Sea Level Rise Modeling for the SAMBI Designing Sustainable Landscapes Project

Matthew J. Rubino

Biodiversity and Spatial Information Center, NC State University, Raleigh, NC 27695

ABSTRACT

The Biodiversity and Spatial Information Center is modeling landscape scale changes to avian habitats based on various climate change scenarios within the South Atlantic Migratory Bird Initiative (SAMBI) geographic planning region. In coastal areas, the Sea Level Affecting Marshes Model (SLAMM) is being utilized to incorporate marsh migration dynamics due to longterm sea level rise (SLR). Inputs to SLAMM include National Wetlands Inventory (NWI) data cross-walked to 22 categories; a digital elevation model (DEM) with 30 meter resolution developed from the National Elevation Dataset (NED); slope derived from the DEM; an impervious surface data layer created for the National Land Cover Dataset (NLCD); tidal datum and sea level rise trend data from NOAA National Ocean Service's Center for Operational Oceanographic Products and Services (NOS/CO-OPS) stations.

The entire coastal extent (Atlantic and Gulf coasts) of the SAMBI was modeled with split processing using 39 USGS 8-digit Hydrologic Units Code (HUC) boundaries for coastal watersheds. NWI was rasterized from polygon data to a 30 meter cell size matching the NED DEM resolution. NWI was conflated with developed/urban classes within the Southeast Gap Analysis Project's land cover map and then cross-walked to 21 categories per SLAMM documentation (mangrove category excluded). Impounded or diked features were identified within the NWI and incorporated in modeling. Areas of inconsistent, erroneous, or systematically flawed data in the NED were identified visually and "fixed" using a variety of other data sources. Higher accuracy Light Detection and Ranging (LiDAR) elevation data were incorporated exclusively (i.e. in place of NED) for coastal regions in North Carolina because of the statewide availability of LiDAR. Tidal datum, present epoch (1983-2001) data and SLR trend data were downloaded from the NOAA NOS/CO-OPS website. Multiple stations may be located within any of the HUC boundaries along the SAMBI extent. Therefore, tidal datum data was used for the station whose values were closest to mean values for stations adjacent or within a HUC. Because not all stations have SLR trend information, data was taken from stations most proximal to a HUC.

The model was run using four climate scenarios (A1B, A2, B1, and A1FI) in 10 year increments for years 2000 to year 2100 using the "protect developed" option. Each of the 39 8-digit HUCs was run separately and then merged to create seamless decadel maps by climate scenario for coastal areas from southern Virginia to northern Florida.

INTRODUCTION

As part of the Designing Suitable Landscapes for Bird Species in the Eastern United States project, the Biodiversity and Spatial Information Center (BaSIC) is modeling landscape scale changes to habitat based on various climate change scenarios within the South Atlantic Migratory Bird Initiative (SAMBI) geographic planning region. Landscape change is being assessed using a combination of the USGS SLEUTH model for urban growth; vegetation dynamics using a spatially explicit stochastic state transition model (VDDT/TELSA) for ecological systems in the Southeast Gap Analysis (SE-GAP) Land Cover map; and sea level rise with the Sea Level Affecting Marshes Model (SLAMM) (Park et al. 1986). SLAMM attempts to simulate transforming coastal environments accounting for nearshore geomorphological processes such as accreation, erosion, and marsh migration dynamics due to longterm sea level rise. To this end, the SLAMM approach is much more robust than a non-dynamic "bathtub" model wherein an increase in ocean water levels simply inundates land.

METHODS

SLAMM Model Inputs

SLAMM was developed as a stand-alone program with a simple one form interface for Windows operating systems. The most recent version at the time BaSIC began modeling in the SAMBI extent was version 5.0.1 (Clough 2008). Six inputs are required to run SLAMM for a given geographic region:

- National Wetlands Inventory (NWI) data cross-walked to 21 categories (Table 1.)
    - A separate data layer for dikes/impoundments can be created from NWI to incorporate these features in modeling
- A digital elevation model (DEM)
- Slope derived from the DEM
- An impervious surface data layer
- Tidal datum information (usually obtained from NOAA National Ocean Service's Center for Operational Oceanographic Products and Services (NOS/CO-OPS) stations)
- Sea level rise (SLR) trend data (also obtained from NOAA NOS/CO-OPS stations)

USGS Hydrologic Unit Codes (HUCs)

The HUC is part of a hierarchical classification system for surface water drainage in the US (Seaber et al. 1987). The numerical code represents a "cataloging unit" of delineation approximately larger than 1800 square kilometers. Within the SAMBI project boundary there are 39 coastal 8-digit HUCs (Figures 2 and 3). HUCs allow for partitioned processing when scripting complex spatial analysis tasks including SLAMM input data development as well as SLAMM modeling itself (see appendices at the end of this document).

