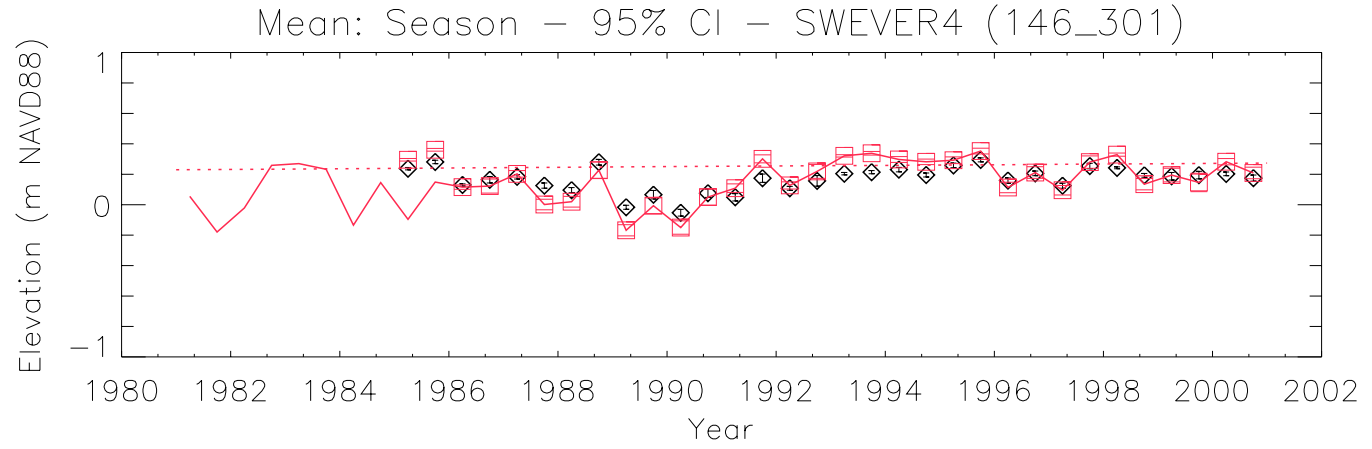

ELM2.8reg500 Raw Data (Obs. N = 5330) – SWEVER3 (153_299)
Elevation (m NAVD88)
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

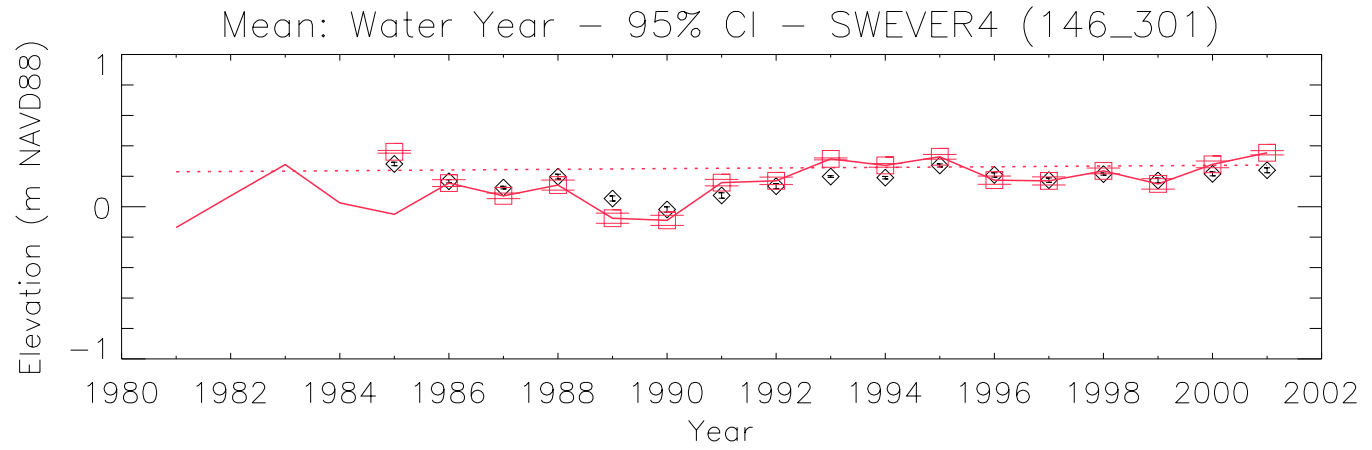

Mean: Season – 95% CI – SWEVER3 (153_299)
Elevation (m NAVD88)
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

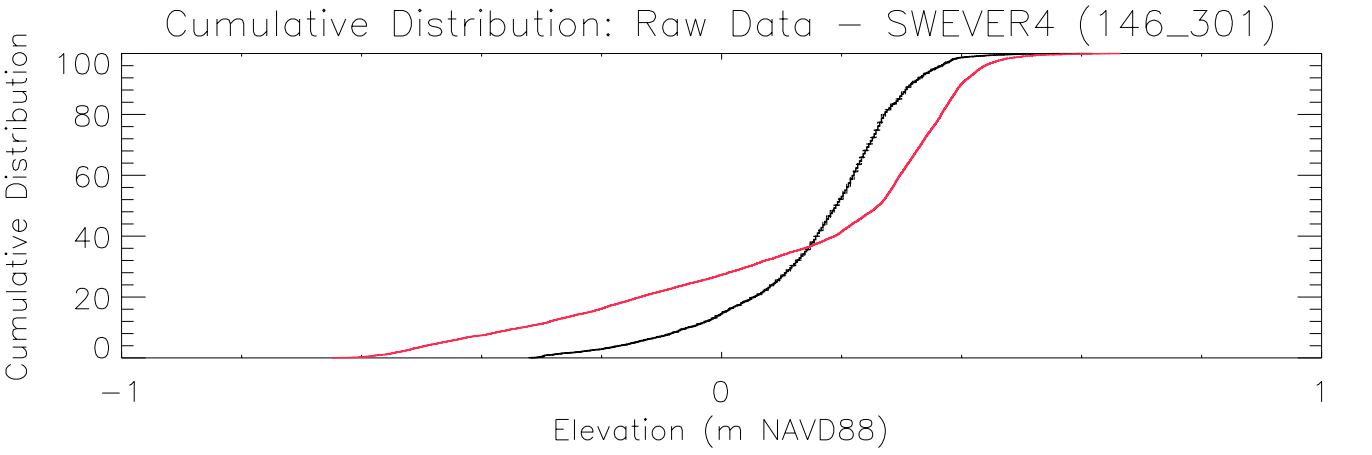

Mean: Water Year – 95% CI – SWEVER3 (153_299)
Elevation (m NAVD88)
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

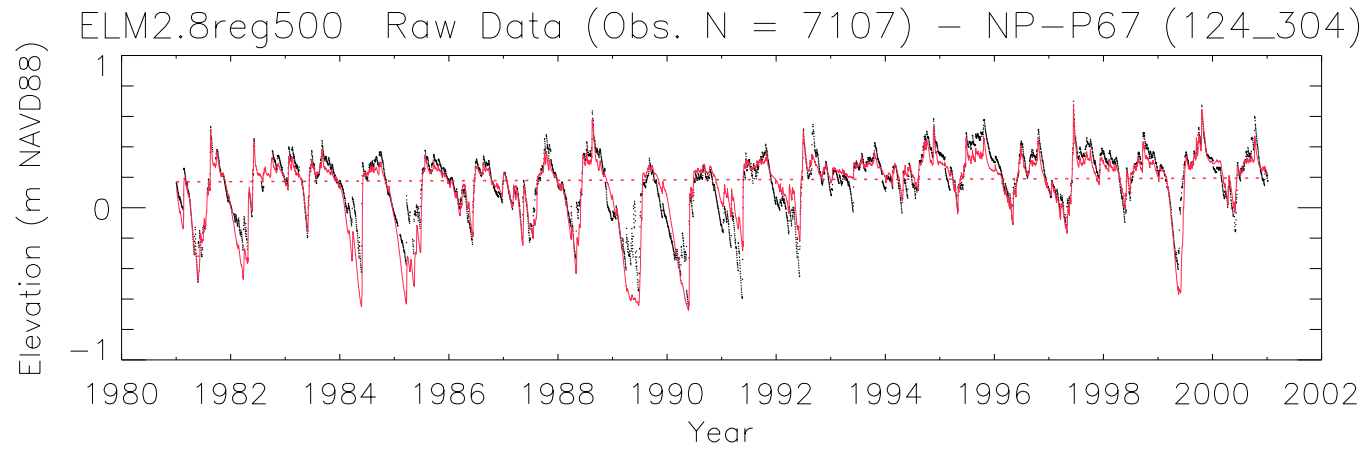

Cumulative Distribution: Raw Data – SWEVER3 (153_299)
Cumulative Distribution
100
80
60
40
20
0
−1
0
1
Elevation (m NAVD88)

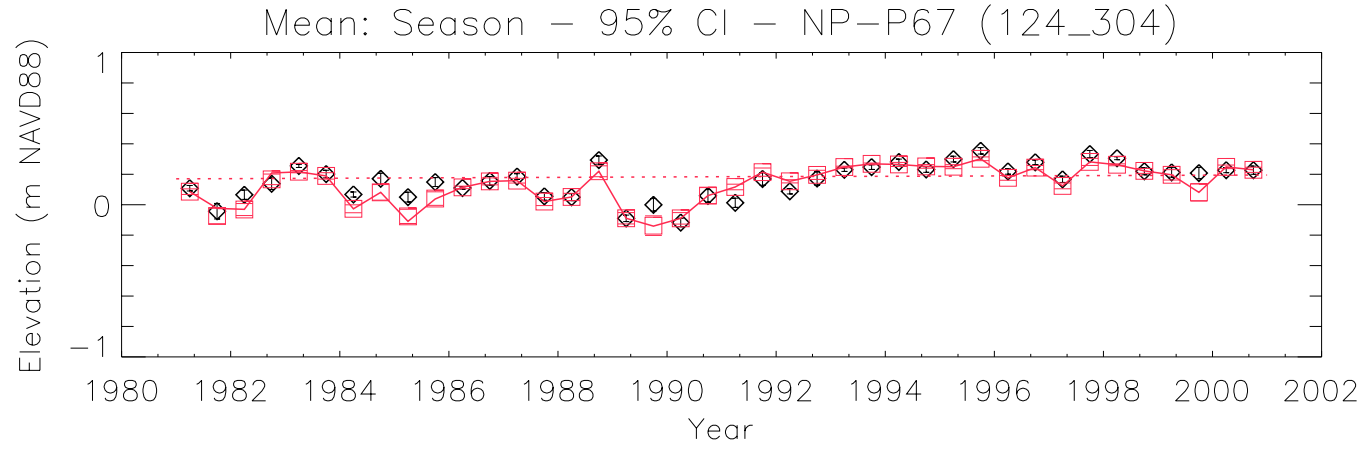

ELM2.8reg500 Raw Data (Obs. N = 5582) – SWEVER4 (146_301)
Elevation (m NAVD88)
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

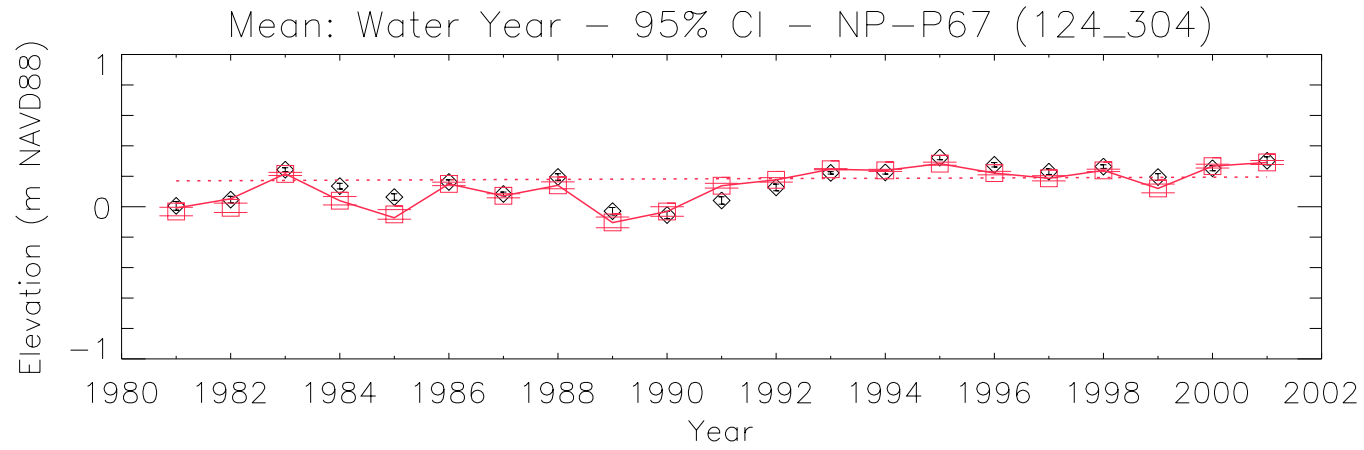

Mean: Season – 95% CI – SWEVER4 (146_301)
Elevation (m NAVD88)
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

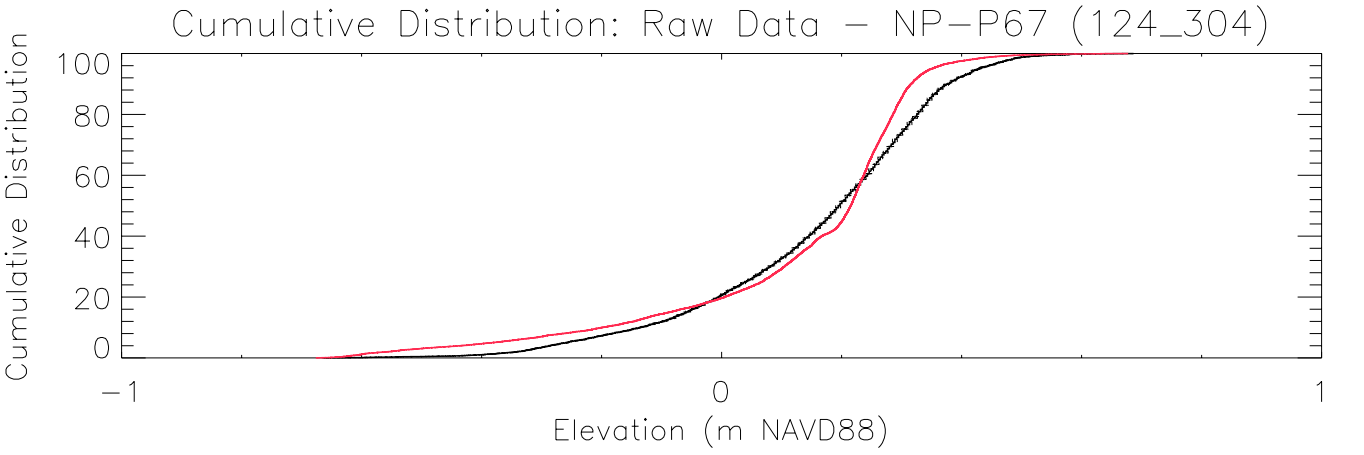

Mean: Water Year – 95% CI – SWEVER4 (146_301)
Elevation (m NAVD88)
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

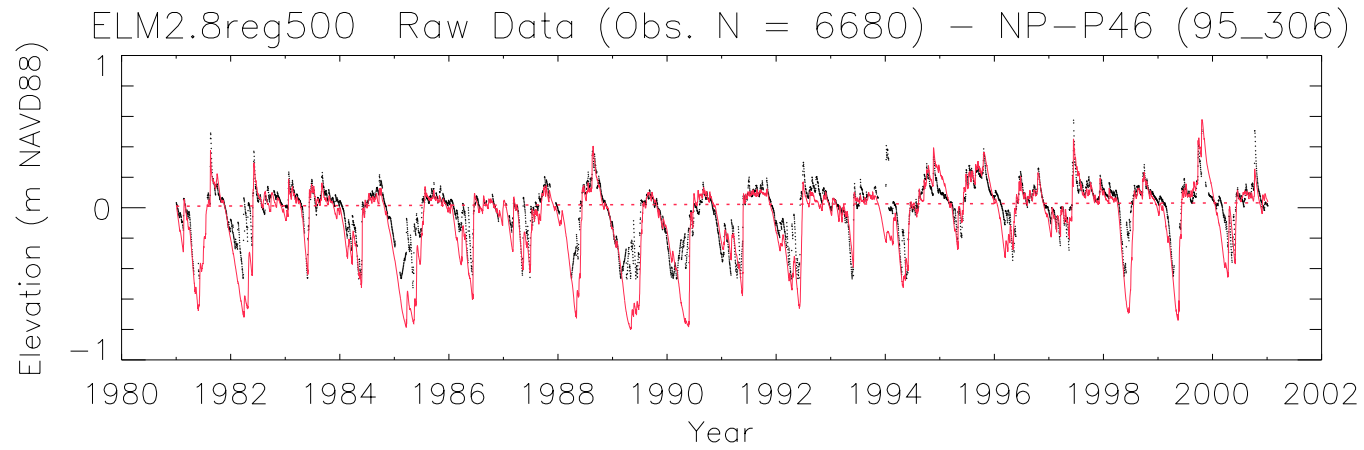

Cumulative Distribution: Raw Data – SWEVER4 (146_301)
Cumulative Distribution
100
80
60
40
20
0
−1
0
1
Elevation (m NAVD88)

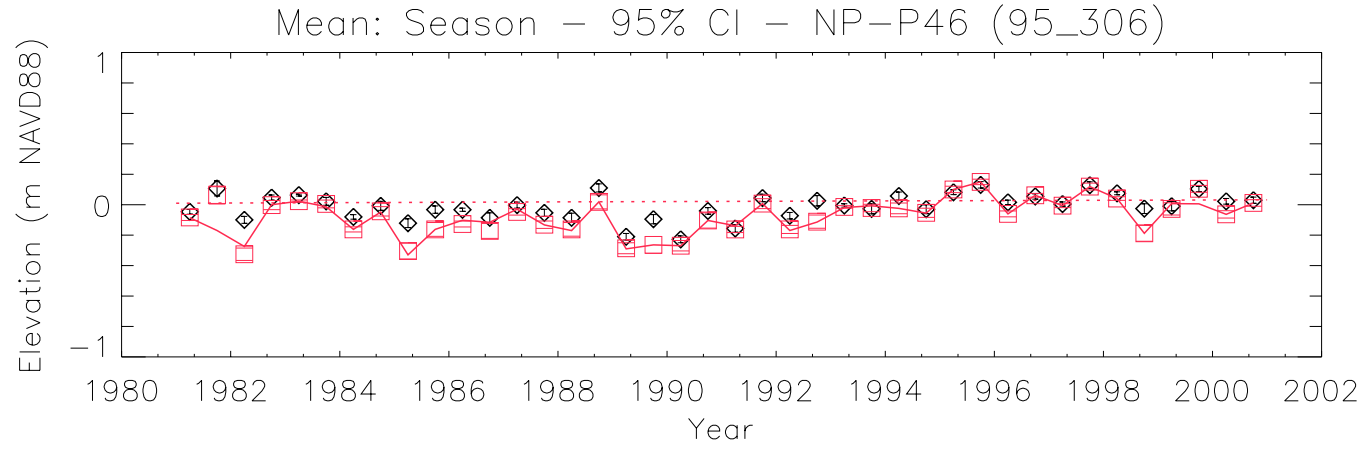

ELM2.8reg500 Raw Data (Obs. N = 7107) – NP–P67 (124_304)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

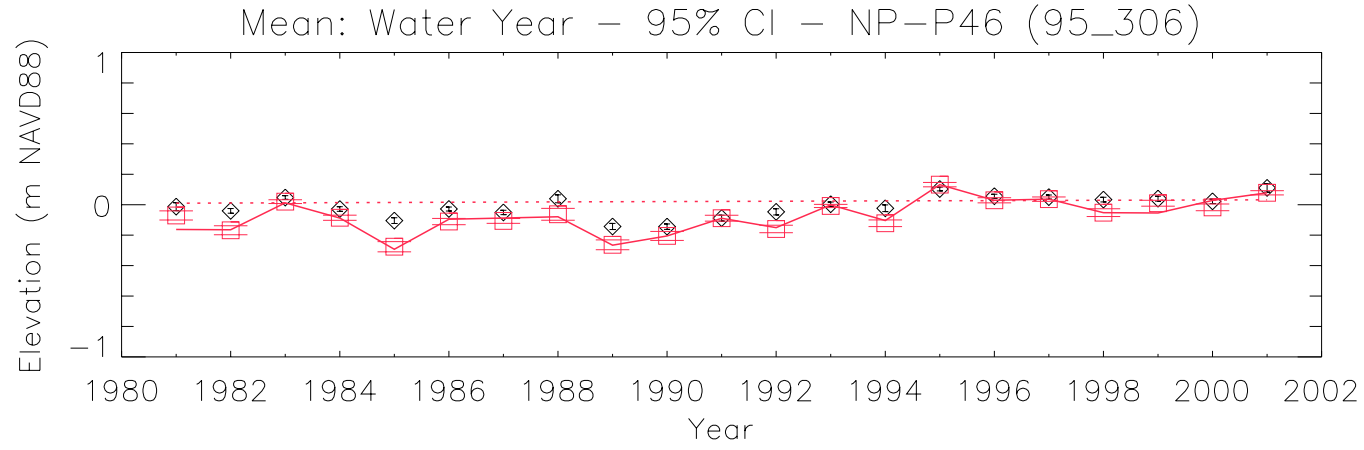

Mean: Season – 95% CI – NP–P67 (124_304)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