National Wetlands Inventory

The NWI digital data of wetlands and deepwater habitats is produced by the United States Fish & Wildlife Service. Data production of polygons derived from aerial photograph interpretation began in 1977 in conjunction with the classification system developed by Cowardin et al. (1979). Generally, the Cowardin classification begins with marine, estuarine, riverine, lacustrine, and palustrine environments where coding starts with the first letter of each system (i.e. M, E, R, L, and P respectively). Subsequent portions of the code account for subsystem, class, subclass, and potentially up to four additional modifiers. For example, a polygon coded E1UB4L6 breaks down as follows:

E = estuarine system	4 = organic subclass
1 = subtidal subsystem	L = subtidal water regime modifier
UB = unconsolidated bottom class	6 = oligohaline coastal halinity modifier

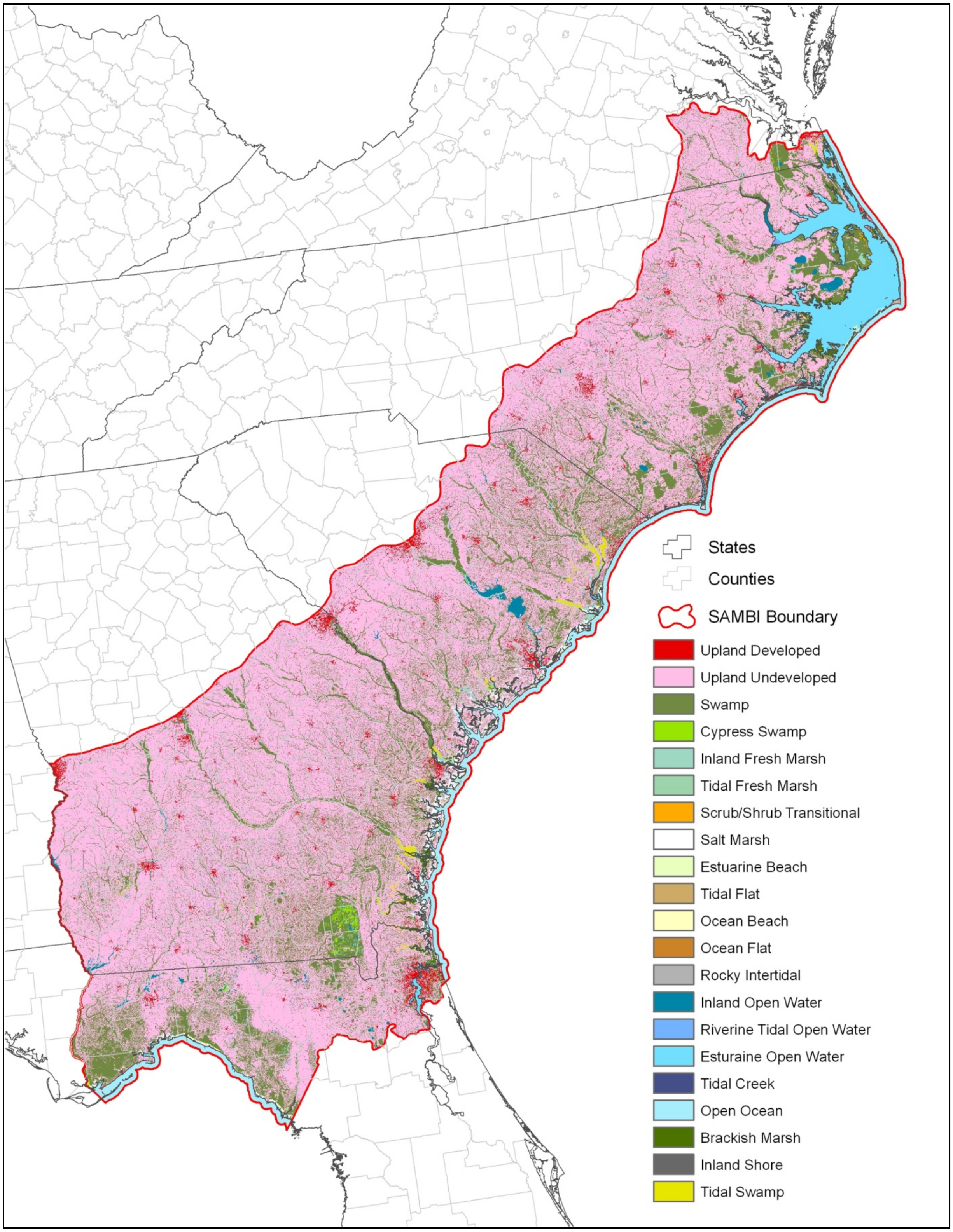
BaSIC compiled NWI data for the SAMBI and converted NWI polygons to rasters with cell values recoded to a 6-digit numeric system derived from the NWI code. This numeric code was then cross-walked to 21 categories according to SLAMM documentation (Table 1., Figure 1.) Developed Dry Land, and Undeveloped Dry Land categories (1 and 2 respectively) were extracted from the SE-GAP project land cover (see BaSIC documentation regarding this data layer). No NWI polygons included the estuarine intertidal forested/shrub scrub systems encompassing the Mangrove SLAMM category due to the SAMBI region extending only as far south as northern Florida.