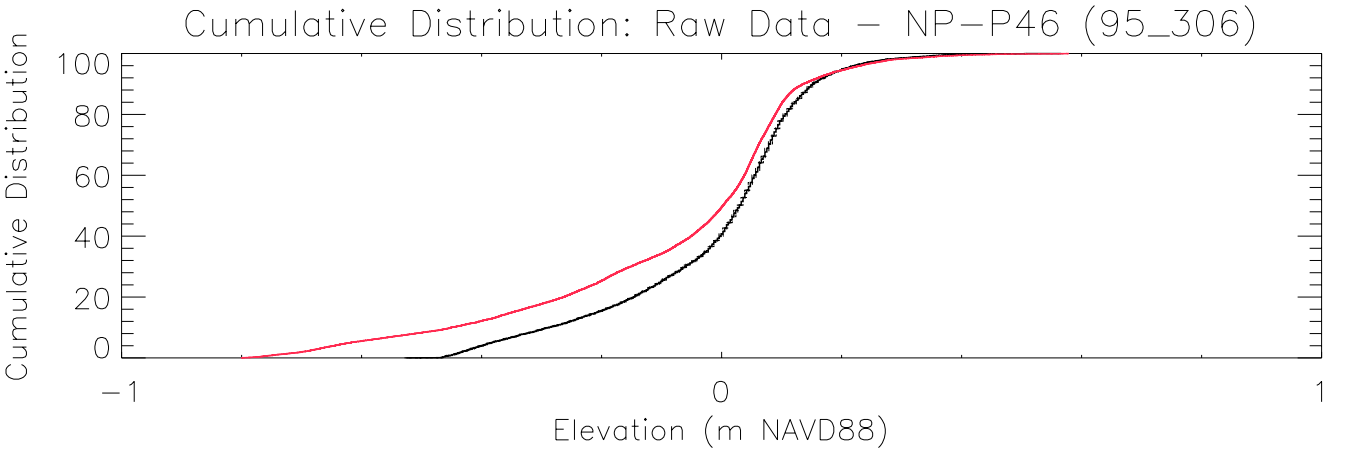

Mean: Water Year – 95% CI – NP–P67 (124_304)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

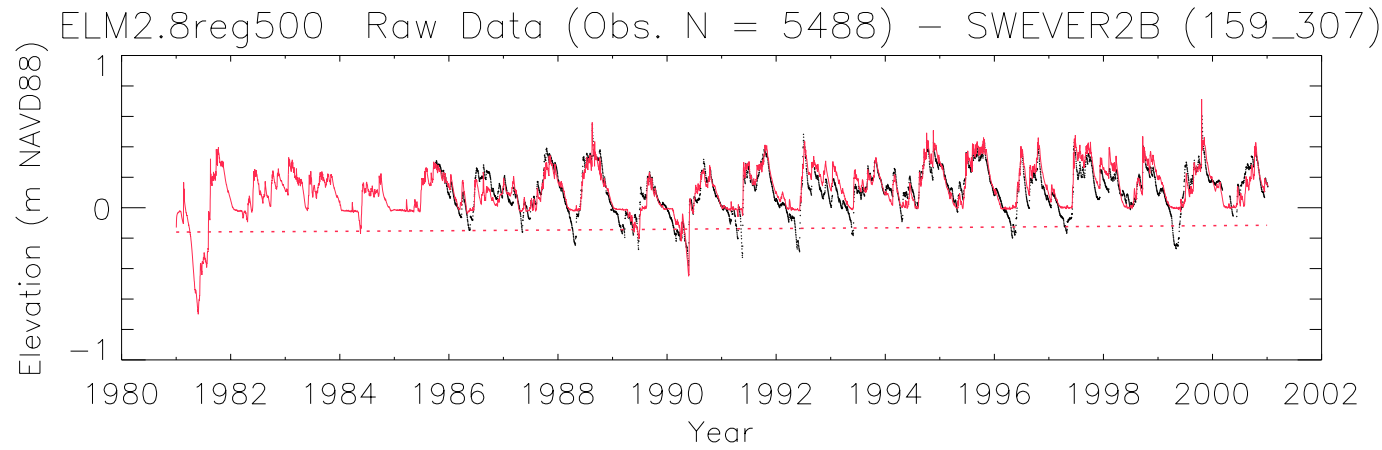

Cumulative Distribution: Raw Data – NP–P67 (124_304)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
Elevation (m NAVD88)

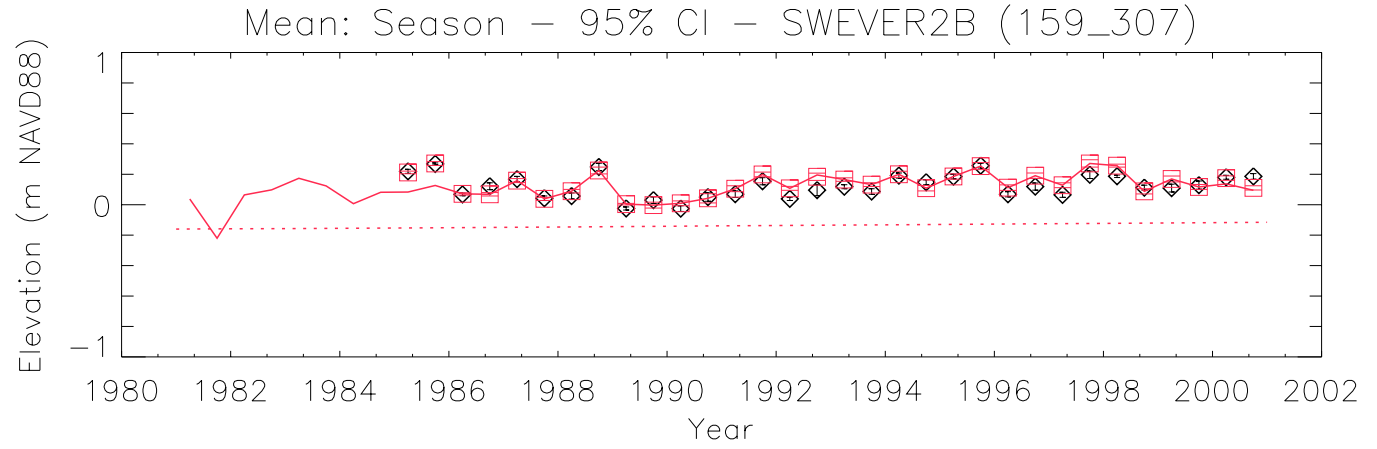

ELM2.8reg500 Raw Data (Obs. N = 6680) – NP–P46 (95_306)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

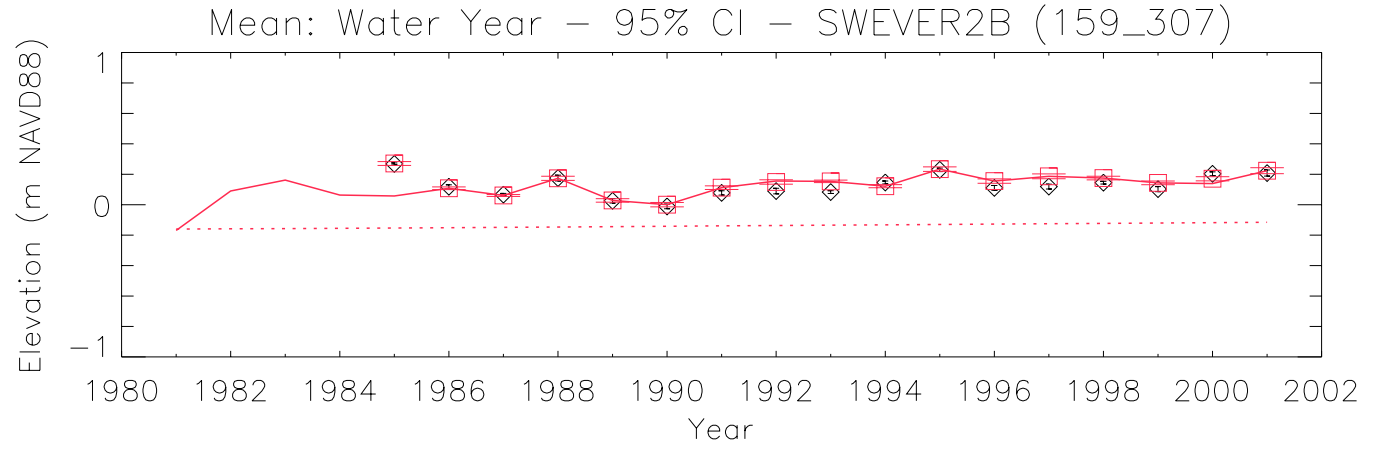

Mean: Season – 95% CI – NP–P46 (95_306)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

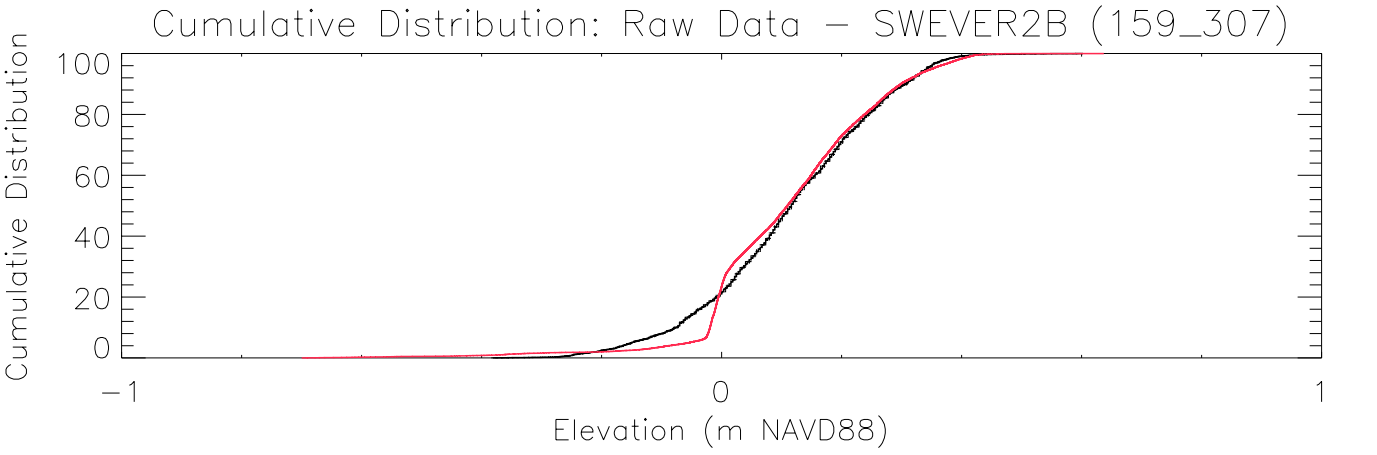

Mean: Water Year – 95% CI – NP–P46 (95_306)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

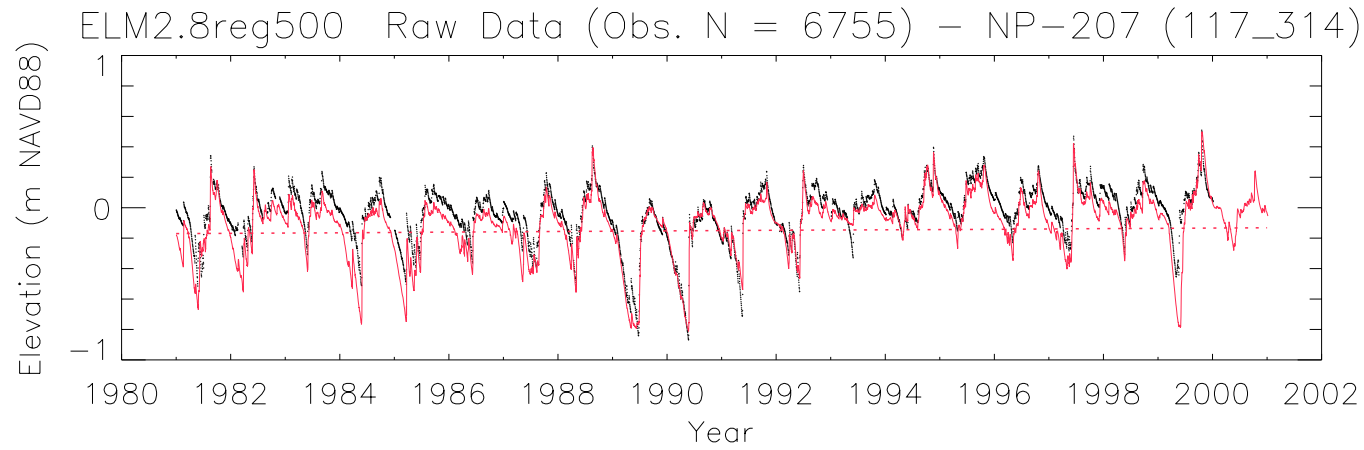

Cumulative Distribution: Raw Data – NP–P46 (95_306)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
Elevation (m NAVD88)

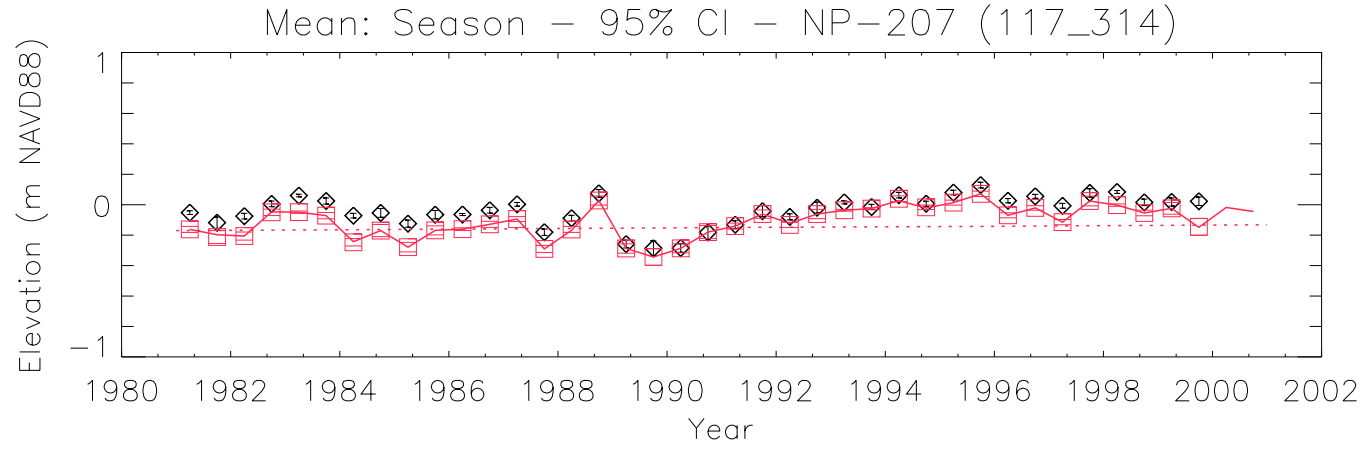

ELM2.8reg500 Raw Data (Obs. N = 5488) – SWEVER2B (159_307)
Elevation (m NAVD88)
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

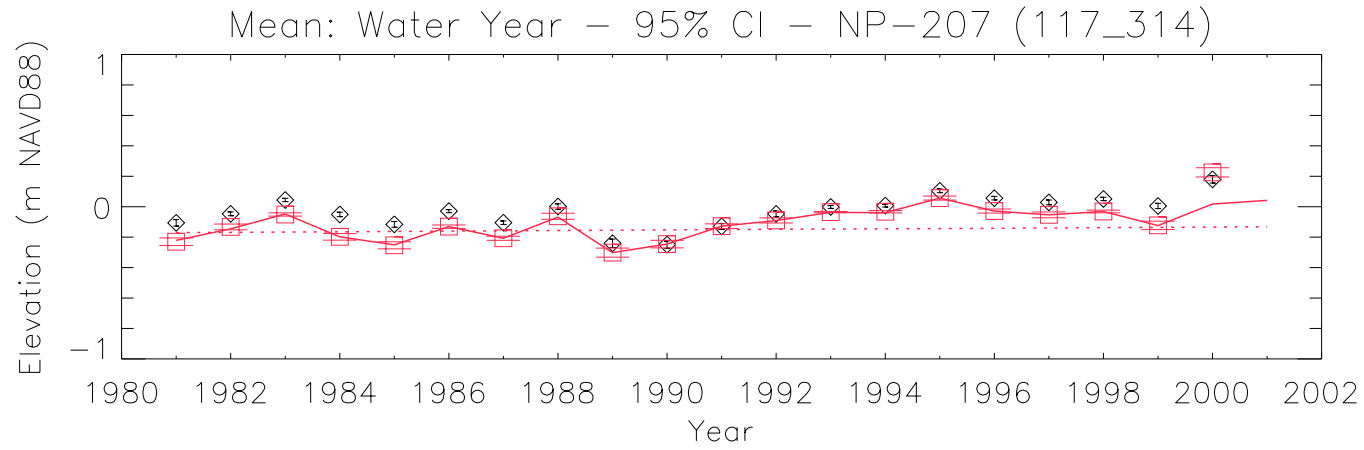

Mean: Season – 95% CI – SWEVER2B (159_307)
Elevation (m NAVD88)
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

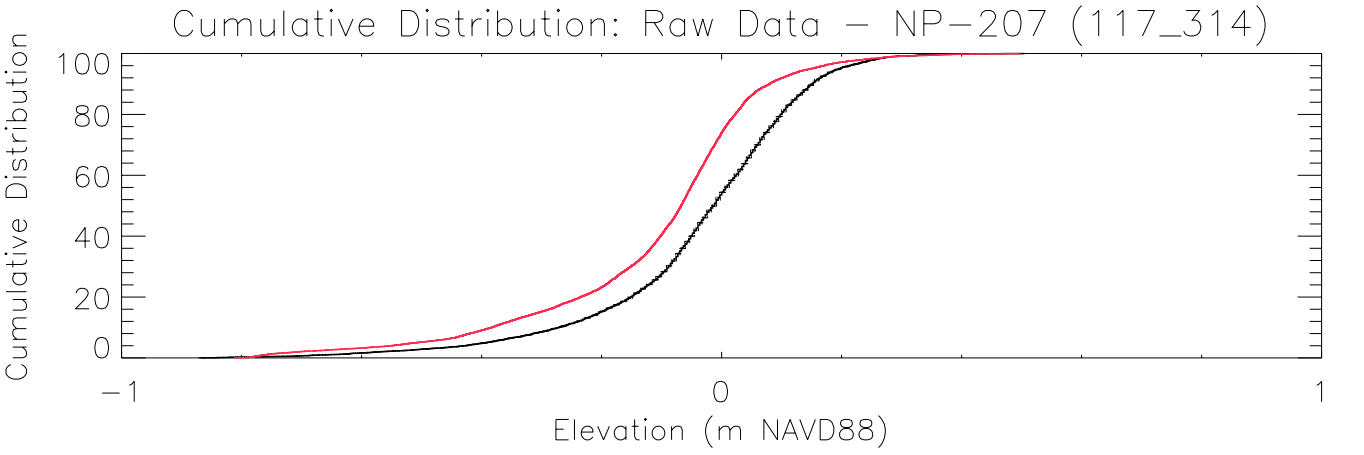

Mean: Water Year – 95% CI – SWEVER2B (159_307)
Elevation (m NAVD88)
1
0
−1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

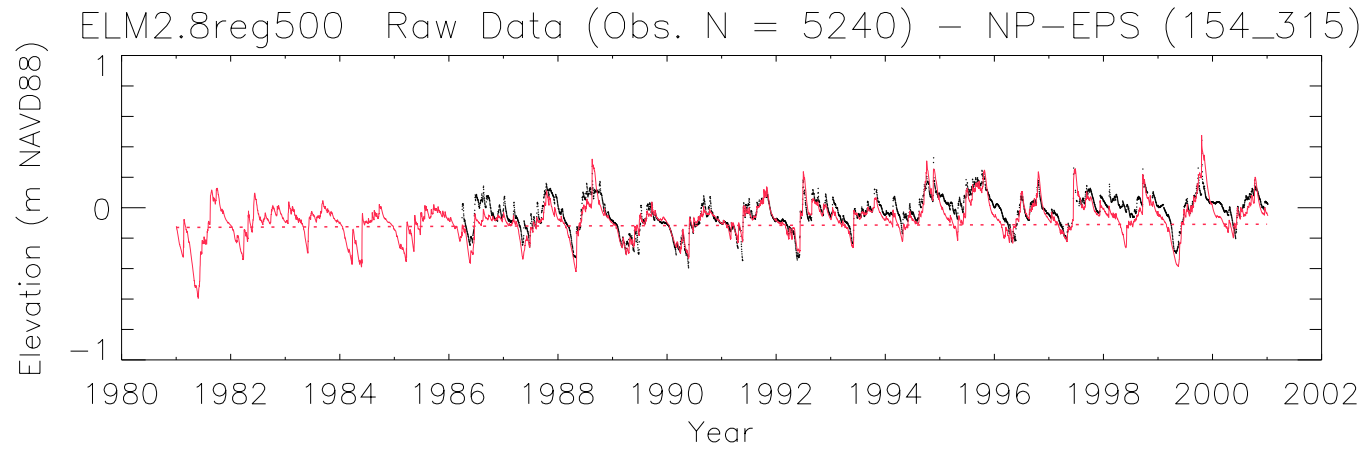

Cumulative Distribution: Raw Data – SWEVER2B (159_307)
Cumulative Distribution
100
80
60
40
20
0
−1
0
1
Elevation (m NAVD88)

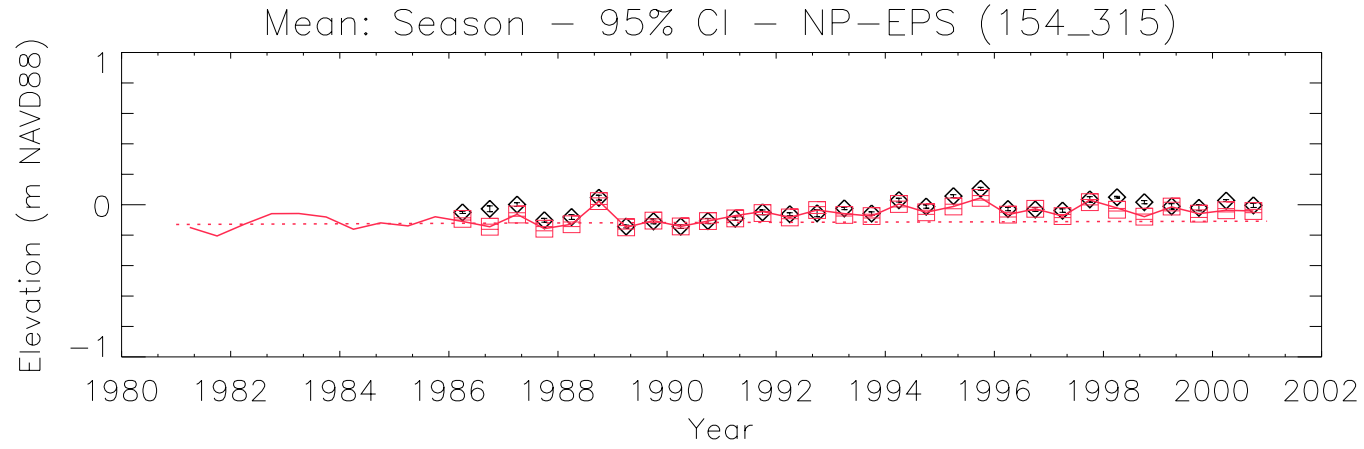

ELM2.8reg500 Raw Data (Obs. N = 6755) – NP–207 (117_314)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

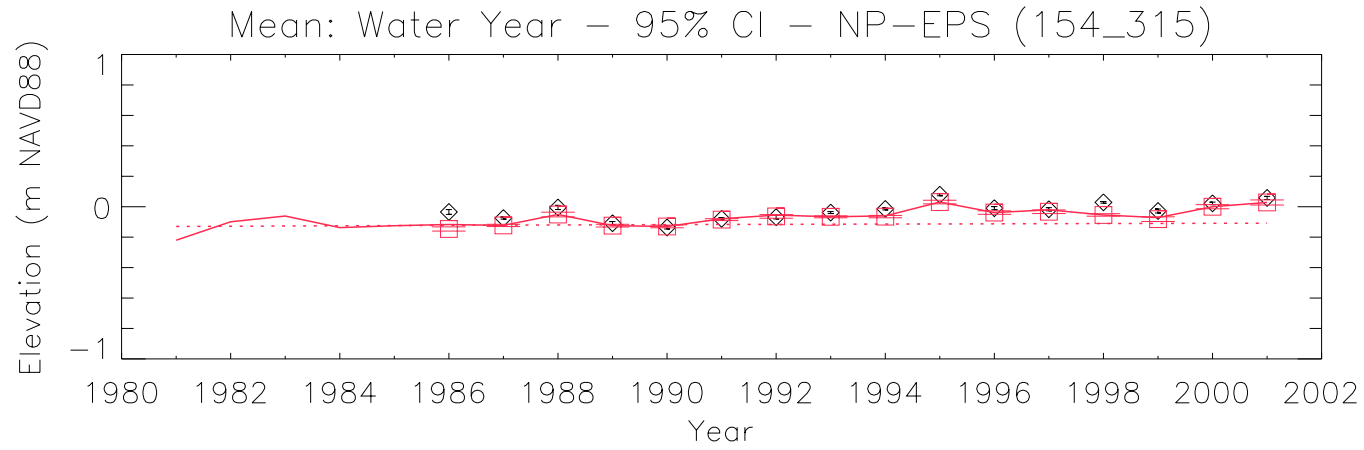

Mean: Season – 95% CI – NP–207 (117_314)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year

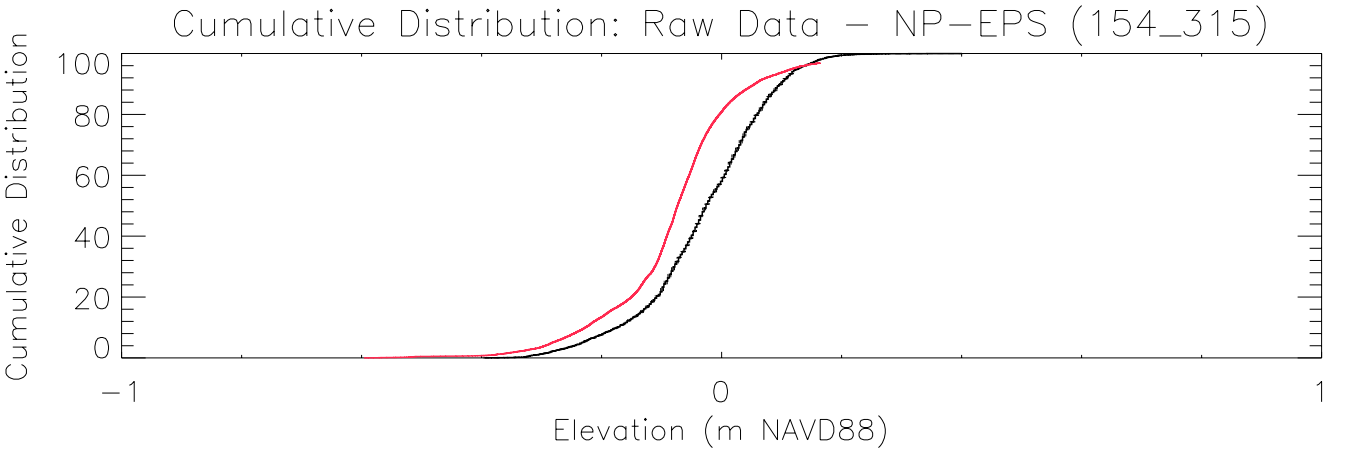

Mean: Water Year – 95% CI – NP–207 (117_314)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year


Cumulative Distribution: Raw Data – NP–207 (117_314)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
Elevation (m NAVD88)


ELM2.8reg500 Raw Data (Obs. N = 5240) – NP–EPS (154_315)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year


Mean: Season – 95% CI – NP–EPS (154_315)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year


Mean: Water Year – 95% CI – NP–EPS (154_315)
Elevation (m NAVD88)
1
0
–1
1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002
Year


Cumulative Distribution: Raw Data – NP–EPS (154_315)
Cumulative Distribution
100
80
60
40
20
0
–1
0
1
Elevation (m NAVD88)

Blank page

Blank page

6.10 Appendix C: Water budgets, ELM & SFWMM

Figures C.1 – C.5. Budget comparisons between ELM and SFWMM for the following basins: WCA-1, WCA-2A, WCA-2B, WCA-3A, and WCA-3B. Each numbered figure contains four graphs:

a) ELM inflows

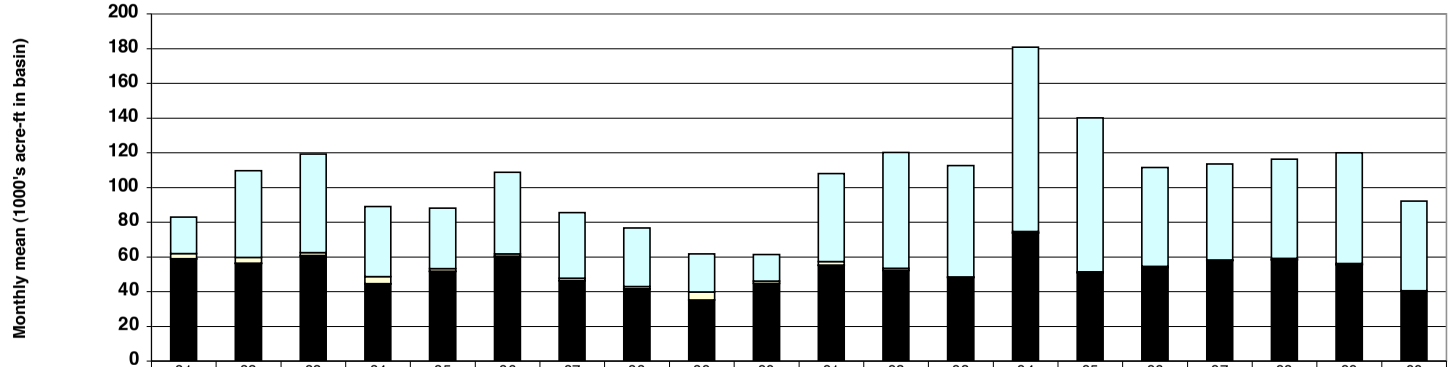
b) ELM outflows.

c) Differences, inflows to SFWMM & ELM

d) Differences, outflows from SFWMM & ELM

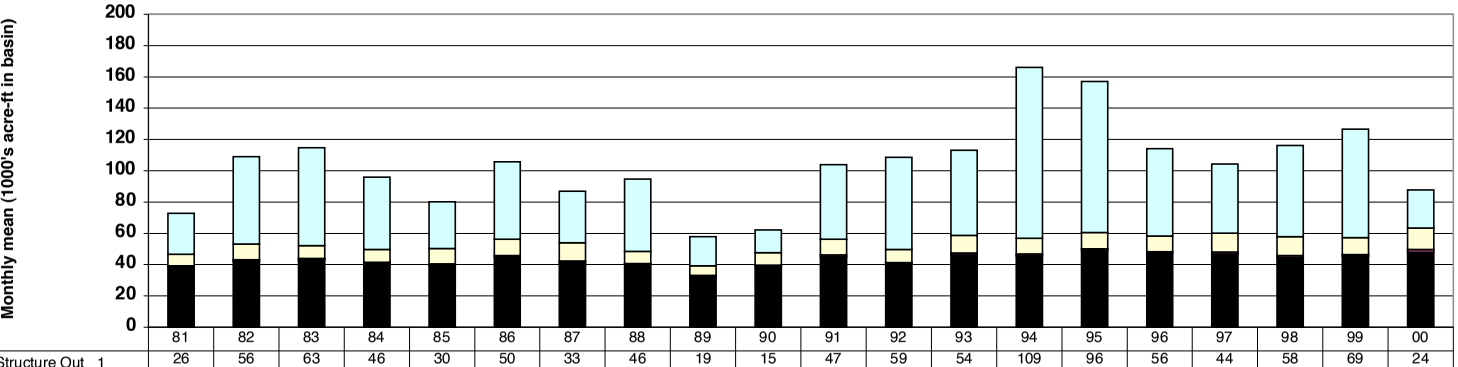
ELM v.2.8.3 performance | ELM hydrologic budget & ELM-SFWMM budget differences | SFWMM calibV5.4 | 4 figures

WCA1 ELM Inflows



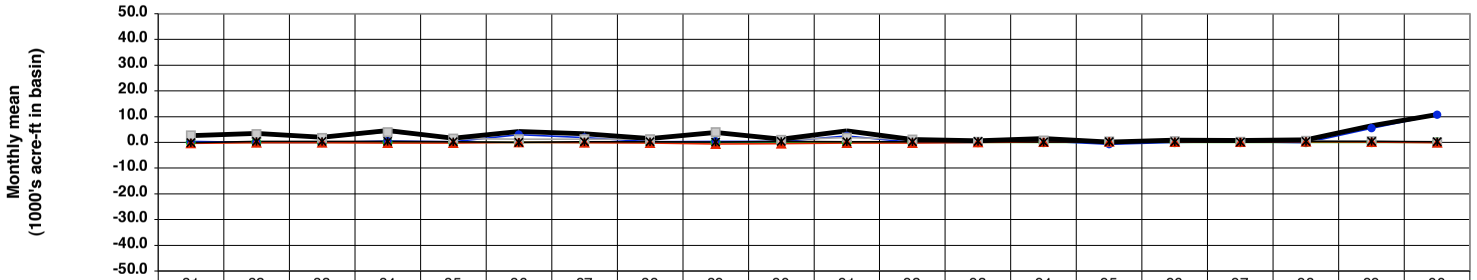
	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Structure In 1	21	50	57	40	35	47	38	34	22	15	51	67	64	106	89	57	55	57	64	51
Groundwater In 1	3	3	2	4	2	1	1	1	4	1	2	1	0	1	0	0	0	1	0	0
Seepage In 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rain 1	59	56	60	44	52	60	46	41	35	45	55	52	48	74	51	54	58	58	55	40

WCA1 ELMOutflows



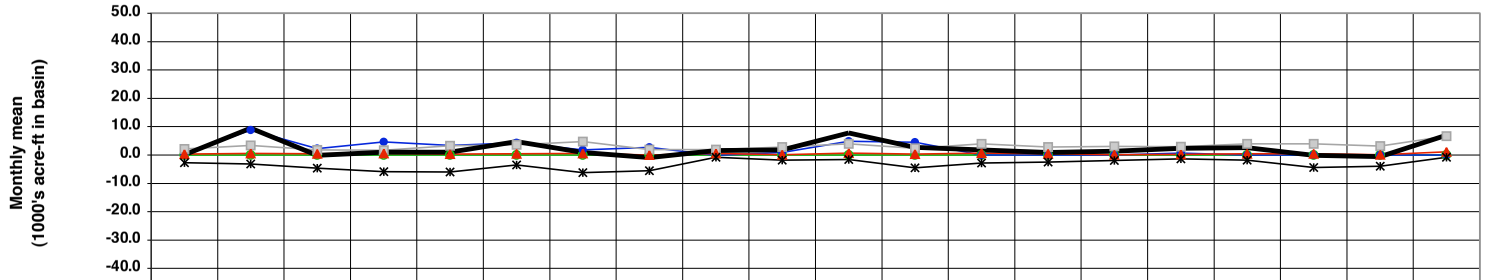
	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Structure Out 1	26	56	63	46	30	50	33	46	19	15	47	59	54	109	96	56	44	58	69	24
Groundwater Out 1	7	10	8	8	10	10	12	8	6	8	10	8	11	10	11	10	12	12	11	14
Seepage Out 1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	2
Evapotranspiration 1	38	42	43	41	40	45	41	40	32	39	45	40	46	46	49	47	47	45	45	48

WCA1 ELM-SFWMM Inflow Differences



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Structure	0.4	0.0	0.0	0.5	0.2	3.0	1.9	0.3	0.4	0.4	2.4	0.0	0.0	0.5	-0.7	0.0	0.2	-0.1	5.4	10.7
Groundwater	2.7	3.3	1.7	4.0	1.4	1.2	1.3	1.3	4.0	0.9	1.9	1.1	0.3	0.7	0.4	0.5	0.3	0.6	0.5	0.1
Seepage	-0.4	-0.1	-0.1	-0.2	-0.3	0.0	-0.1	-0.2	-0.6	-0.5	-0.2	-0.3	0.0	0.1	0.2	0.1	0.1	0.2	0.2	-0.2
Rain	-0.1	0.4	0.4	0.3	0.2	0.0	0.3	0.2	0.1	0.4	0.3	0.3	0.3	0.1	0.2	0.2	0.1	0.3	0.4	0.3
Sum	2.6	3.5	2.0	4.6	1.6	4.2	3.3	1.5	3.9	1.2	4.5	1.1	0.7	1.5	0.1	0.8	0.7	1.1	6.4	10.9

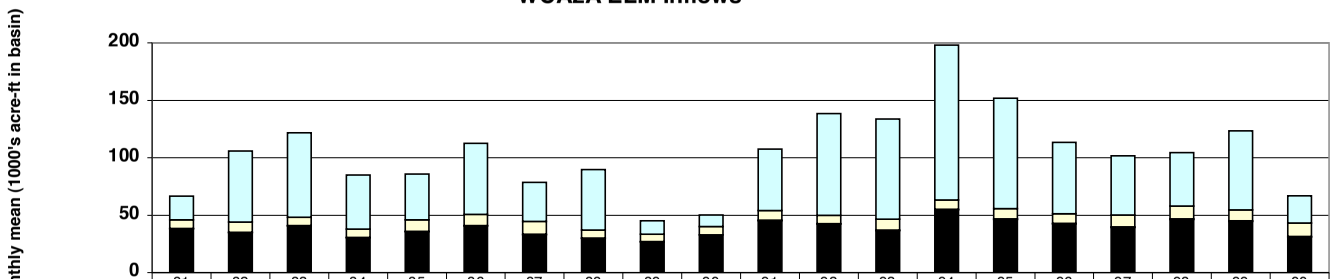
WCA1 ELM-SFWMM Outflow Differences



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Structure	0.4	8.8	2.3	4.6	3.4	4.2	1.8	2.7	-0.1	0.9	4.8	4.6	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0
Groundwater	2.1	3.4	1.9	1.7	3.3	3.6	4.7	1.9	1.9	2.8	3.9	2.3	3.9	2.8	3.0	2.9	3.9	3.9	3.2	6.6
Seepage	0.3	0.5	0.5	0.5	0.3	0.4	0.5	0.0	0.5	0.1	0.7	0.3	0.6	0.5	0.2	0.2	0.4	0.4	0.2	1.2
ET	-2.7	-3.2	-4.6	-5.8	-6.0	-3.5	-6.2	-5.5	-0.8	-1.9	-1.6	-4.6	-2.8	-2.5	-1.9	-1.4	-1.8	-4.4	-3.9	-0.8
Sum	0.1	9.4	-0.1	1.0	1.1	4.8	0.9	-0.9	1.5	1.9	7.8	2.7	1.8	0.9	1.3	2.4	2.6	-0.1	-0.6	7.0

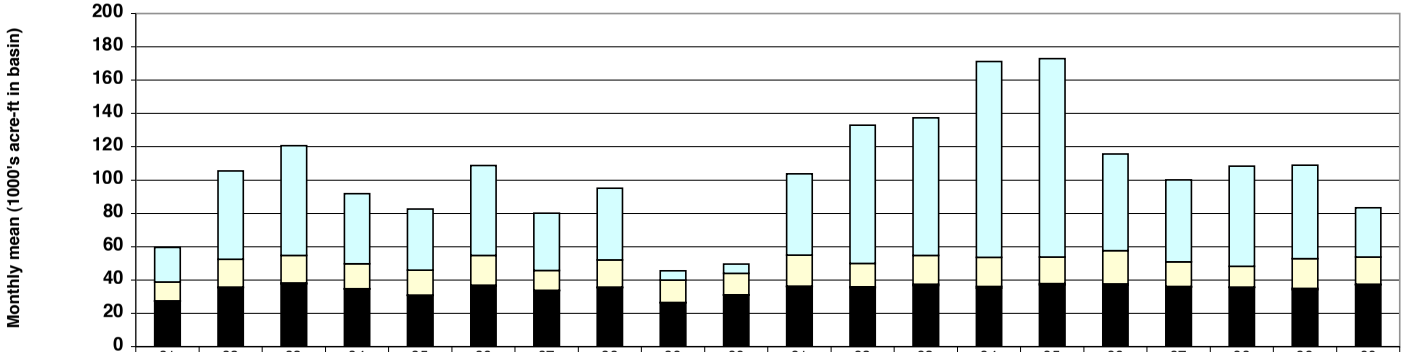
ELM v.2.8.3 performance | ELM hydrologic budget & ELM-SFWMM budget differences | SFWMM calibV5.4 | 4 figures

WCA2A ELM Inflows



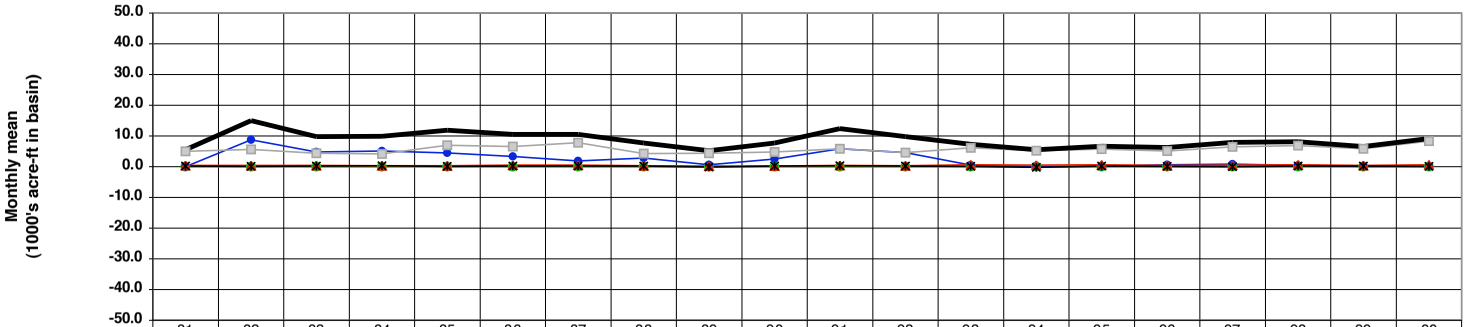
	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Structure In 2	21	62	73	47	40	62	34	53	12	10	53	89	87	135	96	62	52	46	69	24
Groundwater In 2	7	9	7	7	10	10	11	7	6	7	9	7	10	8	9	8	10	11	9	12
Seepage In 2	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	1	1	0	1
Rain 2	38	35	41	30	36	41	33	30	27	33	45	42	36	55	46	42	39	46	45	31

WCA2A ELMOutflows



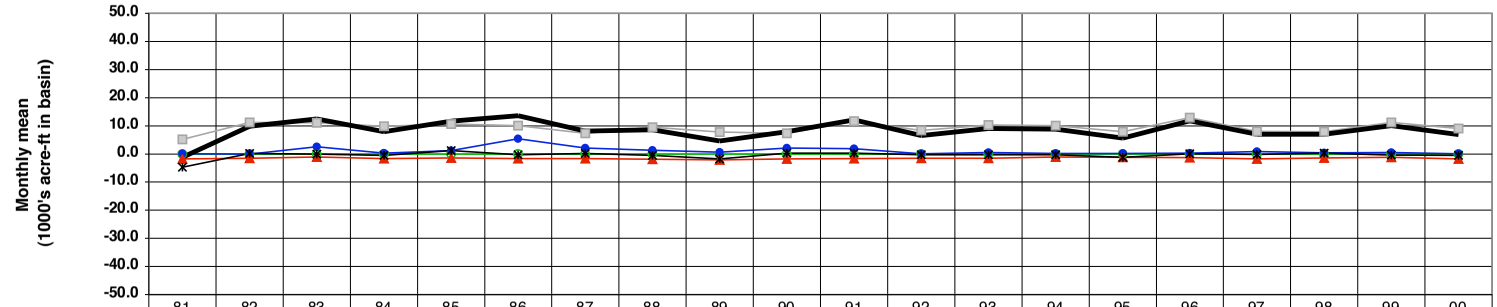
	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Structure Out 2	21	53	66	42	37	54	34	43	6	6	49	83	83	118	119	58	49	60	56	30
Groundwater Out 2	11	17	16	15	15	18	12	16	13	13	19	14	17	18	16	20	15	12	18	16
Seepage Out 2	0	0	1	0	0	1	0	0	0	0	0	0	1	1	1	1	0	0	1	0
Evapotranspiration 2	27	35	38	35	31	36	34	35	26	31	36	35	37	35	37	37	36	35	34	37