An option in SLAMM allows for inclusion of dike features within the modeled extent. BaSIC modeling incorporated dikes as a separate data layer using NWI data coded with an "h" modifier.

Table 1. NWI- SLAMM cross-walk with descriptions.

Raster Cell Code	SLAMM Category Name	NWI Code(s)	Description
1	DevDryland	U	Upland—Developed
2	UndDryland	U	Upland—Undeveloped, categories 1 & 2 need to be distinguished manually
3	Swamp	PFO, PFO1, PFO3-5, PSS, Ps	Palustrine forested (living or dead), and scrub shrub. Also, Palustrine forest and scrub-shrub with tidal influence.
4	CypressSwamp	PFO2	needle-leaved deciduous
5	InlandFreshMrsh	L2EM,PEM[1&2] ["A"-"I"] ,R2EM	Lacustrine, Palustrine, and Riverine emergent
6	TidalFreshMarsh	R1EM, PEM["K"-"U"]	Riverine tidal emergent
7	Scrub Shrub / Transitional Marsh	E2SS1, E2FO	Estuarine intertidal scrub-shrub broad-leaved deciduous
8	Salt marsh	E2EM, [no "P"]	Estuarine intertidal emergent [won't distinguish high and low marsh].
9	Mangrove	E2FO3, E2SS3	Estuarine intertidal forested and scrub-shrub broad-leaved evergreen
10	Estuarine Beach	E2US2 or E2BB (PUS"K")	Estuarine intertidal unconsolidated shore sand or beach-bar, includes salt pans
11	TidalFlat	E2US[N,3,4,M] E2FL,M2AB,E2AB	Estuarine intertidal unconsolidated shore mud/organic or flat
12	Ocean Beach	M2US2, M2BB/UB/USN	Marine intertidal unconsolidated shore sand
13	OceanFlat	M2US3or4	Marine intertidal unconsolidated shore mud or organic (Low energy coastlines)
14	RockyIntertidal	M2RS, E2RS, L2RS, [E,M]2AB1[N,P]	Marine intertidal rocky shore
15	InlandOpenWater	R3-UB R2-5OW L1-2OW,POW,PUB R2UB, (L1-2,UB,AB) PAB, R2AB	Riverine, Lacustrine, and Palustrine open water and aquatic beds
16	RiverineTidalOpenWater	R1OW, R1RB, R1UB	Riverine tidal open water
17	EstuarineOpenWater	E1,(PUB"K" no "h")	Estuarine subtidal
18	TidalCreek	E2SB, E2UBN	Estuarine intertidal stream bed
19	OpenOcean	M1	Marine subtidal
20	BrackishMarsh	E2EM[1-5]P	Irregularly Flooded Estuarine Intertidal Emergent
21	TallSpartina	N / A	2M buffer automatically added to Salt Marsh fringe
22	InlandShore	L2UD, PUS, R[1..4]US/RS	shoreline not pre-processed using Tidal Range Elevations
23	TidalSwamp	PSS,PFO"K"-"V"/EM1"K"-"V"	Tidally influenced Swamp.
NOTE:		NWI codes with an "h" modifier indicate diked/impounded areas	

Figure 1. NWI cross-walked to 21 SLAMM categories for the SAMBI extent.



Digital Elevation Model (DEM) - National Elevation Dataset

The USGS National Elevation Dataset (NED) consists of the most recent, highest quality data assembled for the United States at a spatial resolution of 30 meters (USGS 2003). However, because of the varying quality of the data, it was necessary to incorporate other datasets to create an improved, region wide product. These included data from NASA's Shuttle Radar Topography Mission (SRTM) at 30 meter resolution (NASA 2009), and hypsography data from the USGS's Digital Line Graphs (DLGs) at 1:24,000 and 1:100,000 scales (USGS 2009). Areas of inconsistent, erroneous, or systematically flawed data were identified visually and "tagged" for fixing. A number of algorithms were then used to reassign elevation values using interpolations based on the higher quality data. This essentially promoted the best available information for a given area using a number of sources, as opposed to re-interpolating data from the same flawed source.

The entire state of North Carolina has been mapped using Light Detection and Ranging (LiDAR) data from the North Carolina Floodplain Mapping Program (NCFMP 2009). Because the quality (precision, accuracy, and spatial resolution) is signicantly greater than the NED, BaSIC used only LiDAR derived elevations for the North Carolina portion of the SAMBI. These data were available at a 20 foot spatial resolution in a floating point decimal format and were resampled to 30 meters with a bilinear interpolation method (floating point decimal format was maintained).

Slope – derived from a digital elevation model

Slope is calculated from a DEM by determining the maximum rate of change (rise over run) between a given cell and its eight neighbors. Required input data format for SLAMM is slope measured in degrees.

Impervious Surface - National Land Cover Dataset

The National Land Cover Dataset (NLCD) was developed from LANDSAT Thematic Mapper satellite imagery to identify major categories of land use and cover throughout the United States at a resolution of 30 meters (Homer et al. 2004). Included during that development was a layer of impervious surface mostly representing urbanized areas. Each pixel was classified into 101 potential values (0-100%) based on reflectivity. Clusters of pixels with impervious surface percentages > 20% roughly correspond to human dominated environments. For SLAMM modeling, dry land with percent impervious > 25% is assumed to be "developed dry land."

Tidal Datum & Sea Level Rise Trend Information

SLAMM requires several parameter inputs compiled into a single text file (referred to in SLAMM documentation as a "site" file) with a specific format. Because the SAMBI was divided into 39 coastal HUCs to facilitate region wide processing, each HUC was a unique site with a unique input site file. Below is an example of a site file with appropriate formatting. The file (HUC3010107Site.txt) must include the "Site" text within the name:

```
SLAMM 5.0 SITE DATA:  fill in below data without deleting the category
text:  the delimiting comma must remain and no addtl carriage returns!

Description                               , Albemarle Site (HUC 03010107)
DEMDate (yyyy)                            , 2001
NWI_photo_date (yyyy) {if no grid file}   , 1982
Direction_OffShore (N|S|E|W)              , E
Historic_trend (mm/yr)                    , 2.82
NAVD88_correction (MTL-NAVD88 in meters)  , 0.652
Water Depth    (meters below MLW )        , 2.0
TideRangeOcean (meters: MHHW-MLLW)        , 0.183
TideRangeInland (meters)                  , 1.5
Mean High Water Spring (m above MTL)      , 0.063
Marsh Erosion (horz meters/year)          , 2.0
Swamp Erosion (horz meters/year)          , 1.0
TFlat Erosion (horz meters/year)          , 6.0
Salt marsh vertical accretion  (mm/yr)    , 1.9
Brackish Marsh vert. accretion (mm/yr)    , 4.3
Tidal Fresh vertical accretion (mm/yr)    , 4.8
Beach/T.Flat Sedimentation Rate (mm/yr)   , 0.5
Frequency of Large Storms (yr/washover)   , 25
Use Elevation Preprocessor for Wetlands   , TRUE

Put any additional notes down here:
-------------------------------------------------

Based on input from Oyster Creek Station ID: 8652437
SLR Trend based on input from Oregon Inlet Marina Station ID: 8652587
```

This file includes tidal datum information derived from NOAA NOS/CO-OPS stations located along coastal areas throughout the United States (including Alaska and Hawaii) (Figure 2). Data for the present epoch (1983-2001) is available online at the CO-OPS website:
http://co-ops.nos.noaa.gov/station_retrieve.shtml?type=Datums.
Four input parameters utilized in SLAMM modeling are taken from measurements recorded at these stations:

- historic sea level rise trend
- NAVD88 datum correction
- ocean tide range
- mean high water spring

Historic SLR trend in mm/year is recorded directly at a handful of stations (Figure 3). That data is also available online from the NOAA NOS website:
http://tidesandcurrents.noaa.gov/sltrends/sltrends.html
The NAVD88 datum correction is calculated as the difference between two station measurements – MTL (mean tide level) and NAVD88 (North American Vertical Datum 1988) both measured in meters. Ocean tide range is calculated by subtracting MLLW (mean lower-low water) from MHHW (mean higher-high water). Mean high water spring is calculated by subtracting MTL (mean tide level) from MHW (mean high water).