WCA2A ELM-SFWMM Inflow Differences



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Structure	0.0	8.8	4.8	5.1	4.4	3.3	1.9	2.9	0.7	2.5	5.9	4.6	0.5	0.1	0.1	0.6	0.9	0.4	0.1	0.1
Groundwater	5.0	5.6	4.4	4.2	7.0	6.6	7.8	4.3	4.4	4.8	5.8	4.5	6.2	5.2	5.8	5.1	6.4	6.9	5.8	8.3
Seepage	0.4	0.5	0.4	0.3	0.4	0.5	0.5	0.3	0.2	0.2	0.4	0.3	0.6	0.5	0.6	0.4	0.6	0.6	0.5	0.6
Rain	0.2	0.1	0.2	0.4	0.1	0.2	0.3	0.2	-0.1	0.2	0.3	0.2	0.1	-0.3	0.2	0.1	0.0	0.2	0.2	0.1
Sum	5.5	15.0	9.8	9.9	11.9	10.6	10.5	7.7	5.2	7.7	12.4	9.7	7.3	5.5	6.7	6.3	7.9	8.1	6.6	9.2

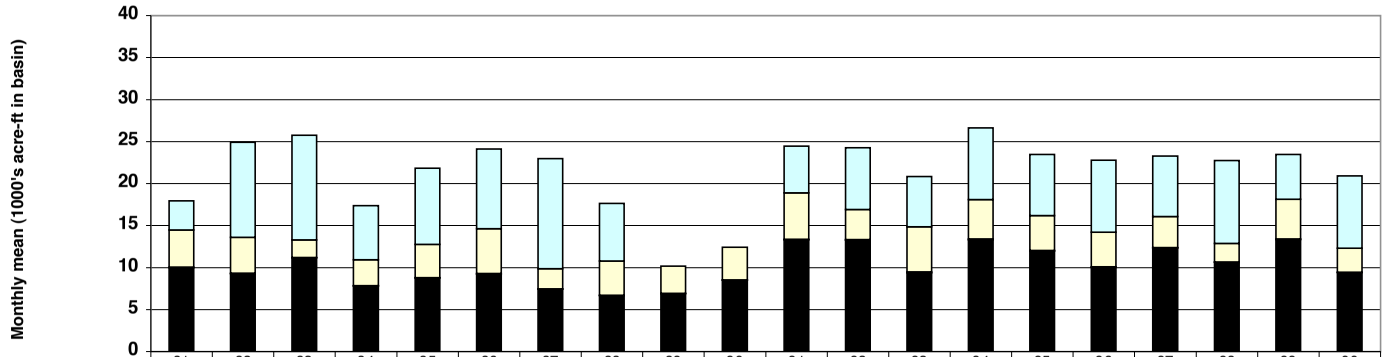
WCA2A ELM-SFWMM Outflow Differences



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Structure	0.0	0.0	2.5	0.2	1.2	5.4	2.1	1.4	0.7	2.1	1.9	0.1	0.5	0.1	0.1	0.3	0.9	0.4	0.5	0.1
Groundwater	5.2	11.2	11.1	9.8	10.7	10.1	7.4	9.5	7.8	7.4	11.7	8.4	10.3	10.1	7.9	12.9	8.0	7.9	11.2	9.0
Seepage	-1.7	-1.5	-1.1	-1.6	-1.4	-1.6	-1.6	-1.8	-2.1	-1.8	-1.6	-1.5	-1.5	-1.1	-1.2	-1.3	-1.8	-1.5	-1.2	-1.8
ET	-4.7	0.2	0.0	-0.5	1.2	-0.3	0.2	-0.5	-1.8	0.3	0.2	-0.3	-0.2	-0.3	-1.2	0.0	-0.2	0.2	-0.4	-0.5
Sum	-1.2	9.8	12.5	7.9	11.6	13.6	8.1	8.5	4.6	7.9	12.1	6.6	9.1	8.8	5.6	11.8	7.0	7.0	10.1	6.8

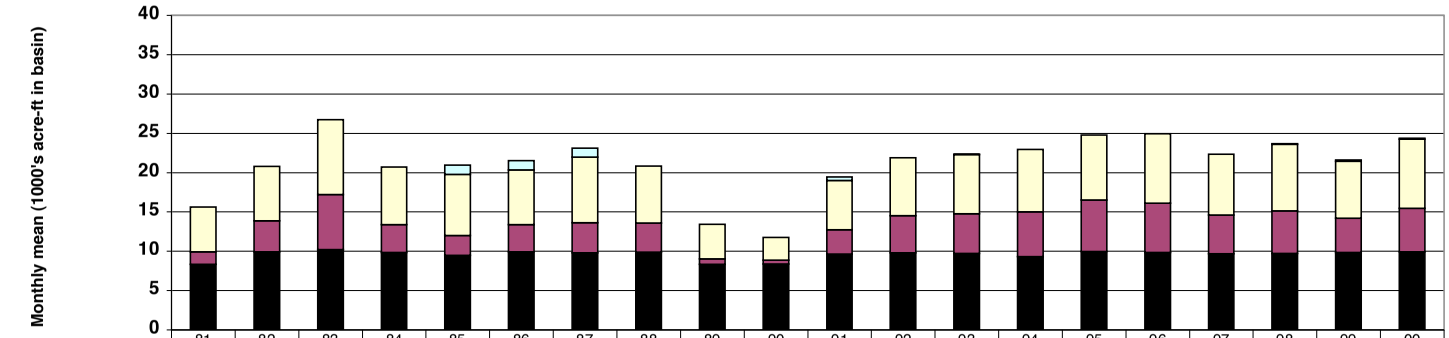
ELM v.2.8.3 performance | ELM hydrologic budget & ELM-SFWMM budget differences | SFWMM calibV5.4 | 4 figures

WCA2B ELM Inflows



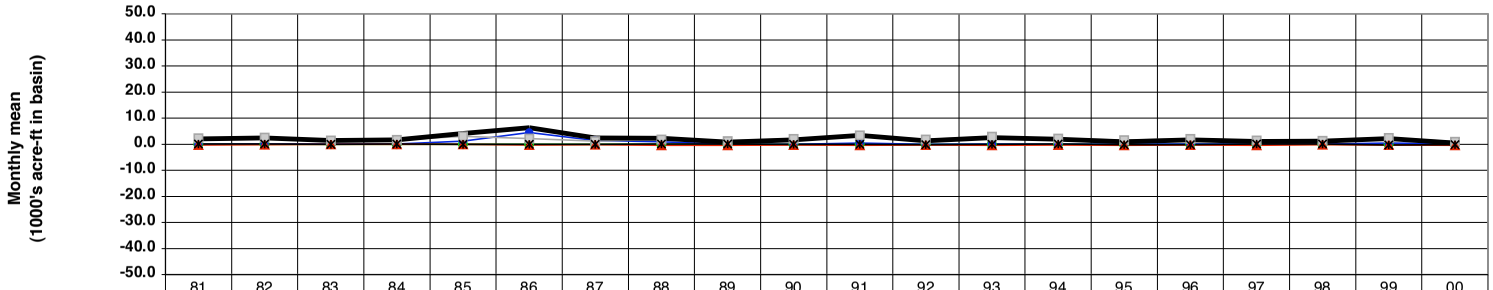
	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Structure In 4	3	11	12	6	9	9	13	7	0	0	6	7	6	9	7	9	7	10	5	9
Groundwater In 4	4	4	2	3	4	5	2	4	3	4	6	4	5	5	4	4	4	2	5	3
Seepage In 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rain 4	10	9	11	8	9	9	7	7	7	8	13	13	9	13	12	10	12	11	13	9

WCA2B ELMOutflows



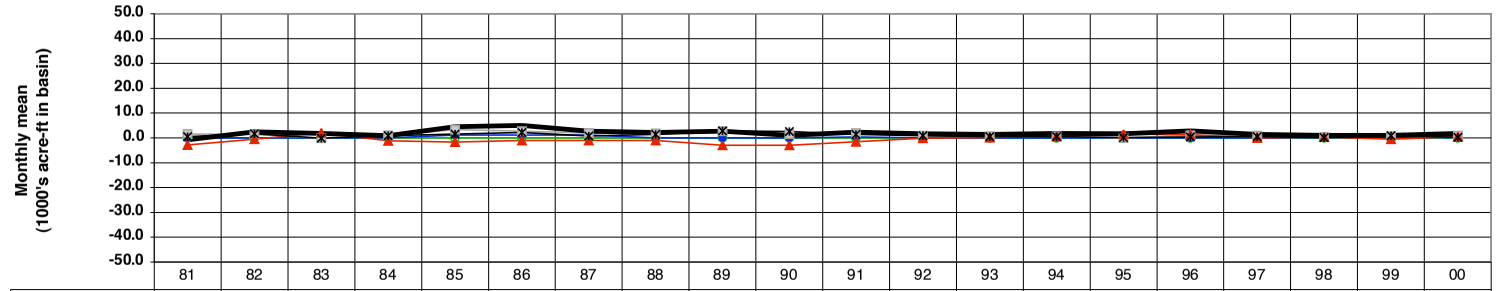
	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Structure Out 4	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Groundwater Out 4	6	7	10	7	8	7	8	7	4	3	6	7	8	8	8	9	8	8	7	9
Seepage Out 4	2	4	7	4	3	3	4	4	1	0	3	5	5	6	7	6	5	5	4	6
Evapotranspiration 4	8	10	10	10	9	10	10	10	8	8	10	10	10	9	10	10	10	10	10	10

WCA2B ELM-SFWMM Inflow Differences



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Structure	0.0	0.0	0.0	0.0	1.2	4.5	1.4	0.9	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.4	0.1
Groundwater	2.3	2.5	1.4	1.7	3.0	2.2	1.2	1.7	1.1	2.0	3.3	1.8	2.8	2.2	1.5	2.1	1.4	1.2	2.4	0.9
Seepage	-0.3	-0.2	0.0	-0.1	0.0	-0.3	-0.1	-0.4	-0.4	-0.3	-0.4	-0.3	-0.4	-0.3	-0.4	-0.3	-0.4	-0.2	-0.3	-0.4
Rain	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.1	0.0	-0.1	0.0	0.0	-0.2	-0.1	0.0	0.0	-0.3	-0.1
Sum	2.0	2.4	1.4	1.6	4.1	6.4	2.4	2.2	0.8	1.7	3.4	1.3	2.5	1.9	0.9	1.7	1.0	1.1	2.1	0.4

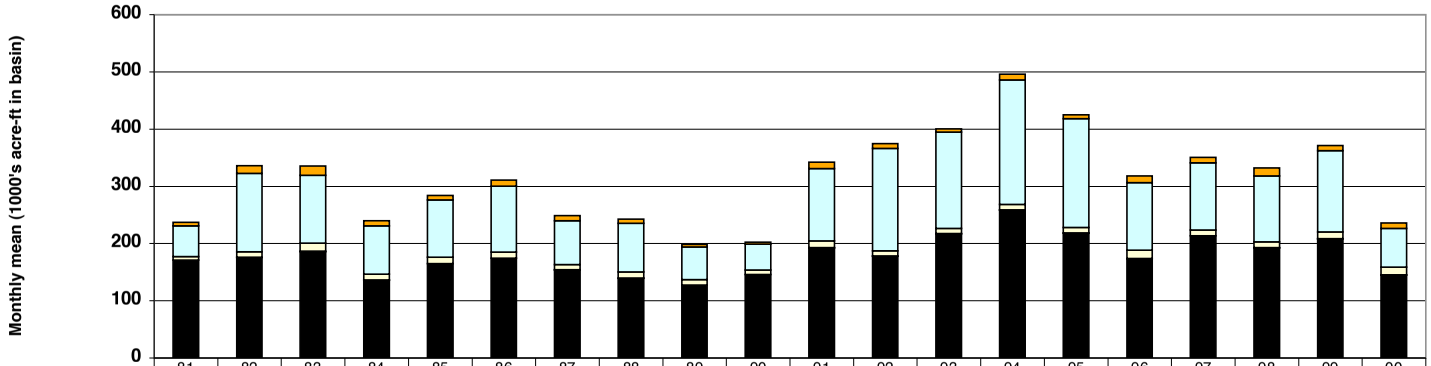
WCA2B ELM-SFWMM Outflow Differences



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Structure	0.0	0.0	0.0	0.0	1.2	1.2	1.2	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.1
Groundwater	1.6	1.4	-0.1	1.1	3.4	2.6	1.9	1.8	2.8	1.5	2.0	0.9	0.7	0.7	0.0	1.1	0.9	0.4	0.6	0.9
Seepage	-2.8	-0.6	2.0	-1.1	-1.7	-1.0	-1.0	-1.0	-2.9	-3.0	-1.6	-0.1	0.1	0.7	1.6	1.3	0.0	0.4	-0.6	0.5
ET	0.5	1.7	-0.1	0.9	1.6	2.2	0.7	1.4	2.9	2.6	1.4	0.9	0.6	0.5	0.1	0.4	0.5	0.1	0.9	0.2
Sum	-0.8	2.5	1.9	0.9	4.5	5.0	2.8	2.1	2.7	1.1	2.3	1.7	1.4	1.9	1.7	2.8	1.4	1.0	1.1	1.8

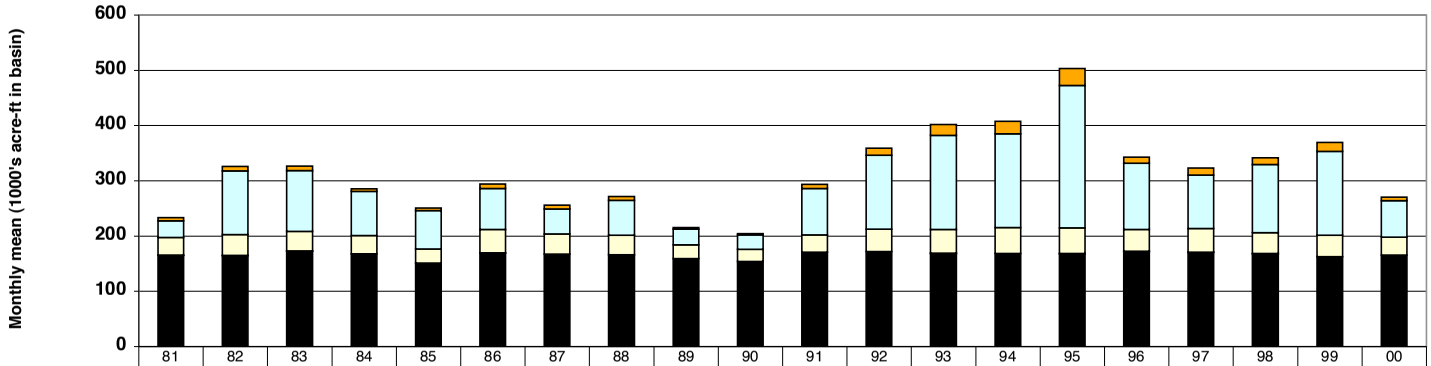
ELM v.2.8.3 performance | ELM hydrologic budget & ELM-SFWMM budget differences | SFWMM calibV5.4 | 4 figures

WCA3A ELM Inflows



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland In 3	6	13	16	9	7	11	9	7	5	3	11	8	6	10	6	12	9	14	9	10
Structure In 3	53	137	119	85	100	115	77	85	57	45	126	179	168	217	190	118	118	115	143	67
Groundwater In 3	6	10	14	10	11	10	9	11	9	8	12	9	9	10	10	15	10	10	12	13
Seepage In 3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
Rain 3	171	176	186	136	165	174	154	139	127	146	193	178	217	258	218	173	213	192	208	145

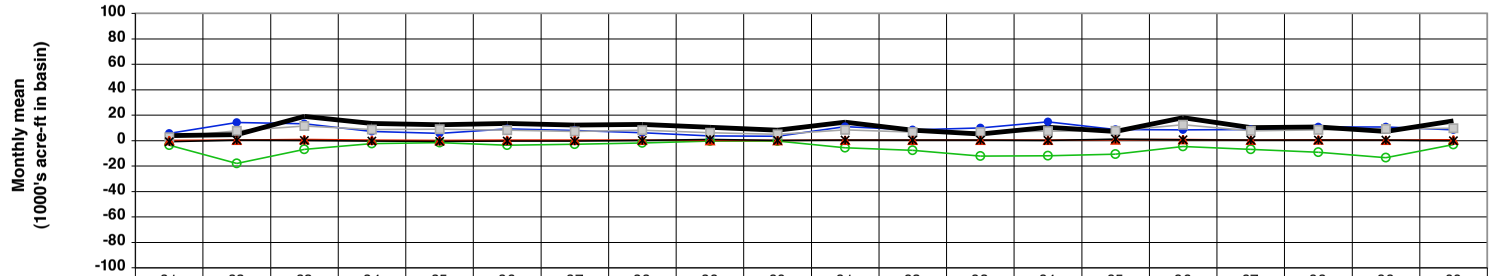
WCA3A ELMOutflows



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland Out 3	6	8	8	5	4	8	7	7	3	3	8	13	19	23	31	11	13	12	16	7
Structure Out 3	30	115	110	80	70	74	45	63	28	26	84	134	170	170	258	120	97	123	152	66
Groundwater Out 3	32	38	36	33	26	42	36	35	25	22	31	40	43	47	46	40	43	38	38	32
Seepage Out 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Evapotranspiration 3	165	165	173	168	151	169	167	166	159	154	171	172	168	168	168	172	171	168	163	166

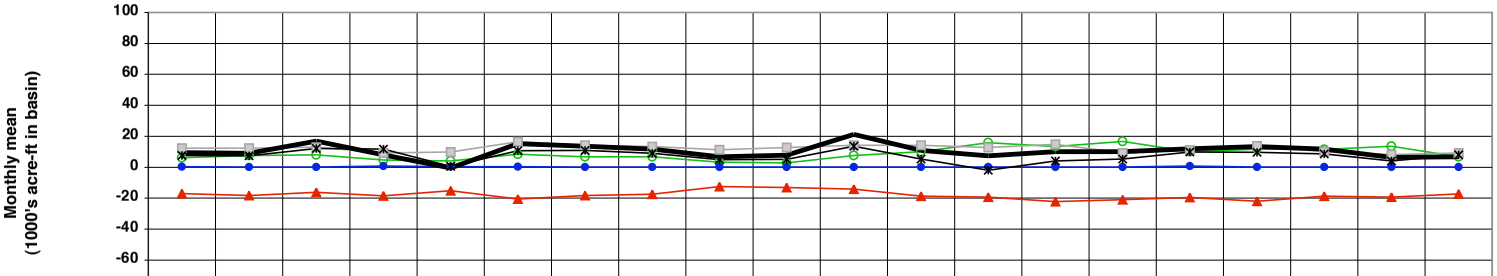
ELM v.2.8.3 performance | ELM hydrologic budget & ELM-SFWMM budget differences | SFWMM calibV5.4 | 4 figures

WCA3A ELM-SFWMM Inflow Differences



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland	-4	-18	-7	-3	-2	-4	-3	-2	0	0	-6	-8	-12	-12	-11	-5	-7	-9	-13	-3
Structure	6	14	13	7	6	9	8	6	4	3	11	8	10	15	9	8	9	11	11	9
Groundwater	3	8	12	9	9	8	7	8	6	5	8	7	7	7	8	13	8	8	9	10
Seepage	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
Rain	-1	0	0	0	-1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
Sum	4	5	19	13	12	13	12	13	10	8	15	8	5	10	7	18	10	11	7	16

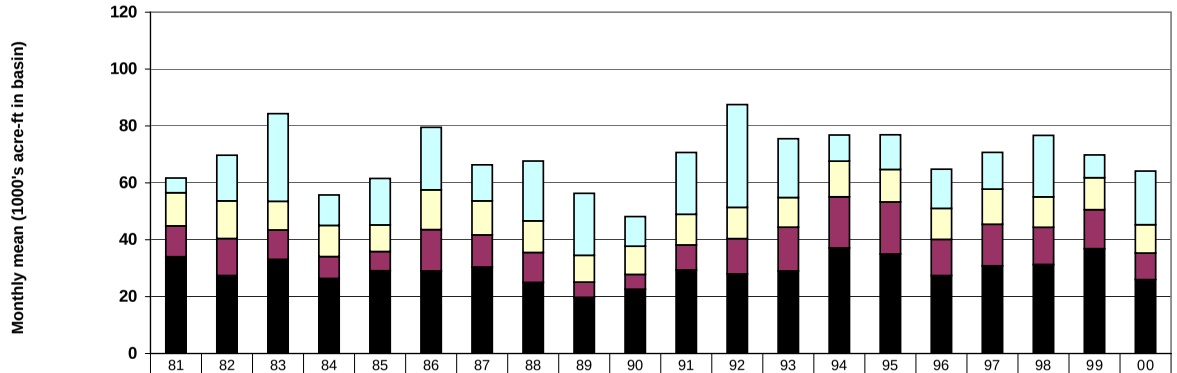
WCA3A ELM-SFWMM Outflow Differences



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland	6	8	8	5	4	8	7	7	3	3	8	10	16	13	17	10	12	12	14	7
Structure	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Groundwater	12	12	13	9	10	16	14	13	11	13	14	14	13	15	9	11	14	10	8	9
Seepage	-17	-18	-16	-19	-15	-21	-18	-18	-13	-13	-14	-19	-19	-22	-21	-20	-22	-19	-19	-17
ET	8	7	12	12	0	11	11	9	5	5	14	5	-2	4	5	10	10	9	4	8
Sum	9	9	17	8	0	15	13	12	7	8	21	11	7	10	10	12	13	12	6	7

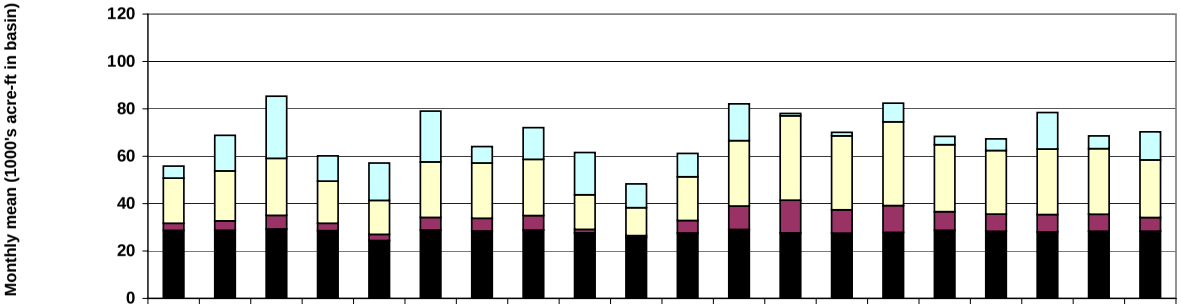
ELM v.2.5 **calib v2.5.2** **ELM hydrologic budget & ELM-SFWMM budget differences** **SFWMM calibV5.4** 4 figures

WCA3B ELM Inflows



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland In 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Structure In 5	5	16	31	11	16	22	13	21	22	10	22	36	21	9	12	14	13	22	8	19
Groundwater In 5	12	13	10	11	9	14	12	11	9	10	11	11	10	13	11	11	12	11	11	10
Seepage In 5	11	13	10	8	7	15	11	11	5	5	9	12	15	18	18	13	15	13	14	9
Rain 5	34	27	33	26	29	29	30	25	20	23	29	28	29	37	35	27	31	31	37	26

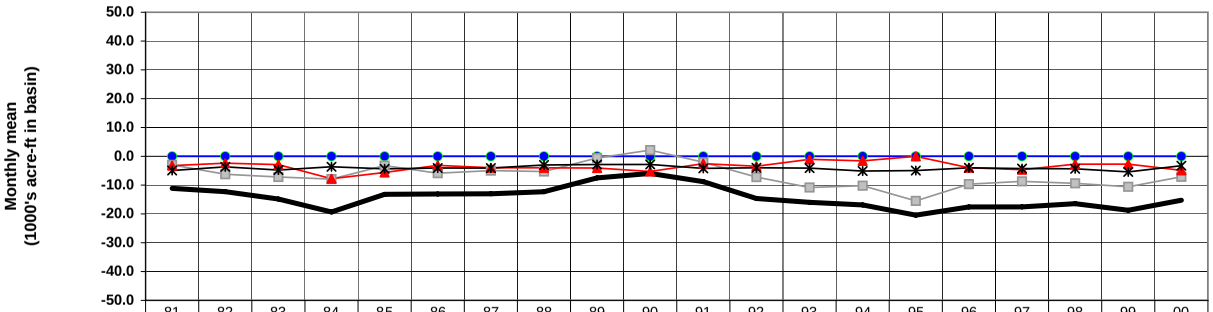
WCA3B ELMOutflows



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland Out 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Structure Out 5	5	15	26	11	16	21	7	13	18	10	10	16	1	1	8	4	5	15	5	12
Groundwater Out 5	19	21	24	18	14	23	23	24	15	12	18	28	36	31	35	28	27	28	28	24
Seepage Out 5	3	4	6	3	3	5	5	6	1	1	5	10	14	10	11	8	7	7	7	6
Evapotranspiration 5	29	29	29	29	24	29	28	29	28	26	28	29	28	28	28	29	28	28	28	28

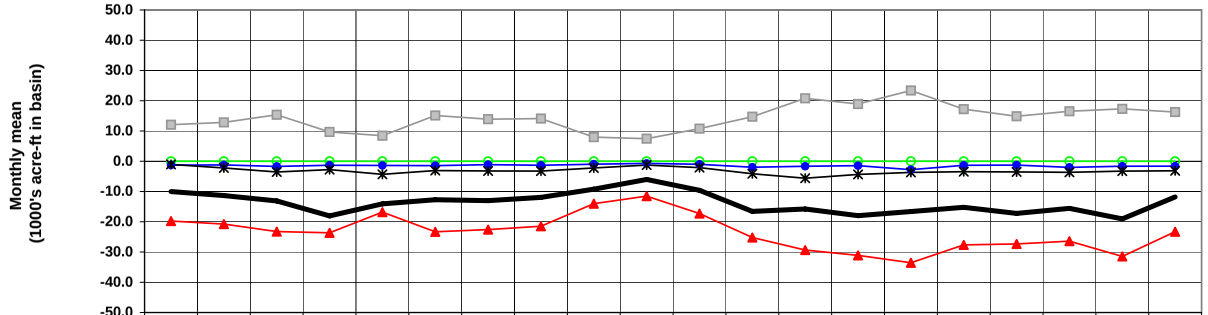
WCA3B ELM-SFWMM Inflow Differences

WCA3B ELM-SFWMM Inflow Differences



	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Structure	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Groundwater	-3.0	-6.2	-7.2	-7.9	-3.2	-5.9	-5.0	-5.3	-0.6	2.1	-2.1	-7.2	-10.8	-10.2	-15.5	-9.7	-8.7	-9.4	-10.6	-7.1
Seepage	-3.2	-2.4	-2.9	-7.8	-5.6	-3.1	-4.0	-4.0	-4.1	-5.3	-2.5	-3.5	-1.1	-1.6	0.0	-4.0	-4.6	-2.7	-2.7	-4.9
Rain	-4.9	-3.7	-4.8	-3.6	-4.4	-4.1	-4.1	-3.0	-2.8	-2.8	-4.2	-4.0	-4.1	-5.1	-4.9	-4.0	-4.2	-4.3	-5.4	-3.3
Sum	-11.1	-12.3	-14.8	-19.3	-13.2	-13.1	-13.0	-12.3	-7.5	-5.9	-8.8	-14.6	-16.0	-16.9	-20.4	-17.6	-17.6	-16.4	-18.7	-15.2

WCA3B ELM-SFWMM Outflow Differences

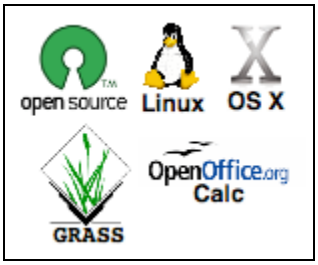


	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Overland	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Structure	-1.3	-1.2	-1.7	-1.3	-1.4	-1.4	-1.1	-1.3	-1.0	-0.7	-1.0	-2.0	-1.7	-1.4	-2.8	-1.3	-1.3	-2.0	-1.6	-1.6
Groundwater	12.1	12.8	15.4	9.7	8.4	15.1	13.9	14.1	8.0	7.5	10.8	14.7	20.8	18.9	23.4	17.2	14.9	16.5	17.3	16.3
Seepage	-19.8	-20.8	-23.2	-23.6	-16.8	-23.3	-22.6	-21.5	-14.0	-11.5	-17.3	-25.2	-29.3	-31.1	-33.5	-27.6	-27.3	-26.4	-31.4	-23.3
ET	-1.1	-2.2	-3.5	-2.8	-4.3	-3.1	-3.2	-3.2	-2.2	-1.3	-2.1	-4.1	-5.6	-4.4	-3.7	-3.5	-3.5	-3.7	-3.3	-3.2
Sum	-10.0	-11.3	-13.1	-18.1	-14.1	-12.7	-13.0	-11.9	-9.2	-6.0	-9.6	-16.6	-15.8	-18.0	-16.6	-15.2	-17.2	-15.6	-19.0	-11.8

BLANK PAGE

Documentation of the Everglades Landscape Model: ELM v2.8

Chapter 10: User's Guide



http://ecolandmod.ifas.ufl.edu

February 13, 2009

Chapter 10: User's Guide

10.1 Overview

The ELM is a freely available, "Open Source" project that we hope will be used and modified by others in the scientific community in a collaborative spirit. Other Chapters of the Everglades Landscape Model (ELM) documentation describe the input data, scientific algorithms & source code, model performance, and other material. This Chapter is intended to instruct users on the steps needed to install and apply the ELM in historical (e.g., calibration) simulations.

To use the ELM, one starts with a computer running some "flavor" of the unix operating system (such as Linux). Basic familiarity with unix is required, but advanced expertise is not absolutely necessary. The ELM is installed from a single script that extracts data and code from two compressed file archives. The executable is then built (compiled & linked) from a script, and the model is ready to go.

In the most common/simple application of ELM, a single script is run to verify what output is desired, execute a model run, and archive the results. The user is guided through the several fundamental checks of model output to verify that the model indeed performed as expected. The outputs are described, covering a range of spatial and temporal scales of the landscape. Their interpretation is dependent on an understanding of the science of ELM covered in the other Chapters of this documentation.

As should be apparent from this and other Chapters of the model documentation, the ELM was designed to be applied by modifying databases, not the model source code. "User-friendly" supporting databases are available to select different outputs, change parameters, or explore/edit aspects of the supporting data. However, those databases need not be immediately opened/modified, depending on the user's initial interest.

A few of the more advanced applications of ELM are covered in brief, but are generally beyond the scope of this User's Guide. These topics include the automated sensitivity analysis of the model, the creation of new subregional applications, and evaluating scenarios of future restoration alternatives. While these applications of the model are all data-driven and relatively straightforward, the details of changes to data and requisite quality assurance are left to a subsequent extension of this guide.

UPDATES to this Chapter, v2.5 to v2.8:

1. numerous edits for clarification; updated installation script and documentation
2. documented use of new netCDF and floating point generic binary map output
3. added scripts and documentation for use of OpenDX for visualizations (Appendix)
4. added details on extracting SFWMM water control structure output for input to ELM future-scenario applications (Appendix)
5. added the "Unix & ELM Cheat Sheet" Appendix, for novice unix users

Computing environment

The ELM is truly a multi-platform simulation model, capable of running in a variety of computing environments without modification. No changes to the C source code, scripts, or "makefile" are needed to move among any of the computing environments that we have tested. The compilation and run scripts detect the type of unix operating system, with no user intervention. This allows the ELM developers to modify one set of code (stored on one file system), and routinely compile & run the OS-specific executables from any available platform. The production environment for ELM is Apple OS X (Darwin) or SuSE Linux on an Intel chip.

10.1.1 Hardware

The ELM can be installed and executed on any one of a variety of common hardware architectures that have some form of unix[1] available (Table 10.1 below). Available storage on the file system (hard disk) should be at least 600 MB: roughly 500 MB is needed for all of the input data/databases and source code, while a 20-year run with basic outputs, including animated monthly time series of a handful of variables, uses approximately 100 MB disk space. Different subregional applications (of various grid sizes) vary the memory (RAM) requirements, but the regional ELM application that is in the standard distribution uses less than 90 MB RAM, irrespective of the simulation length.

10.1.2 Software

No commercial software is necessary. The only requirement to install and use ELM is a unix operating system that includes a gcc[2] compiler. No custom libraries need to be modified/installed beyond those already available in a standard operating system installation with a functional compiler. Tools that are technically "optional", but highly desirable, include a Geographic Information System (the Open Source GRASS GIS is recommended), and spreadsheet software (Open Office Calc is recommended). For optional/recommended software tools, see Appendix: Software recommendations.

---

[1] Some platforms are no longer available to us, and thus we have not tested the ELM code in older platforms and compilers; there is unlikely to be problems with those combinations, however.

[2] GNU Compiler Collection, gcc, at http://gcc.gnu.org/ There are no compiler-specific dependencies, and thus other ANSI C compliant compilers should be compatible with ELM code.

Table 10.1. ELM compilation and execution has been tested in these environments. At the unix command line, type "gcc --ver" and "uname -a" for this information.

Compiler	Operating System	OS release version	CPU
gcc v.3.2 (untested, v2.8)	Sun Solaris	5.8	sparc
gcc v.3.2.2 (untested, v2.8)	Red Hat Linux	2.4.20-27.9smp	i686 (Pentium)
gcc v.3.3 (untested, v2.8)	Apple Darwin	6.8	Power Mac (G4)
gcc v.3.3.3	SuSE Linux	2.6.4-52-default	i686 (Celeron)
gcc v.3.4.3	Red Hat Linux	2.6.9-5.ELsmp	i686 (Pentium)
gcc v.4.0.1	Apple Darwin	8.11.1	i386 (Xeon)

10.1.3 Runtimes

One of the platforms available to the ELM developers is an inexpensive Dell™ laptop with an Intel Pentium™ 2.66 GHz processor. On this computer, the run-time for the regional-ELM implementation (10,394 active grid cells @1 km resolution), with standard output, is slightly over 1 hour for a 20-year simulation. On an Intel Xeon 3.0 GHz processor, the regional application at 500 m grid resolution (41,576 active grid cells) takes less than 4 hours for a 20-year simulation.

10.2 Installing the model

Using an Open Source[3] philosophy, we hope to encourage collaboration in the modeling community. Towards that end, the source code and data are available for download on the ELM web site, and all C source code in the ELM project is documented in detail using the automated "Doxygen" web-based documentation system (see Model Structure Chapter).

10.2.1 Standard

The ELM project is installed in a directory of the user's choosing, without affecting existing operating system "libraries" or other components of the user's file system. To install the ELM, one places the code & data archives into an empty directory, and runs a single script, by following these steps (replacing "X.Y" with "2.8" for ELM v2.8):

1) Obtain the code and data (from CD or http://my.sfwmd.gov/elm)
    a. ELMinstall.sh (installer shell script); also, copy either the ELMsetenv_bash or ELMsetenv_tcsh scripts (see below)
    b. ELMX.Y.data.updateA.B.tar.gz (compressed archive of data, ELM version X.Y, update A.B)
    c. ELMX.Y.src.updateA.B.tar.gz (compressed archive of code, ELM version X.Y, update A.B)
2) Make a home and install your project
    a. Create an empty directory anywhere on your file system, put above 4 files

[3] http://www.opensource.org/

into it, and "cd" into that directory

b. Run the install script on unix command line: "./ELMinstall.sh"

c. Note: the install script guides you on how to set up the several environment variables that are needed. Two additional scripts (one for the "bash" shell, one for the "tcsh" shell) will set the environment variables for you.

3) ***Optional*:** if netCDF output is desired, download & install netCDF libraries from Unidata's web site: http://www.unidata.ucar.edu/software/netcdf/

a. Install netCDF (version 3) libraries according to their instructions

b. in your Driver.make makefile (SME/SMDriver/Sources/Driver_Sources/Driver.make), point the LIB_DIR and INC_DIR variables to the netcdf libraries

c. on line 211 in the ELM globals.h header file, set NCDF3 to true (#define NCDF3 1).

4) Build the ELM executable

a. Run the build script on unix command line (ELM version X.Y): "build ELMX.Y". *NOTE: if you get a message that the "build" command could not be found, a simple first thing to try is type the name of your shell (bash or tcsh...), hit enter. Try the "build" command again: if it still can't be found, your environment may not have been set up correctly.*

10.2.2 Custom

The standard installation is generally all that is needed. However, the user has more flexibility in choosing the location(s) of model output, along with customization of other characteristics of the model. Note that the choice of operating system does not influence any of the installation procedures. For the details of the potential customizations, see this Chapter's Appendix: Environment variables and Directory/file structure.

10.3 Running the model

The ELM is run from the unix command line through the use of "shell" scripts. Basic familiarity with unix is required, but advanced expertise is not absolutely necessary.

10.3.1 Quick start

For those who want to run a simulation "right now" using the defaults set in the standard distribution of source code and data, simply jump in and invoke a script (after installing the model as described above!). In the commands below, replace "X.Y" with "2.8" for ELM v2.8. For all commands and filenames, remember that unix is case-sensitive.

1) Invoke the Run script, responding to its prompts (ELM version X.Y): "Run ELMX.Y myFirstRun"

2) The Run script asks you a couple of questions. Say no to both for now: the model will run, and then the results will be archived in a new directory called "myFirstRun", within the archive directory "$ELM_HOME/arc_out/"

3) Check/interpret the output as outlined in the "Output" section later in this Chapter.

10.3.2 Runtime configuration files

There are two model configuration (text) files[4] that can be modified prior to running the model. One file, "Driver.parm", is edited to select which ecological module(s) to execute, set the starting and ending dates of simulation, and other such model settings. The other, "Model.outList", is edited to select which variables to output, their output type & location, and output frequency. These two configuration files are directly read by the model during the initialization sequence.

10.3.2.1 Driver.parm

This is the primary configuration file, providing significant flexibility to the user. Some of the more common changes that may be made in this configuration are:

- change location of model output
- change start and end dates of simulation
- change output intervals for budgets, canals, and internal variable averaging
- turn on/off habitat switching module
- turn on/off water management modules
- turn on/off various hydro-ecological vertical solution modules
- run sensitivity analyses on parameters

This text file is self-documented at a brief level of detail. This Chapter's "Appendix: Driver.parm" expands on the information for each of these runtime parameters.

The "Check" script is used to quickly check these settings, and edit them if desired (using the standard unix text editor "vi").

10.3.2.2 Model.outList

This text file is exported from the "ModelOutlist_creator_*version*.xls" interface. That spreadsheet database is found in the "$ELM_HOME/SME/Projects/Dbases/" directory. It is "user-friendly" and fully self-documenting, and is perhaps most commonly used for initially selecting and configuring the different output command options. For basically any dynamic variable in the model, the user can select the following combinations of commands to produce output:

- integer-based[5] map time series (animations): "G(AnimDir#,1,VarName)" command (replace AnimDir# with the 1 through 60 suffix of AnimationXX directory name; replace VarName with a descriptive name of variable)

---

[4] in $ELM_HOME/SME/Projects/ELMX.Y/RunParms/

[5] The map time series that are produced are in "unsigned character" binary formats that have the smallest file sizes, and the output maps are scaled by the user via the interface. In addition, ELM v2.8 newly supports output of netCDF and floating point maps.

- scale the values of integer-based map time series output: "S(multiplier,offset)" command (required w/ integer-based output maps)
- floating point map time series (animations): "G(M,4,VarName)" command (replace VarName with a descriptive name of variable; the "4" is the byte size of the output values). NOTE: in addition to this Model.outList change, you must create a directory with this variable's name in your project's ./Output/ directory
- netCDF (floating point) map time series (single file, for animations): "G(C,4,VarName)" command (replace VarName with a descriptive name of variable; the "4" is the byte size of the output values)
- point time series (individual grid cells): "P( )" command
- time interval for output (independent for each variable): "O( )" command

The map time series consists of multiple domain-wide spatial maps of the selected variable at the selected output interval, with each variable's multi-file time series put in a separate output directory ("./Output/*.*/"). The point time series are put in the "./Output/PtSer/" output directory, with a time series at the selected output interval in a separate file for each variable, with each file containing multiple points (grid cells).

Note: summaries of all canals & water control structures ("./Output/Canal/"), and all user-defined Basin/Indicator-Region data ("./Output/Budget/") are always output. The user can modify the output frequency of those data via the "Driver.parm" configuration. (Basins and Indicator Regions are defined in an input map; see the "Modifying Data" section of this Chapter).

Although it is relatively quick and easy to use, it is not necessary to routinely use the ModelOutlist_creator spreadsheet interface: once a user becomes familiar with the output commands, the "Check" script can be used to most quickly check the settings in the "Model.outList" text file, and edit them if desired (using the unix text editor "vi").

10.3.3 Scripts

The following are the scripts that are available for a variety of tasks associated with using the ELM. Most of the scripts are "stand-alone", but are designed in a modular fashion so that they can also be controlled by higher-level calling scripts. (For example, the "Run" script shown above is a main controller script that calls the stand-alone scripts of "Check", "go", and "ArchiveRun", while those latter scripts call others such as "PathModel"). Table 10.2 describes the script usage and hierarchy.

Table 10.2. Scripts used in the ELM project. The three grey-shaded scripts are all that are needed to install and run the ELM. Scripts are modular and nested in a hierarchy; all scripts can be executed as stand-alone applications (w/ 1 exception). Syntax: *ProjName* is the name of the ELM project (e.g., ELM2.8); *runName* is a user-defined name to denote a particular simulation run

Primary script	Secondary script	Syntax	Included/called scripts	Purpose of script
Model installation				
ELMinstallX.Y.sh		ELMinstall*X.Y*.sh, where *X.Y* is model version	none	Install the ELM project in the user's directory. Fully self-documented. (script name came w/ distribution)
build		build *ProjName*	PathELM_HOME, PathModel, PathOSTYPE	Builds an executable of the model project from the make file (compiles, links source code).
Model run				
Run		Run *ProjName runName*	Check, go, CopyInput, ArchiveRun, PathELM_HOME, PathModel, PathOutput, PathArchive	Controller script that configures, runs, and archives a simulation.
	Check	Check *ProjName*	PathELM_HOME, PathModel	View and change the model runtime configuration.
	go	go *ProjName*	PathELM_HOME, PathModel, PathOutput, PathOSTYPE	Simply runs the model executable. NOTE: output from a simulation run made via this script is OVERWRITTEN by a subsequent invocation of this script; use ArchiveRun script to save a simulation.
	ArchiveRun	ArchiveRun *ProjName runName*	PathELM_HOME, PathModel, PathOutput, PathArchive, mkOutDirs	Archives a simulation's output and input as a "keeper" under a user-defined name. It moves all output files and copies selected input files to a user-defined new directory in the $ELM_ARCHIVE_PATH.
finishOutList		finishOutList *ProjName target*, where *target* is file made by ModelOutlist_creator.	PathELM_HOME, PathModel	If ModelOutlist_creator was used: Does the final processing needed on the Model.outList text file that was created by the ModelOutlist_creator workbook (OpenOffice/Excel).
Model distribution/backup				
ArchiveData		ArchiveData *ProjName descript*, where *descript* is descriptive identifier	none	Archives all input data required for ELM historical (e.g., calibration) runs to a compressed tar archive. Used for ELM-version distributions. To use, modify source and target directories in the script.
ArchiveSrc		ArchiveSrc *ProjName descript* where *descript* is descriptive identifier	PathELM_HOME	Archives all required ELM source code to tar archives in two locations: an uncompressed one in $ELM_HOME, and a compressed one in a remote directory. Used for ELM-version distributions. To use, modify destination directory in the script.
Utility				
	PathArchive	PathArchive	none	Checks validity of $ELM_ARCHIVE_PATH for model archiving and exports it if needed.
	PathELM_HOME	PathELM_HOME	none	Determines if the (fundamental) $ELM_HOME variable appears valid.
	PathModel	PathModel	PathELM_HOME	Checks validity of the base $ModelPath for the model Project data/executable, and creates & exports that path if needed.
	PathOutput	PathOutput *ProjName*	PathELM_HOME, PathModel	Checks for validity of an existing OutputPath for model output (defined in Driver.parm file) and exports it if valid.
	PathOSTYPE	PathOSTYPE	none	Determines if the $OSTYPE variable reflects a tested platform. (The name of the script is for consistency with similar script names, and OSTYPE is only used in relation to paths/filenames).
	CopyInput	NA (is not stand-alone)	none	Only called from the "Run" script. It has 2 primary purposes: 1) Force the user to write some Notes on simulation about to be run; 2) Create named copies of frequently-changed data files.
	mkOutDirs	mkOutDirs *OutputPath ProjName*	none	Create the required output directory names if they have been removed from the model Project OutputPath.
rmAnim		rmAnim *OutputPath ProjName*	none	Delete all files in Animation* directories for a project in a given path. For a measure of safety, this is only used as a stand-alone script, and the user needs to manually type in the path, then confirm the deletions.
Advanced: acquire water control structure flows				
getDSSflow		getDSSflow	none	Acquire flow data. Full instructions for advanced applications not in this current documentation.
StrNames		StrNames	none	(Compiled binary) to extract names of structures from a "DSS" catalog. Full instructions for advanced applications not in this current documentation.
Advanced: GRASS for animations, vector canal input/visualization, other				
AnimGrass		NA (not distributed, FYI only)	PathOutput, PathArchive	GRASS (script not distributed): Links model output to a GRASS directory in preparation for animation using xganim.
AnimGrassNow		NA (not distributed, FYI only)	none	GRASS (script not distributed): Runs xganim within GRASS.
AnimGrass_rm		NA (not distributed, FYI only)	none	GRASS (script not distributed): Deletes the links to model output and the other GRASS animation files for a particular variable.
reachin		NA (not distributed, FYI only)	none (but uses ELM variable "$ModelPath")	GRASS (script not distributed): Creates GRASS ascii vector files for all canals contained in the CanalData.chan ELM-input file.
reachinvect		NA (not distributed, FYI only)	none	GRASS (script not distributed): Import ALL reaches from ascii into Grass binary vector format.
reach_calib_v2.4		NA (not distributed, FYI only)	none	GRASS (script not distributed): Display canal reaches in distinguishing colors, and show thel water control structures.

10.4 Input data modification

Several databases are used to modify and document a variety of important components of the ELM. The purpose of this section is to call the user's attention to these self-documenting databases, which are *critical to the use of the ELM, particularly when learning the model*. Other data sources (described in the Data Chapter) are used for time series data that are input to the model, and some form of GIS (below) is needed to visualize and modify the spatial maps that are input to the model.

10.4.1 Databases

Our goal has been to create a system of integrated, relational databases using the Open Source MySQL. However, these prototype databases are not ready for release, and we instead use the Open Office Calc spreadsheet software[6] to perform the necessary data management functions. Table 10.3 provides an overview of the primary functions of these data management systems.

Table 10.3. Spreadsheet-based databases used in a) data maintenance and documentation of model parameters and model variables, b) generating source code of model, and c) generating output configurations for model runs. Databases are found in $ELM_HOME/SME/Projects/Dbases/.

Database name	Database functions
GlobalParms_vX.Y.xls	1) Maintain and document (incl. units and source/metadata) parameters that are globally distributed across model domain. 2) Generate code of header file, transferring parameter documentation to model source code. 3) Generate upper and lower values of all parameters for automated sensitivity analysis.
HabParms_vX.Y.xls	1) Maintain and document (incl. units and source/metadata) parameters that are specific to different habitats in the model domain. 2) Generate code of header file, transferring parameter documentation to model source code. 3) Generate upper and lower values of all parameters for automated sensitivity analysis.
ModelOutlist_creator_vX.Y.xls	1) Generate all input-configuration commands for any model variable, map and point time series output. 2) For all Everglades monitoring sites, calculate model grid cell row-column (at any grid scale) from its geographic coordinates. 3) Maintain and document (incl. units and source/metadata) all variables used in the model. 4) Generate code of multiple header files, transferring documentation of variables to model source code.

[6] fully compatible with Microsoft Excel

The single exception to the use of Open Source software is our database ($ELM_HOME/SME/Projects/Dbases/Structs_attr_vX.Y.fmp) of the attributes of water control structures, for which we continue to use FileMaker Pro software. This database continues to be very useful in creating new subregional applications or modifying water control structures for evaluating alternative water management scenarios. However, it is not essential to the use of ELM in the mode intended for this User's Guide Chapter: the water control structure attributes for the current simulations are documented through snapshots of the records for all of the necessary water control structures, and the text input file can be viewed or modified using spreadsheet software (see Data Chapter).

10.4.2 GIS

Any software capable of reading raw/generic binary data arrays can be used to edit/visualize the map inputs. The ELM developers use the GRASS GIS (see Appendix: Software recommendations). Through the use of unix symbolic links between the GRASS and the ELM data directories, the ELM directly reads GRASS project data files (uncompressed binary data and text header) as model input. However, no GRASS-specific encoding of binary information is used, and thus the data files may be opened with any program that can read raw binary data arrays. Scripts are available to directly input and visualize the ELM (text) canal vectors in GRASS.

There are three sub-directories within an ELM project's input ./Data/ directory: Map_bin, Map_head, and Map_hist. The model reads each raw binary data file in the Map_bin subdirectory, and reads its associated header description in the Map_head subdirectory. The history and other pertinent metadata are in the Map_hist subdirectory, but that information is not used in the model.

All spatial data are referenced to zone 17 of the Universal Transverse Mercator (UTM) geographic coordinate system, relative to the 1927 North American Datum (NAD). The ELM v2.8 regional application uses 0.25 km^2 square grid cells that encompass an area of 10,394 km^2 (4,013 mi^2) in the active domain. See the Data Chapter 4 of this documentation for other application domains. All of the maps of the regional application are bounded by a rectangle of UTM coordinates in zone 17 (NAD 1927), as shown in the lines in the below regional-domain example of the text header files:

zone:	17 (UTM zone)
northing:	2,953,489 m (UTM north coord)
southing:	2,769,489 m (UTM south coord)
easting:	580,711 m (UTM east coord)
westing:	472,711 m (UTM west coord)
columns:	216 (number of columns in 2D array)
rows:	368 (number of rows in 2D array)
east-west resol.:	500 m (grid cell length in 2D array)
north-south resol.:	500 m (grid cell width in 2D array)
format:	*"X"* bytes/cell, as defined below
compressed:	0 (no compression)

The *"X"* value of "format" of the raw binary data is one of the following:

0.: 1 byte per grid cell
1: 2 bytes per grid cell
3: 4 bytes per grid cell

10.5 Output

During the initialization phase of a simulation, the various configurations that the user chose are echoed to the console (screen). Subsequently, the simulation date is iterated on the console as the computations are made. A successful simulation will end with the following message printed to the screen:
"END. The simulation(s) took zz.zzz minutes to run using your yyyyy OS box.", followed by other messages depending on the scripts that are running.

10.5.1 Quick start

Upon completion of (or during) a simulation, the user is advised to make the following minimal checks to verify that the simulation was "well-behaved".

4) To verify that no errors were in the simulation, search the "Debug/Driver1.out" text file (see below) for the all-caps string "ERROR", which can be the full word or part of a word (i.e., "capacityERROR");

5) View the "Budget/budg_Wcm1" text file, and verify that the cumulative mass balance error variables, "SumERR_*" for each Basin & Indicator Region identity, is reasonable, i.e., on the order of tens of microns height.

6) Peruse one of the spatial time series of map outputs to verify the spatio-temporal dynamics "pass the laugh test" . Viewing an animation, or individual maps, of the "SfWatAvg" (surface water depth, averaged during output intervals) variable is a good choice - assuming the user kept that variable's output commands in the Model.outList configuration.

7) Dive into the other output files as desired, using the below descriptions as your guide.

10.5.2 Output file structure

During a simulation, all output is always written to the "Output/" directory in the user's output path (Table 10.4). After a simulation terminates, the output may be moved (archived) to the user's archive path via the "ArchiveRun" script (which is also called by the "Run" script). In the below directory descriptions, "ProjName" is the Project name (such as ELM2.8) that was input by the user on the command line.

Un-archived output location: "OutputPath/ProjName/Output/", where "OutputPath" is the absolute path to model output, changeable in the "Driver.parm" file.

Archived output location: "$ELM_ARCHIVE_PATH/ProjName/runName", where

"$ELM_ARCHIVE_PATH" is the archive path set up by the user[7], and "runName" is a user-defined name to denote a particular simulation run.

Table 10.4. Output directories and description of files they contain. These directories are relative to the un-archived or the archived output locations described above.

Output directory	Output description
Animation1...Animation60 (integer-based maps)	Map time series for individual variables, with separate directory for each variable. After archiving a simulation, non-empty directories are moved & renamed with the variable names.
or: VariableA, VariableB, ... (integer based maps)	Map time series for individual variables, with separate directory for each variable. Prior to archiving a simulation, the directory names are simply Animation1, Animation2, ..., Animation60 (maximum).
VariableA, VariableB, ... (floating point maps)	Map time series for individual variables, with separate directory for each variable.
NCDF	Directory containing (one or more) single netCDF file(s) per variable; each file contains a map time series
Budget	[BIR] Time series of budgets and pre-set Performance Measures in Basins/Indicator Regions (BIR).
Canal	Time series of a) canal depths and constituent concentrations, and b) water control structure flows and constituent concentrations.
Debug	Variety of detailed output for debugging and error checking.
PtSer	Time series of individual variables at point (grid cell) locations distributed through model domain.

10.5.3 Debug (errors and warnings)

The "Debug/" directory will always contain at least two debug-related files. Truly critical errors (such as missing inputs, memory constraints, etc) will terminate the simulation with an informative message. Numerical errors or warnings do not necessarily terminate the simulation (in order to allow the user to debug the problem). It is important to monitor the Driver*.out files for any errors or warnings, particularly after configuring a new application:

- Driver0.out: text file that echoes input data that were successfully read, including simulation start-end dates, hydro-ecological parameters, output configurations and others.
- Driver1.out: text file that contains a variety of warnings, error messages; details of model output are printed, depending on the level of the "debug" parameter (in "Driver.parm", see Runtime configuration section of this Chapter).

[7] "$ELM_ARCHIVE_PATH" was set up by the user when installing the ELM. The location may be set to anywhere, but initial installation was in "$ELM_HOME/arc_out/"

The "Debug/" directory will contain two debug-related files when running the Water Management modules:

- ON_MAP_CANAL.txt: a tab-delimited 2D array text file of the modifications to the "ON_MAP" file that was done by the "Canal-marsh flux module of the Water Management code (see Model Structure Chapter).
- CanalCells_interaction.txt: text file of list of cells that interact with each canal reach

10.5.3.1 Postprocessing Debug text files

All files in the "Debug/" directory are text files. The Driver*.out files are intended to be searched/queried using any text editor. The "ON_MAP_CANAL.txt" file is best visualized after import into any spatial mapping program or GIS (such as GRASS).

10.5.4 Spatial: Basin & Indicator Region (BIR) time series

Budgets and preset Performance Measure variables are output at the different spatial scales defined by the hydrologic Basins and Indicator Regions (BIR) input map. As discussed in the Model domains section of the Data Chapter ("basins" input Data file), hydrologic basins are "parent" regions that (may) contain "child" Indicator Regions, and parent basins' data include (e.g., sum) the data on all child Indicator Regions contained within them. Basin 0 is the entire model domain. Well-drawn BIR spatial distributions are particularly useful for evaluating output dynamics (budgets and Performance Measures) along ecological gradients. Table 10.5 provides an overview of the budget and Performance Measure variables in each of the output files.

Table 10.5. Budget and preset Performance Measure variables in Basins & Indicator Regions (BIR). The variables are output for each BIR in the individual files.

Budget file name	**Budget file description**	**Budget variables for each BIR**																					
budg_Wacr1...5	Water budget: all storages, acre-feet units	total volume - prior interval	total volume - current interval	rain-in	evaporation-out	transpiration-out							structure-in	structure-out	overland-in	overland-out	levee seepage-in	levee seepage-out	ground water-in	ground water-out	error-current interval	error-cumulative sum	avg-inputs	avg-outputs
budg_Wcm1...5	Water budget: all storages, height (cm) units	total volume - prior interval	total volume - current interval	rain-in	evaporation-out	transpiration-out							structure-in	structure-out	overland-in	overland-out	levee seepage-in	levee seepage-out	ground water-in	groundwater-out	error-current interval	error-cumulative sum	avg-inputs	avg-outputs
budg_S1...5	Salt/tracer budget: all storages, mass-per-area units	total mass - prior interval	total mass - current interval										structure-in	structure-out	overland-in	overland-out	levee seepage-in	levee seepage-out	ground water-in	groundwater-out	error-current interval	error-cumulative sum	avg-inputs	avg-outputs
budg_Par1...5	Phosphorus budget: all storages, mass-per-area units	total mass - prior interval	total mass - current interval	rain-in									structure-in	structure-out	overland-in	overland-out	levee seepage-in	levee seepage-out	ground water-in	groundwater-out	error-current interval	error-cumulative sum	avg-inputs	avg-outputs
budg_P1...5	Phosphorus budget: all storages, mass units	total mass - prior interval	total mass - current interval	rain-in									structure-in	structure-out	overland-in	overland-out	levee seepage-in	levee seepage-out	ground water-in	groundwater-out	error-current interval	error-cumulative sum	avg-inputs	avg-outputs
budg_Pwat1...5	Phosphorus budget: water-borne storages, mass-per-area units	total mass - prior interval	total mass - current interval	rain-in	settle-out	surfwat. mineralization-out	porewat. mineralization-out	periph. uptake-out	macroph. uptake-out	desorb-in	sorb_to-out										error-current interval	error-cumulative sum	avg-inputs	avg-outputs
budg_Pliv1...5	Phosphorus budget: live biotic storages, mass-per-area units	total live mass -prior interval	total live mass -current interval	macroph. mass-current interval	calc.periph. mass-current interval	noncalc.periph. mass-current	macroph. NPProd.-in	calc. periph. GPProd.-in	noncalc. periph. GPProd.-in	macroph. mortality-out	calc. periph. mortality-out	noncalc. periph. mortality-out									error-current interval	error-cumulative sum	avg-inputs	avg-outputs
budg_Pded1...5	Phosphorus budget: dead biotic storages, mass-per-area units	total dead mass -prior interval	total dead mass -current interval	macroph. ToSoil-in	periph. ToSoil-in	decomp.-out	settle-in			sorb_to-in	desorb-out										error-current interval	error-cumulative sum	avg-inputs	avg-outputs
Perf. Measure file name	**File description**	**Performance Measure variables for each BIR**																					
BIRavg1...5	Hydro-ecological Performance Measures	surface water depth (m)	unsaturated water depth (m)	TP conc. surface water (mg/L)	TP conc. pore water (mg/L)	TP conc. Soil (mg/kg)	noncalc. periph. Biomass (gC/m^2)	calc. periph. Biomass (gC/m^2)	macroph. Biomass (kgC/m^2)	land elevation (m, NGVD or NAVD)													

10.5.4.1 Budgets (in BIR)

The "Budget/" directory contains tab-delimited text files with budgets of water, phosphorus, and salt/tracer in the BIRs. The reporting time interval is selected by the user (see Runtime configuration section of this Chapter). In each budget, all inflows and outflows to/from each BIR are summed for the relevant variables within each reporting interval. For example, a 30-day reporting interval will result in a hydrologic budget that reports the sum of the different inflows (rain, seepage inflow, etc) and outflows (ET, seepage outflow, etc.) within each 30-day period during the simulation. Numerical errors in mass conservation[8] are always calculated for all budgets, both cumulative during each reporting interval, and cumulative across the model simulation period.

10.5.4.2 Preset Performance Measures (in BIR)

The "Budget/" directory also contains tab-delimited text files with preset Performance Measure averages in BIRs. The reporting time interval is selected by the user (see Runtime configuration section of this Chapter), and is used to calculate the daily arithmetic mean value of each performance measure within the interval. These Performance Measures include hydrologic, biogeochemical, and biological dynamics within the region.

10.5.4.3 Postprocessing BIR text files

All BIR budget and Performance Measure files are in tab-delimited text format, and thus can be directly read into any spreadsheet program such as Open Office Calc or Microsoft Excel. The primary method for ELM postprocessing is the use of scripts written in the Python scripting language. The ELM developers have a flexible set of Python postprocessing scripts that will produce a variety of summaries of these data for visualization and analysis, but that development is not complete enough for release. Spreadsheet templates for different summaries of the output data are available from the developers, but are unsupported.

10.5.5 Spatial: Domain-wide map time series

Virtually any variable in the model may be output as domain-wide maps at a user-specified output interval (see Runtime configuration section of this Chapter). These maps may then be analyzed individually, summarized across time, or animated using visualization software.

Integer-based (scaled from floating point) maps: If a simulation has not yet been archived, the integer-based output maps of any user-selected model variable are placed in

[8] The *maximum* magnitude of cumulative errors in mass balance of water storage dynamics ranges within the order of (positive or negative) 1 to 10 microns, depending on the cumulative interval (monthly or multi-decade period-of-simulation), the presence/absence of canal interactions, and the spatial scale of the budgeted region. The *maximum* magnitude of cumulative errors in mass balance of phosphorus storage dynamics ranges within the order of (positive or negative) 0.001 to 0.01 ug/m^2, depending on the cumulative interval (monthly or multi-decade period-of-simulation), the presence/absence of canal interactions, and the spatial scale of the budgeted region.

one of the AnimationZZ directories in the Output directory, where "ZZ" is an integer between 1 – 60. As described earlier, the model archiving process renames the directories to those of the variable it contains.

10.5.5.1 Postprocessing map files

As configured by the user via the ModelOutlist_creator interface (see Runtime configuration section of this Chapter), all output maps are 2D rectangular arrays in either a) scaled integer-based or unscaled floating point, generic/raw binary format (i.e., they are not encoded with any software-specific attributes), or b) self-documenting netCDF formatted files.

Integer-based (scaled from floating point) maps: To save significant disk space compared to floating point arrays, the integer-based map files are output as "1-byte, unsigned integer" data. In any given directory containing a time series of maps of a given variable, the numeric values in the 2D arrays range from 0-255. The value of "255" is reserved for grid cells that are "off-map", or outside of the active domain. The parameters in the scaling equation chosen by the user (via the ModelOutlist_creator) for each output variable must be used to rescale the integer maps back to the actual (floating point numbers and) units of the model using the equation:

model_floatValue = outMap_intValue * Multiplier + Offset,

where model_floatValue is the actual value of the floating point number calculated in the model, outMap_intValue is the integer number stored in the map array, and Multiplier and Offset are the scaling multiplier and offset, respectively, input by the user in the Model.outList. The units of the model_floatValue for each variable were given in the ModelOutlist_creator interface. For example, ponded surface water depth (SURFACE_WAT) is often scaled for output using a Multiplier of 0.01 and Offset of 0.0; a value of "90" in the output map is equal to 0.90 m depth calculated by the model.

The array size (number of rows and columns) of the output maps in this integer format match the input maps for the model project.

Floating point maps: If the user selected either generic, binary floating point format or netCDF format for map output, no scaling is used for the output.

The array size (number of rows and columns) of the output maps in this floating point format have two additional rows and two addition columns, relative to the size of the input maps for the model project. This is self-documenting in the netCDF files, and displayed in the header for the floating point generic binary files.

*Software:*Any software capable of opening or importing generic/raw binary spatial arrays can be used to analyze and/or animate the time series of output maps. The Open Source GRASS GIS and its associated "xganim" animation program can be used to analyze and visualize the output. As reviewed in the Appendix of this Chapter, many other tools, such as the Open Source OpenDX, or the commercial IDL, are available for geospatial analysis and visualization. The ELM developers have various postprocessing codes (using a custom C program, GRASS, and IDL scripts) for summarizing and visualizing spatial output, but they are not fully developed for public release.

10.5.6 Spatial: Point (grid cell) time series

The "PtSer/" directory contains tab-delimited text files with point (grid cell) time series output. Virtually any variable in the model may be output in this format, at user-selected grid cell locations and output intervals (see Runtime configuration section of this Chapter). A separate file is created for each model variable that is requested for output, and each file has multiple fields (columns) for multiple grid cell locations.

10.5.6.1 Postprocessing point time series text files

All point time series files are in tab-delimited text format, and thus can be directly read into any spreadsheet program such as Open Office Calc or Microsoft Excel. The primary method for ELM postprocessing is the use of scripts written in the Python scripting language. The ELM developers have a flexible set of Python postprocessing scripts that will produce a variety of summaries of these data for visualization and analysis, but that development is not complete enough for release. Spreadsheet templates for different summaries of the output data are available from the developers, but are unsupported.

10.5.7 Spatial: Canal (vector) time series

The "Canal/" directory contains tab-delimited text files with canal (vector) time series output of

- CanalOut: instantaneous water depth in all canal reaches,
- CanalOut_P: instantaneous total phosphorus concentration in all canal reaches,
- CanalOut_S: instantaneous salt/tracer concentration in all canal reaches.

These variables are all of the state variables used in the canals of water management simulation, and the user can select the output interval for this group of outputs (see Runtime configuration section of this Chapter).

10.5.7.1 Postprocessing canal time series text files

All canal (and water control structure) time series files are in tab-delimited text format, and thus can be directly read into any spreadsheet program such as Open Office Calc or Microsoft Excel. The primary method for ELM postprocessing is the use of scripts written in the Python scripting language. The ELM developers have a flexible set of Python postprocessing scripts that will produce a variety of summaries of these data for visualization and analysis, but that development is not complete enough for release. Spreadsheet templates for different summaries of the output data are available from the developers, but are unsupported.

10.5.8 Spatial: Structure (point/cell) flow time series

The "Canal/" directory contains tab-delimited text files with water control structure (vector) time series output of

- structsOut: summed (across each output interval) water flows through all water control structures,
- structsOut_P: flow-weighted (across each output interval) mean total phosphorus concentration at all water control structures, and

- structsOut_S: flow-weighted (across each output interval) mean salt/tracer concentration at all water control structures.

These variables are all of the state variables used in the structure flows of water management simulation, and the user can select the output interval for this group of outputs (see Runtime configuration section of this Chapter).

10.5.8.1 Postprocessing structure time series text files

All water control structure (and canal) time series files are in tab-delimited text format, and thus can be directly read into any spreadsheet program such as Open Office Calc or Microsoft Excel. The primary method for ELM postprocessing is the use of scripts written in the Python scripting language. The ELM developers have a flexible set of Python postprocessing scripts that will produce a variety of summaries of these data for visualization and analysis, but that development is not complete enough for release. Spreadsheet templates for different summaries of the output data are available from the developers, but are unsupported.

10.6 Advanced applications

The following topics are generally beyond the scope of this User's Guide Chapter, but are included in brief summary in order that users may have some guidance if they desire to advance beyond standard, historical simulation runs.

10.6.1 Sensitivity analysis

The user can run the automated sensitivity analysis on model parameters whose results were was described in the Uncertainty Chapter. The "S_ParmName" parameter in the Driver.parm configuration file (see Model configuration section of this Chapter) is used to control which parameters are modified as follows:

- S_ParmName= ALL: evaluate model sensitivity to changes in each of the parameters listed in the input data file SensiParm_list,
- S_ParmName= ParameterName: evaluate model sensitivity to changes in the single parameter whose name is ParameterName, or
- S_ParmName= NONE: no sensitivity analysis, and thus a normal, single simulation run using only the nominal parameter sets

The values of the parameter ranges are changed in the GlobalParms and the HabParms databases: separate "worksheets" are available to calculate and export _LO and _HI (low and high estimates of parameters in) parameter files that are read by the model during the sensitivity analysis. Upon invoking a sensitivity analysis via the S_ParmName parameter, a suite of simulations are executed sequentially when the user executes the model (from either the Run or the go script). An Open Office Calc template is available from the ELM developers for postprocessing the single output file[9] from the multiple

[9] actually, the single BIRavg output file for all of the sequential simulations can be spread over multiple files (unrelated to sensitivity) if the number of Indicator Regions is large, i.e., BIRavg1 – BIRavg5 as described in the Model output section of this User's Guide Chapter

runs.

10.6.2 Evaluating project alternatives

To evaluate most (likely all) water management alternative scenarios, no source code needs to be changed, and ecological parameters (in GlobalParms and HabParms databases) generally are not expected to be changed. For a new management alternative, the user just needs to modify the following input data files (which are all described in the Data Chapter):

- CanalData.chan: any changes to the canal/levee topology and attributes,
- CanalData.struct: any changes to the water control structure attributes,
- CanalData.struct_wat: water control structure (daily) water flows (that are output from SFWMM or other tool),
- CanalData.struct_TP: water control structure (daily) Total Phosphorus concentrations,
- CanalData.struct_TS: water control structure (daily) Total Salt/tracer concentrations.
- (?) GlobalParms_NOM: if appropriate, alter the parameter that estimates the annual rate of sea level rise

To add a new canal, a new canal reach ID is added to the CanalData.chan text file, adding the canal reach attributes and the geographic point coordinates that define the segments of a reach. Existing canal reaches can be "turned off" (ignored by model) by assigning a negative width attribute to that reach. GRASS scripts are used to aid in this process and visualize any new topology of the canal network. Other scripts are used to determine which, if any, new water control structures are required, extracting the appropriate time series of flows from a "DSS" formatted file that was output from the SFWMM (which is the current modeling tool for evaluating hydrology of management alternatives). See the Appendix of this Chapter for information on extracting data from the "DSS" file.

The meteorological boundary conditions for the 1965-2000 period of record are contained in the current (rain.BIN, ETp.BIN) input files. The general assumption in forecasting the responses of the system to management changes is the following: If the system were to be subjected to the same meteorological conditions as those observed between 1965-2000, how would the system respond under a new suite of management rules and/or infrastructure?

Obviously (?), there are other assumptions that are involved with forecasting the system responses to future management alternatives. While the data modification/input methods are generally simple and scripted, the details of the steps, including the assumptions and the necessary data quality assurance, are beyond the scope of this User's Guide.

10.6.3 New subregional applications

To implement a new subregional application of the ELM, no source code needs to be changed. The following input files require modification/re-scaling:

- Input maps: all input maps must be reconciled to the spatial resolution and extent of the new domain (i.e., with new data, or rescaling/interpolating existing data)
- CanalData.chan: canal reaches from the regional model application that are within the new domain may be kept (as they use geographic, not grid cell, coordinates); the upper left corner of the origin of the rectangular domain requires changing (if necessary),
- CanalData.struct*: water control structure attributes and flow/concentration data from the regional model application that are within the new domain may be kept, but the Structs_attr.fmp database (or another calculator) should be used to calculate the new grid cell locations of the geographic coordinates of the water control structures; unused structures need to be removed from all CanalData.struct* files,
- Driver.parm: modify the parameter that defines the model grid cell area
- Model.outList: use the ModelOutlist_creator interface to calculate the new model grid cell locations of the named monitoring stations for which output is desired
- gridmapping.txt: run the GridMap preprocessor application to generate the new linked list of the SFWMM grid cells that are mapped to the grid cells of the new ELM application (for boundary condition data on meteorological inputs and stage at the periphery of the new domain)

While the data modification/input methods are generally simple, the details of the steps, including the necessary data quality assurance, are beyond the scope of this User's Guide.

10.7 Appendix 1

10.7.1 Driver.parm configuration file

The following table contains extended documentation of all of the adjustable parameters in the "Driver.parm" input file that is input to the model to configure a simulation run.

Parameter	Brief metadata	Extended instructions
/MyOutputPath/	{output path (absolute path, w/o ProjName) }	Path for model output can be on any file system. If user requests many animations at high output frequency (e.g., 20 variables, daily), a local hard disk directly attached to host machine can become important to model run time.
1/1/1981	{Sim start date (yyyy/mm/dd), min= 1965/01/01 }	User is informed of error if attempting to start simulation outside of the range of available boundary condition data (1/1/1981 or 1/1/1965 through 12/31/2000, depending on project).
12/31/2000	{Sim end date (yyyy/mm/dd), max= 2000/12/31 }	User is informed of error if attempting to end simulation outside of the range of available boundary condition data (1/1/1981 or 1/1/1965 through 12/31/2000, depending on project).
00/00	{Sim re-init date (mm/dd)(no Position Analysis, mo=00)}	Used only in "Position Analysis", in which simulation is re-initialized annually on a given month/day. If month=00, Position Analysis is not invoked. Position Analysis is not fully updated/supported in v2.8.
ELM	{model name (needs to match CanalData input files)}	Used in distinguishing subregional model projects (e.g., ELM_wca2@500m) from the default regional "ELM". Used primarily to ensure model is using correctly geo-referenced data in CanalData.* input files in subregional projects.
Model version= v.2.8	{model version (e.g., v.2.1)}	Model version identifier to label output files.
CellArea= 1000000.0	{grid cell area, m^2}	The area of an individual model grid cell; standard regional application is 1,000,000 m^2 (1 km^2).
budg_Intvl= 0.0	{interval (julian days), BIR stats (0=calendar-month)}	Time interval for summary calculations in all budget output files (./Budget/budg_*) in Basins/Indicator Regions (BIR). Value >0 is julian day interval; a value=0.0 is an exact calendar-month interval (accounting for leap years etc.).
BIRhyd_avg_Intvl= 7.0	{interval (julian days), BIR hydro stats (0=calendar-month)}	Time interval for summary calculations in the BIRhydro output files (./Budget/BIRhydro*) in Basins/Indicator Regions (BIR). Value >0 is julian day interval; a value=0.0 is an exact calendar-month interval (accounting for leap years etc.).

avg_Intvl= 30.0	{interval (julian days), cell-avgs (0=calendar-month)}	Time interval for all internally-calculated temporal means in BIRavg output files (./Budget/BIRavg*) in Basins/Indicator Regions (BIR). Value >0 is julian day interval; a value=0.0 is an exact calendar-month interval (accounting for leap years etc.).
seed= 568	{random number seed; UNUSED in current version}	UNUSED
dt= 1.0	{time step (days, use 1.0) for vertical fluxes}	The model time step for vertical solutions only. The dt should remain at 1 day for any scale application.
hyd_iter= 12	{**EVEN number**, number of horiz iterations per dt}	The number of iterations, or time slices, per dt for horizontal solutions such as cell-cell overland flow. To determine the appropriate value for a new application, see the ELM documentation for theoretical estimates for different model scales and expected velocities. The 1 km^2 regional ELM application uses hyd_iter = 12 (i.e., a 2 hour time step).
debug= 2	{0:Minimal 1:BasinChek 2:Default 3:More 4:Canal 5:Lots}	The choice of how much information to print to a debug (text) output file (./Debug/DriverX.out, X'th simulation, X=1 in a standard run w/o Sensitivity Analysis). The recommended standard is debug= 2. Higher values will produce very large volumes of information and should be used in relatively short simulations. **See text below this Table for details.
debug_point= 62 43	{focal cell (row col) for Driver1.out if debug>2}	The row-column coordinates of the focal grid cell for 5x5-cell windows of output data that are written to the (text) debug file at high values of the debug parameter.
S_ParmName=NONE	{Sensitivity analysis: "NONE", "ALL", or ParameterName}	Invoke an automated sensitivity analysis on "ALL" parameters in the input data file "SensiParm_list", or on a single parameter whose exact name is provided, or "NONE" for a standard, single simulation run. See text of User's Guide for details.
HabSwitchOn= 0	{Habitat switching (succession) on=1, off=0}	Invoke the habitat switching (succession) module of the model. See text of Model Structure Chapter in the ELM documentation for some details on module.
WatMgmtOn= 1	{Water management and canal network on=1, off=0}	Invoke the water management modules, with flows through water control structures in the network of canal/levee vectors. Normally this is "on". If turned "off", all water management network topologies and managed flow dynamics are inoperative, and thus the only flow constraints are those imposed along the periphery of the model domain (aka a simulation of the "Natural System" that is not compartmentalized).

Scenario= calib	{scenario/alternative name (case sensitive)}	Model scenario (alternative) identifier to label output files.
Scenario modifier= myRun	{scenario/alt modifier or descriptor}	An additional descriptor of specifics to add to the model scenario (alternative) identifier to label output files.
Sectors= 1 0 7 10 9 2 8 12 4 99;		The (left-to-right) sequence of calls to ecological modules (sectors) in the time loop of the simulation. See text of Model Structure Chapter in the ELM documentation for details on the structure of the model time loop, and summaries & details of each module. A single-phrase description of each module is given below in this table (and the "Driver.parm" file).
{Below are not input fields; for descriptive purposes only}		
Sequence for calling modules:	1 0 7 10 [13] 9 2 8 12 4 99	Recommended sequence of module calls. See text of Model Structure Chapter in the ELM documentation for details on the structure of the model time loop.
Module #0	hydrology: horiz raster fluxes (& water management if it is on)	See text of Model Structure Chapter in the ELM documentation for details on module.
Module #1	global forcings: vertical fluxes (& succession if it is on)	See text of Model Structure Chapter in the ELM documentation for details on module.
Module #2	algae/periphyton: vertical fluxes	See text of Model Structure Chapter in the ELM documentation for details on module.
Module #4	DOM/DOP: vertical fluxes	See text of Model Structure Chapter in the ELM documentation for details on module.
Module #7	hydrology: vertical fluxes	See text of Model Structure Chapter in the ELM documentation for details on module.
Module #8	macrophytes: vertical fluxes	See text of Model Structure Chapter in the ELM documentation for details on module.
Module #9	phosphorus: vertical fluxes	See text of Model Structure Chapter in the ELM documentation for details on module.
Module #10	salt/tracer: vertical fluxes	See text of Model Structure Chapter in the ELM documentation for details on module.
Module #12	Floc: vertical fluxes	See text of Model Structure Chapter in the ELM documentation for details on module.
Module #13	ESP P settling model mode, do NOT invoke 2,4,8,9,12	See text of Model Structure Chapter in the ELM documentation for details on module.
Module #99	summary budget & stats	See text of Model Structure Chapter in the ELM documentation for details on module.

*10.7.1.1 **Debug levels:*

- debug =0 Echo short console info on iteration# etc, print critical error/warning info. USE WITH CAUTION.
- debug =1 Report mis-configured basin flows. Currently same level as debug=2.
- debug =2 DEFAULT for general use, more warnings etc.

- debug =3 Echo long console output, prints additional (non-critical) errors/warnings to DriverX.out (for X'th simulation run) file
- debug =4 Prints details of cell vertical and/or horizontal flux data, and details of indiv canal fluxes, to DriverX.out (for X'th simulation run)
- debug =5 Prints grid_map information, and prints to another canal debugging file for special purposes

10.7.2 Environment variables

The required environment variables are the following:

Environment variable	Unix path	Description
ELM_HOME	/My/Directory/	The absolute path to the "home" directory where you install the source code (and by default, the data of multiple projects) of ELM. Can be anywhere on the user's networked file system(s).
ELM_ARCHIVE_PATH	/Any/Directory/arc_out/	The absolute path to the directory where simulation run "keepers" of (multiple) ELM project(s) are archived (and thus not overwritten in subsequent simulation runs!). Can be anywhere on the user's networked file system(s). Suggested default during ELM installation was within the $ELM_HOME.

The highly recommended addition to the user's path (to executables) is:

Add to user's path env.	Description
$ELM_HOME/SME/scripts/	The location of all ELM scripts.

The optional environment variable is the following:

Environment variable	Unix path	Description
ModelPath	/Anywhere /SME/Projects/	The absolute path to the (multiple) project(s) of ELM data and executables. Can be anywhere on the user's networked file system(s). For testing different code sets with one single data location, we can set the $ModelPath as a system environment variable. In the default (distribution) version, the $ModelPath is determined from $ELM_HOME and is not needed as an environment variable.

10.7.3 Directory/file structure

The complete directory structure of an ELM project.

Directory structure	File type	File descriptions
$ELM_HOME/		
include/sme/	source code	header files
SME/		
scripts/	source code	unix shell scripts
SMDriver/Sources/		
Driver_Sources/	source code	main program, utilities
SpatMod/	source code	spatial fluxes
Tools/	source code	I/O tools
UnitMod/	source code	unit model
Dbases/	databases	databases for data export to model
Projects/		
ELM2.8/		
Data/	input data	all input data files (maps in subdirs)
Map_bin/	input data	all map binary arrays
Map_cats/	input data	all map category definitions
Map_head/	input data	all map header definitions
Map_hist/	input data	all map metadata/history
RunParms/	input data	runtime configuration parameters
Load/	executable	compiled model executable
Output/ [1]		
Animation1...60/	output data	multiple directories to hold map outputs
NCDF	output data	(optional) netCDF formatted map output
Budget/	output data	budgets and preset Performance Measures
Canal/	output data	canals and structures
Debug/	output data	debug-related
PtSer/	output data	point (cell) time series
$ELM_ARCHIVE_PATH/		
ELM2.8/		
MyFirstRun/		
VarNameA	output data	archived map output of VarNameA
VarNameB	output data	archived map output of VarNameB
VarNameXYZ	output data	archived map output of VarNameXYZ
NCDF	output data	(optional) netCDF formatted map output
Budget/	output data	archived budget and preset PMs
Canal/	output data	archived canal and structure summaries
Debug/	output data	archived debug-related files
PtSer/	output data	archived point (cell) time series
Input/	input data	archived input data (subset, parameter files)

[1] Output directory may be anywhere, including outside of $ELM_HOME

10.7.4 Software recommendations

In order to interpret input and output data, it is recommended that the user at least has access to the Open Source software of the GRASS GIS and the Open Office Calc spreadsheet system. Both are available as pre-compiled binaries for a number of computing platforms, and thus are very simply installed.

10.7.4.1 Geographic Information System (GIS)

The GRASS[10] GIS can be used to analyze model input and output data. GRASS excels in raster data processing and analysis, with many useful functions for landscape analysis. It also fully supports the vector (canal) and point (water control structures, monitoring locations) data required for ELM. Through the use of unix symbolic links between the GRASS and the ELM data directories, the ELM directly reads GRASS project data files (uncompressed binary data and text header) as model input. However, no GRASS-specific encoding of binary information is used, and thus the data files may be opened with any program that can read binary data arrays. Scripts are available to directly input and visualize the ELM canal vectors in GRASS. Other GIS and/or spatial mapping software tools can serve similar purposes.

10.7.4.2 Animated visualization

The GRASS GIS and its associated "xganim" animation program can be used to visualize animations of the output. We also use other tools, such as the Open Source OpenDX[11] and IDL[12] for such purposes, as both have advanced functionality relative to xganim.

10.7.4.3 Data management

While MySQL[13] is our targeted relational database system, we currently use the functionality of spreadsheet data systems in Open Office Calc[14] (which is fully compatible with Microsoft Excel). FileMaker Pro[15] has been used for a relational database system for parts of ELM, but will be entirely phased out with MySQL in the future.

10.7.4.4 Advanced scripting

Python[16] (and an associated graphics library PyChart[17]) is our choice for developing object-oriented, advanced script applications for post-processing the model and other tasks.

---

[10] http://grass.itc.it/ (Open Source)
[11] http://www.opendx.org/ (Open Source)
[12] http://www.rsinc.com/ (commercial)
[13] http://www.mysql.com/ (Open Source)
[14] http://www.openoffice.org/product/calc.html (Open Source)
[15] http://www.filemaker.com/ (commercial) 30-day trial version of the software
[16] http://www.python.org/ (Open Source)
[17] http://home.gna.org/pychart/ (Open Source)

10.8 Appendix 2: Unix & ELM Cheat Sheet

Handout used in ELM training classes

The User's Guide Chapter 10 of the ELM v2.8 Documentation assumes that you have some familiarity with running simple commands in the unix operating system. Because many in the ELM training classes do not have this background, we will necessarily have to learn just a bit about how to get around in unix. The ELM is created and used by invoking a variety of "scripts", which basically take care of most of the intricacies of unix convolutions. This cheat sheet will hopefully provide you with adequate information on the basics of unix commands, allowing you to get around in the file structure, and to run the scripts. (A Graphical User Interface can tie these scripts together so that a user never sees the "command line" of unix; Beheen Trimble (SFWMD) has created a prototype for ELM, but development on that has stopped.)

First, an overview of what kinds of "stuff" is available for using unix/linux.

Operating Systems

In case you are thinking about making a unix/linux box for your own use:
There are many flavors of the unix operating system: the most common these days is some variant of Linux that will run on any kind of computer based on the Intel/AMD (and other) processing chips. Below is a short-list of some of the common unix operating systems that are available.

- In class, we will be using RedHat Enterprise Linux (http://www.redhat.com), which is not now Open Source. The SFWMD IT folks did a great job of creating a "virtual" machine for our class, installing Open Source applications like QGIS, GRASS, and OpenDX for our use.
- The Open Source (free) development variant of RedHat is Fedora (http://fedoraproject.org/). Fedora is popular, and suitable for home or business use, and can probably be easily installed by inexpert users. I have not personally installed Fedora.
- Another popular Open Source (free) linux is SUSE (http://www.opensuse.org/), which also comes in an Enterprise ($$) version by Novell. I've made a dual-boot SUSE-WinXP machine from an old Intel Celeron computer at home – it's as easy as installing WinXP.
- The Apple OS X has Berkeley BSD unix at its core, providing full unix (~linux) capabilities to any Apple computer (right along with its famous GUI).
- Some agencies probably have Sun workstations lying around, which (usually) run the Solaris flavor of unix. There are Open Source versions of Solaris that people have installed on non-Sun machines.
- There are others – Debian linux, etc etc.

Applications

Essential: To install the ELM, you must at least have a compiler installed on your operating system (in order to make the model into an executable program). "Enterprise" versions of linux come with a compiler, but a "basic" (i.e., home-user) installation of something like SUSE linux requires that you obtain and install the Open Source "gcc" compiler.

To install an application like the gcc compiler, all of the popular linux versions have package managers (variously called RedHat Package Manager, YaST2, and Fink, for RedHat/Fedora, SUSE, and Apple, respectively). These package managers can make installations of applications often very easy, or almost trivial; on the other hand, it can take a fair amount of time to successfully install some applications if you have an "old" linux version and "new" applications (or vice verse), depending on the complexity of the application itself.

Optional: In class, we'll be using (or at least introduced to) the following Open Source applications. However, the ELM does not depend on the use of any of these – use the equivalent application that you are comfortable with.

- Open Office (http://www.openoffice.org/) – a highly functional equivalent of Microsoft Office suite, including spreadsheet, word processor, etc, that works quite seamlessly on MS Office files.
- GRASS (http://grass.itc.it/) - GIS for raster, vector, polygon data, that is very efficient for modeling on regular-grids, with many scientific analysis routines.
- QGIS (http://qgis.org/) – another GIS, which we merely use as a GUI to run GRASS commands to view ELM maps (never actually needing to run GRASS itself in class)
- OpenDX (http://www.opendx.org/) – an advanced application for visualization (incl. animation) of spatial data
- XEmacs (http://www.xemacs.org/) – we're not using XEmacs, but Emacs, a pared-down version of this programmer's text editor. Some form of a good text editor is essential for modeling.
- [Python (http://www.python.org/) – an object-oriented scripting language that was used to develop an ELM post-processing system, but which we are not using in this class.]
- [FileMaker (http://www.filemaker.com/) - COMMERCIAL product, Windows/Mac only – a relational database program that is used for maintaining the ELM database of water control structures. We have yet to get around to replacing this

database system with the Open Source mySQL (http://www.mysql.com/) database methods. We won't be using Filemaker in the class, though a free demo version of the software is available.]

Linux graphical user interface

Different unix/linux operating system versions have different types of graphical user interfaces, most of which are intended to make WindowsXP users feel like they're in familiar territory. Along those lines, some version of the equivalent of Windows File Explorer is always available – for example, double-click on the desktop icon representing your "home" directory to navigate the file directory. Moreover, if a filename has an extension that is associated with an installed application, you can double-click on the file to open it in the application (like Windows/Mac).

Depending on the graphical interface, there is something similar to the "Start" menu in Windows, allowing you to open any installed application from that menu. Be aware, however, it is possible that an application was installed without being added to this menu (like Windows/Mac). Enough said on the GUI for the operating system – explore.

Unix command-line basics

To run ELM, you must have the unix command line available to you. There are a variety of ways to get to this "window", but you should at least find the application called "Terminal" somewhere in the GUI's Start-menu equivalent. Once you have this Terminal window open, the following are some useful hints. For a variety of reasons, some people (including me) prefer to work unix commands from an xterm (or X terminal, which is associated with the unix "X Window System" for graphical output).

The Linux graphical user interface can accomplish all of the below tasks, without "resorting" to the unforgiving command line. But you should become familiar with these basics, and thus be comfortable when you need to run a script, like "`go ELM2.8`" (type "`go ELM2.8`", no quotes, at the command prompt, then hit the Enter/Return key – see ELM hints later in this document).

Syntax: the "`prompt>`" below is whatever text (prompt) is staring at you when you look at a unix terminal screen
Syntax: unix is case-sensitive – the command "cp" is different from "CP" (there is no "CP" command)
Syntax: after you type the letters/words to form a command, hit Enter/Return to execute the command
Syntax: {ignore anything written below that is within the braces "{garb}" - it's not part of command}

How does this command work?
man – Manual pages – be told the syntax and results of any unix command
`prompt>man ls` {shows the manual pages for the command "`ls`"; hit the space-bar to advance one page in the manual; hit the letter "`q`" to quit the manual}

Open an X Window Terminal
xterm – open up an X terminal, which allows graphics commands (vs. the plain "terminal" that you started with)
`prompt>xterm &` {run the application called "`xterm`", with the "`&`" allowing you to later type other commands in the terminal window you typed this command from}

Where am I?
pwd - Print Working Directory - be told the directory path to where you are - your current directory
`prompt>pwd`

What's in this directory?
ls - LiSting - be told what files/directories are inside your current directory (i.e., where you are)
`prompt>ls` {basic, brief information}
`prompt>ls -l` {give more detailed information, like time of modification, file size}
`prompt>ls -a` (show even files that are "hidden" (hidden file names start with a dot ".")}

Change your location to another directory.
cd - Change Directory - change into another directory location that you specify
`prompt>cd ThisDirectory` {moves you into `ThisDirectory`, which must be a sub-directory in current directory}
`prompt>cd ThisDirectory/ThatDirectory` {moves you into `ThatDirectory`, which must be a sub-directory of `ThisDirectory` in your current directory}
`prompt>cd ..` {moves you into the "parent" directory that contains your current directory}
`prompt>cd .` {moves you into the "current" directory (i.e., does nothing! Just to show what a single period means in unix paths)}

Move a file.
mv - MoVe – move (don't copy) a file from a source location to a destination location

`prompt>mv thisFileName subdir/thisFileName` {`thisFileName` is moved into the `subdir` subdirectory within your current directory}

Copy a file.

cp - CoPy – copy a file from a source location to a destination location

`prompt>cp thisFileName subdir/thisFileName` {`thisFileName` is copied into the `subdir` subdirectory within your current directory}

Delete a file.

rm - ReMove - permanently delete a file

`prompt>rm thisFileName` {thisFileName is gone forever after you enter the command}

Change permissions on a file.

chmod – change the permissions of a file to allow reading, writing, and/or executing, by different type of users

`prompt>chmod u+x thisFileName` {`thisFileName` now can be executed (x) by the owner/user (u) }

`prompt>chmod g+w thisFileName` {`thisFileName` now can be written-to (w) by a particular group (g) }

`prompt>chmod g-w thisFileName` {`thisFileName` now can NOT be written-to (w) by a particular group (g) }

Unix command-line tips for ELM (see Chapt 10 of ELM documentation)

Verify that you have your `ELM_HOME` set up (unix command, unix environment variable).

`prompt>echo $ELM_HOME` {echos (prints to screen) the path of your `ELM_HOME` environment variable; the "`$`" tells unix to look up the definition of the word that follows}

Build the model (ELM script).

`prompt>build ELM2.8` {compile/link the ELM v2.8 code to create an executable program}

Check (and edit if you want) the model runtime configuration (ELM script).

`prompt>Check ELM2.8` {shows you the current `Driver.parm` file's runtime parameters, and any output requests found in the `Model.outList` file (i.e., where the output frequency is greater than every zero iterations during the run) }

Go run the model – no frills (ELM script).

`prompt>go ELM2.8` {run the ELM v2.8 executable program, using whatever parameters are currently saved}

Archive the model output (ELM script).

`prompt>ArchiveRun ELM2.8 myFirstRun` {moves the model output from your `$ELM_HOME/SME/Projects/ELM2.8/Output/` directory, to a new directory at `$ELM_HOME/arc_out/ELM2.8/myFirstRun/` }

Run the model and archive the output (ELM script – see note on basic "vi" text editing below).

`prompt>Run ELM2.8 myFirstRun` {invokes the above `Check`, `go`, and `ArchiveRun` scripts, but also creates some copies of important parameter files that will be archived by `ArchiveRun`, and forces you to write some metadata on what is important about the simulation run}

OpenDX animation of the model output (invoke three ELM scripts – *these are new, and not fully documented w/ ELM 2.8*).

`prompt>DXsetup ELM2.8 myFirstRun SfWatAvg A` {creates the first, "`A`" (uppercase), set of header files for animating the ELM2.8 archived run named `myFirstRun`, for output variable named `SfWatAvg`}

`prompt>DXsetup ELM2.8 myFirstRun TPSfWatAvg B` {creates the second, "`B`" (uppercase), set of header files for animating the ELM2.8 archived run named `myFirstRun`, for output variable named `TPSfWatAvg`}

`prompt>DXanim ELM2.8 myFirstRun SfWatAvg TPSfWatAvg` {invokes the OpenDX program for a 3D animation of the above two variables in the ELM2.8 archived run named `myFirstRun`. **Note**: you could have previously run the `DXsetup` scripts on any/all of the output variables, and animate any combination from within OpenDX at this point – you do NOT have to re-run `DXanim` multiple times for multiple variables}

The "Run" script above forces you to use "vi" to write some metadata on what is new about the simulation run. Below are the steps to take in doing this:

So, if you must use vi to edit a text file: (use Xemacs or something else unless you are forced to use vi)

vi - Very Incomprehensible :-) - a standard text editor used in unix; there are many other options

`prompt>vi myTextFile.txt` {opens up the "`vi`" application, editing the file called "`myTextFile.txt`"

--in vi, there is no prompt. Use the following "commands" to do basic stuff (use lower-case)

First: type the letter "`i`" for insert mode

Second: type your message/information text for the file (it will all be one long line, unless you type Enter/Return to split lines/paragraphs)

Third: type the "`esc`" (Escape) key to escape/exit out of the insert mode

Fourth: if you made a mistake, use the arrow keys to move the cursor to a location of text

Fifth: to delete a character where the cursor is, type the letter "`x`"

Sixth: repeat the First – Third steps above to insert text

Seventh: type the "`:`" (colon) key, and see a colon show up at bottom of window

Eighth: type the letter "`w`" for write/save, followed by the letter "`q`" for quit to quit vi

Stuck???: within vi, try hitting the "esc" key several times, then type the ":" followed by "`q!`" (letter q followed by an exclamation mark) to quit without saving your work

Old saying: When in doubt, read the manual... (update: Google it).

10.9 Appendix 3: Acquiring SFWMM structure flows

Class handout:

Acquisition of SFWMM water control structure flows for input to an ELM simulation

The body of the User's Guide Chapter 10 of the ELM v2.8 Documentation does not describe the automated methods available to couple the SFWMM to the ELM for future-project simulations. The User's Guide (p. 10-17) provides an overview of the data files that likely will need modification for future-project simulations, but does not discuss the code/scripts that were developed for that task. This Appendix documentation sheet is a beta version of documentation of those tools and methods, and is intended for use by those who participate(d) in recent (2007) training on ELM application.

There are tools available to (largely) automate this process of getting daily flows from SFWMM water control structures, for input into an ELM simulation (primarily for a future-project alternative, where all managed flows that drive the ELM simulation are output from the SFWMM). There are two code utilities for this process:

a) "binStrNames.linux" (or .darwin, etc, depending on the operating system): this binary executable (compiled by the user, from C source code) is used to match water control structures within the ELM domain that are "believed" to be needed from the SFWMM output. Thus, the user must still have sufficient knowledge of the SFWMM to know which structure names *should be* used in a particular future-project alternative run of the SFWMM. The outputs are a list of positive (ELM-SFWMM) matches of structure names, and a list of non-matching structure names.

b) "getDSSflow": this unix shell script uses the list of matching water control structure names from a) above, and calls two (pre-installed) HEC DSS programs in order to extract the requested daily flow data from the "DSS" database of outputs from the SFWMM (HEC DSS includes a set of tools that are binary executables that have been previously installed, from US COE, or SFWMD). The output is an ascii file of all ELM-required daily flows, from the SFWMM output during the requested period of simulation.

Because these tools and methods still require fairly significant knowledge of regional water management infrastructure and the associated SFWMM data, and in particular knowledge of the naming conventions for SFWMM water control structures in a future-project SFWMM simulation, they have not been previously documented in the ELM v2.8 calibration-validation (historical) application release.

Quick-steps

Basic procedures for developing the water control structure input files, with references to the full descriptions of the file content and formats in Chapter references to *Documentation of the Everglades Landscape Model: ELMv2.8.* The following input data files will be edited or generated, and will be expected in your $ELM_HOME/SME/Projects/ELM2.x/Data/ directory:

- CanalData.struct - Water control structure names and attributes (Data, Chapter 4, pg 4-12)
- CanalData.struct_wat – Daily water flows for all managed water control structures (Data, Chapter 4, pg 4-18)
- CanalData.struct_TP – Optional: Daily phosphorus (TP) concentrations in water flows for all managed water control structures that introduce water into the ELM domain (Data, Chapter 4, pg 4-20)
- CanalData.struct_TS – Optional: Daily salt (TS, chloride being used) concentrations in water flows for all managed water control structures that introduce water into the ELM domain (Data, Chapter 4, pg 4-21)

1) ***Install*** *the programs (attached handout "Table 1. ELM Directory Structure: Update...").*
2) ***Select*** *the water control structures necessary for the simulation, developing the input data file "CanalData.struct"*
3) ***Acquire*** *the daily water control structure flow data for all managed flows from the SFWMM "DSS" file, creating the input data file "CanalData.struct_wat".*
4) ***Develop*** constituent concentrations for domain-inputs.
5) ***Run*** the ELM simulation, verifying that water control structure flows among basins match those of the SFWMM.

1) INSTALL - programs

Because these programs were not in the ELMv2.8 distribution, they were not automajically installed for you during your normal ELM installation.

A) The "getDSS" and "catDSS" binary executables **MUST BE previously installed** on your computer to acquire SFWMM data from a "DSS" database file (which will be done as part of the below process). You need to get these programs (and generally other DSS tools) using the appropriate agreements that your agency has with the US COE or SFWMD.

B) The "binStrNames.linux" program is installed from the C code and script contained in the "StrNames.tar" file, found within the "add_onsB_sfwmm" directory of the ELM Training distribution. **Be in a unix shell that contains the correct definition of your "$ELM_HOME" environment variable and path pointing to ELM scripts**[18]. First, untar the archive file into any convenient directory, and then run the script named "StrNamesCreate" that was unpacked into your current directory:

   a. your_unix_prompt> tar –xvf StrNames.tar
   b. your_unix_prompt> StrNamesCreate
   c. The result will be a compiled, binary executable named "binStrNames.linux" (or another extension, depending on your operating system) that is placed into your "$ELM_HOME/SME/scripts" directory, and thus will be executable from any directory for later use.

C) The "getDSSflow" unix shell script is simply copied from the "add_onsB_sfwmm" directory of the ELM Training distribution. (It is not necessary for this simply copy, but it may be convenient to follow the steps indicated below, and thus be in a unix shell that contains the correct definition of your "$ELM_HOME" environment variable as in B) above). Simply be in the "add_onsB_sfwmm" directory of the ELM Training distribution:

   a. your_unix_prompt> cp getDSSflow $ELM_HOME/SME/scripts
   b. The result will be the "getDSSflow" script in your "$ELM_HOME/SME/scripts" directory, and thus will be executable from any directory for later use.

2) SELECT - ELM water control structure names & attributes (CanalData.struct)

For the selection process, the first step is to assemble a (ascii) data file of water control structure names and attributes for (eventual) input to the ELM to run the intended future-project application. This text file, named "CanalData.struct", will initially be input to the "binStrNames.linux" program, to see if those water control structure names are available from the particular SFWMM simulation (DSS output file).

A) Develop your best estimate of the structures needed for your simulation.

- *Database manipulation:* Filemaker (commercial software[19]): if you have the Filemaker database program, you can make use of the existing database of ELM structures, with all of its documentation for "real-world" and SFWMM-named water control structures, for historical and CERP applications. The database files themselves are in your $ELM_HOME/SME/Projects/Dbases directory, as discussed in the ELM v2.8 documentation Chapters 4 and 10. Queries are pre-established for a variety of (older) CERP simulations; export the tab-delimited text file ("CanalData.struct" will be the input name to ELM) when the desired structures have all been created/selected for a particular simulation.
- *Text-file manipulation:* if you do not use Filemaker, you can modify a copy of the "CanalData.struct" input file of the ELM that was distributed in the default ELM2.8 distribution (that you have run for historical, calibration/validation simulations). Using your knowledge of what water control structures are simulated within the Everglades (ELM domain) by the SFWMM for this particular future-project simulation, assemble a text data file that has the required (see Chapter 4) attributes for the water control structures that you believe will be in this particular SFWMM simulation.

B) Find matches and non-matches between ELM and SFWMM structure names.

- *ELM_HOME:* Be in a unix shell that contains the correct definition of your "$ELM_HOME" environment variable and path pointing to ELM scripts (see above).
- *SFWMM DSS file:* go into the directory containing the SFWMM output file for the daily water control structure flows; a common name for this output file is "daily_str_flw.dss"; the associated DSS catalog file with the suffix *".dssd" must be present*[20].
- *ELM "CanalData.struct" file:* make a copy of (or, preferably, a symbolic link to) the CanalData.struct" file that you just created, and place it into the directory containing your SFWMM data file(s). Give it a descriptive suffix, such that you end up with a filename that is informative for the scenario, such as "CanalData.struct.D13R"
- *Match names:* run the program "binStrNames.linux" from the command line prompt in this directory:

  prompt> binStrNames.linux

[18] As covered in class: Typing the command "echo $ELM_HOME" should point to your ELM_HOME directory; if in doubt, re-run the installation script "ELMinstall.sh", which will show you how to re-set your ELM_HOME, and add the path to all of your scripts for ELM to your unix $path environment variable.

[19] See the "UnixELM_CheatSheet.doc" for information on getting the software for a 30-d free trial.

[20] Run the "getDSS" program once in this directory to quickly create a catalog if it is not present.

Follow the on-screen prompts. The result will be two files, each with the suffix "descriptor", which is the descriptive suffix you provided above:

- "dss_structs.descriptor" is a simple list of water control structure names that were found in both the ELM and the SFWMM files
- "noMatch.descriptor" is a list of water control structure names that were requested (via the ELM data file), but which were not found in the SFWMM file

C) Iterate A) and B) above until you have assembled the "final" list of water control structure names, for which you will acquire daily flow data from the SFWMM output (DSS) file. This selection process indeed requires that you know (or guess) which SFWMM structures will be needed for an ELM simulation, and will also be present in the SFWMM DSS file: **there is a significant dependence on your knowledge of the SFWMM documentation, and your communication with the developers of the particular SFWMM simulation run**. The utility of the structure name matching program is in assuring a documented matching of the structure flows that you acquire from the SFWMM DSS file, and those structures that you are using in ELM. The sequence of structure names in the CanalData.struct attribute file must match the sequence of structure names in the daily flow time series CanalData.struct_wat file (that will be generated in the next step). If you do not use the automated selection process, you will end up with your "own" third list of structures that you will have to compare with the ELM's CanalData.struct CanalData.struct_wat input files, in the correct sequence. With on the order of 100 structures to keep track of for one simulation, it is highly recommended that you use this matching tool to very rapidly accomplish what can otherwise become a complex task, all the while maintaining good documentation of your ELM simulation.

D) Copy this (tab-delimited) water control structure attribute data file into (or preferably make a symbolic link to) your $ELM_HOME/Projects/ELM2.x/Data/ directory, with the name "CanalData.struct".

3) ACQUIRE - SFWMM flow data for ELM-required water control structures (CanalData.struct_wat)

To acquire the necessary managed flow data from a SFWMM simulation run, you must use some of the HEC DSS tools (that you have installed on your system). There are single DSS programs that could be used to extract individual, or relatively small sets of, water control structure flows from a DSS file. However, the most efficient method of extracting ca. 100 water control structure flows is the unix shell script provided here[21], which calls two HEC DSS programs. Moreover, this scripted method uses a text input file of structure names (instead of user typing or cut&pasting), and lends itself to more complete automation and incorporation into a GUI framework in the future.

A) Extract daily flow data for the list of structure names developed in step 2) above.

- *ELM_HOME:* You should still be in a unix shell that contains the correct definition of your "$ELM_HOME" environment variable and path pointing to ELM scripts (see above).
- *SFWMM DSS file:* You should still be in the directory containing the SFWMM output file for the daily water control structure flows; a common name for this output file is "daily_str_flw.dss".
- *Acquire flow data:* run the script "getDSSflow" from the command line prompt in this directory, supplying the two arguments of the 1) beginning year and 2) ending year of the period for which you desire flow data[22]

  prompt> getDSSflow 1965 2000

  Follow the on-screen prompts. For use in ELM, you should not select any special formatting, nor select for including the DSS header in the resulting output file. The result of the script will be one ascii file of daily flows for all water control structures.
  - "user-named-flowname.user-suffix" contains a record (line) for each day, column-1 = year, column-2 = month, column-3 = day, followed by sequence of columns for each water control structure's daily flows (daily mean cfs)

B) Supply two header lines to flow data file "user-named-flowname.user-suffix".

- Import this text data file into a spreadsheet program (OpenOffice Calc, MS Excel), with a separate spreadsheet column for each column of (date and flow) data. Insert two lines at the top of the file, as follows:
- *Line 1:* The first continuous string (word) should be the brief scenario name that must match the scenario name you will enter into the $ELM_HOME/Projects/ELM2.x/RunParms/Driver.parm runtime configuration file; the remaining text on this line can be used for any (unformatted, any column) description of the source of the data. You should include units of the data, version of the SFWMM, and other important metadata.
- *Line 2:* In column 1 (A), enter the text string of "Year" (no quotes); then enter "Month" and "Day" in the next two columns. Then paste the list of structure names generated earlier, "dss_structs.descriptor", to fill the remaining columns of line 2.

---

[21] As noted in the script code, the original script was developed by Lehar Brion of the SFWMD, and was modified for the current application use.

[22] Flow data will be extracted inclusive of Jan 1 of the beginning year , and Dec 31 of the ending year. Currently (SFWMM v5.x, ELM v2.x), the simulation period spans 1965 – 2000; see the DSS catalog file.

The line 2 column headings should define every column of data when viewed in the spreadsheet. Save this file as a tab-delimited text file. The result is a tab-delimited (from line 2 through the last line) ascii data file.

The manual addition of the header lines can be readily automated in later versions of the ELM.

C) Copy this tab-delimited flow data file into (or preferably make a symbolic link to) your $ELM_HOME/Projects/ELM2.x/Data/ directory, with the name "CanalData.struct_wat".

4) DEVELOP – constituent concentrations for domain-inputs (optional, CanalData.struct_TP, _TS)

All managed water control structure flows that introduce "new" water into the domain of the ELM must be associated with concentrations of phosphorus and chloride. Develop those constituent concentrations, then **either**:

- *develop daily phosphorus and chloride concentrations for flows into (not within) the domain (and thus develop the input files CanalData.struct_TP, CanalData.struct_TS) – for each structure that will be using a time series of concentrations, enter the string "tser" (no quotes) into the concentration field(s) for the water control structure in the CanalData.struct attribute file*
- ***OR** simply assume constant concentrations through time, and enter those concentration values in the respective fields for each structure in CanalData.struct (the _TP, _TS files are not needed if no time series are used at all).*

If using time series for some, or all, domain-inflow structures' constituent concentrations, the expected format of the _TP and _TS files is directly analogous to the _wat (water flow) input file; the input files for the ELM v2.8 distribution Project can be used as examples. If used, these data files are expected in the same Project Data directory as the other CanalData.struct files.

5) RUN – the ELM simulation, verifying that budget- flows match the SFWMM

The monthly budget output files from the SFWMM and the ELM are vital resources to use in determining whether the flows used in the ELM match those used in the SFWMM. While it is (rarely) possible to have new/unknown SFWMM structures that define "within-basin" flows that do not "make it" into the ELM simulation (and cannot be detected by budget analysis), it is **essential to at least compare the input-output budgets between the ELM and the SFWMM**, for each major hydrologic basin, at least for water control structure flows. There are a variety of other budget- consistency checks that should be conducted (e.g., levee seepage, ET, etc), but it is essential to verify that the water control structure flows in the ELM budgets for all hydrologic basins match those of the associated SFWMM simulation.

We have another (C code) program that converts the page-based table format of the SFWMM budget output file into a monthly-record based format that is directly usable in statistical or spreadsheet-based programs (as used in the ELM budget output files). We also have a spreadsheet template that will accept ELM and (reformatted) SFWMM budgets, providing annual summaries of all input and output budget flows per basin for ELM and for SFWMM, comparing them in graphical and tabular forms. These tools require a number of levels of user-intervention, are not always user-friendly, and are not documented for current application release or current training.

Thus, it is up to the user to use their preferred method to evaluate the two budget files, from the ELM and the SFWMM, to verify that all among-basin water control structure flows match between models.

NOTE ON WATER QUALITY MODELING and SFWMM STRUCTURE FLOWS: One of the complexities of water quality modeling involves the need to identify the upstream source water for every specific flow associated with an input of "new water" into the model domain. The SFWMM partitions the total flow for some structures (such as S-8, named "S8") into multiple contributing source flows, with the different sources being Everglades Agricultural Area runoff, multiple Stormwater Treatment Area outflows, etc. Thus, the ELM (or other water quality models) necessarily must use multiple SFWMM structure flows (and thus structure names) for a single "real-world" structure (such as S-8) that has multiple sources. This means that the SFWMM budget will often contain one total flow for a particular structure (e.g., "S8") in a basin budget, while the ELM structures that were used for the same total flow were actually other, multiple, SFWMM structures (that do not appear in the SFWMM budget) – and the sum of these multiple structure flows must equal the total flow through the "real-world" structure (e.g., "S8"). Applications of ELM, or any other water quality model, must carefully consider the documentation of each SFWMM flow, considering the source of any and all flows.