Station latitude and longitude coordinates were used to a create point file. The station point file was intersected with HUC polygons to generate a file containing recorded measurements and HUC identification information. Because processing was done using 8-digit HUCs as unique sites and each HUC boundary contains multiple CO-OPS stations, it was necessary to normalize station measurements. Mean values for station measurements were calculated by HUC code and station data closest to those means were assigned to each HUC for site parameter inputs.

Figure 2. SAMBI extent, 8-Digit HUC boundaries, and NOS/CO-OPS station locations.

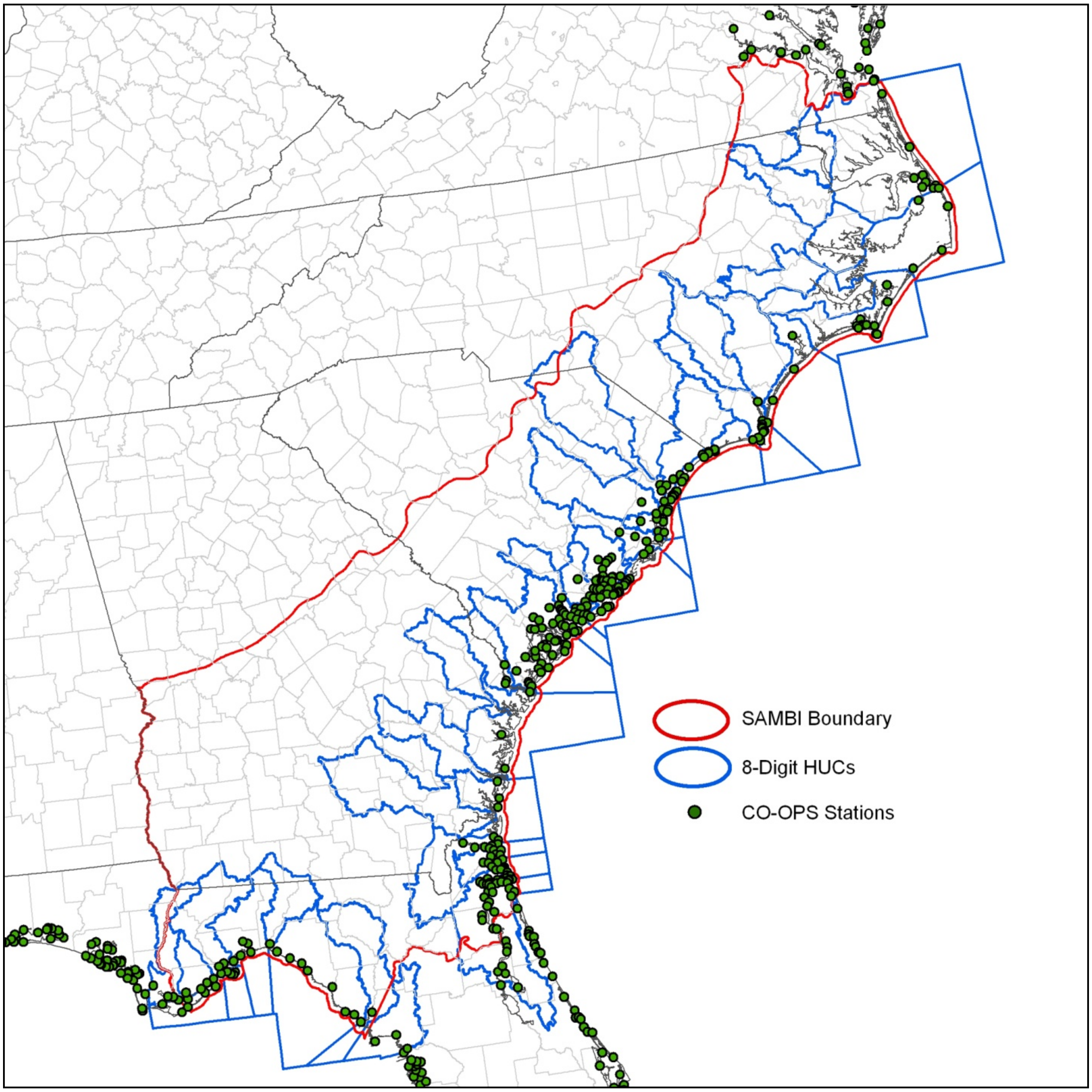
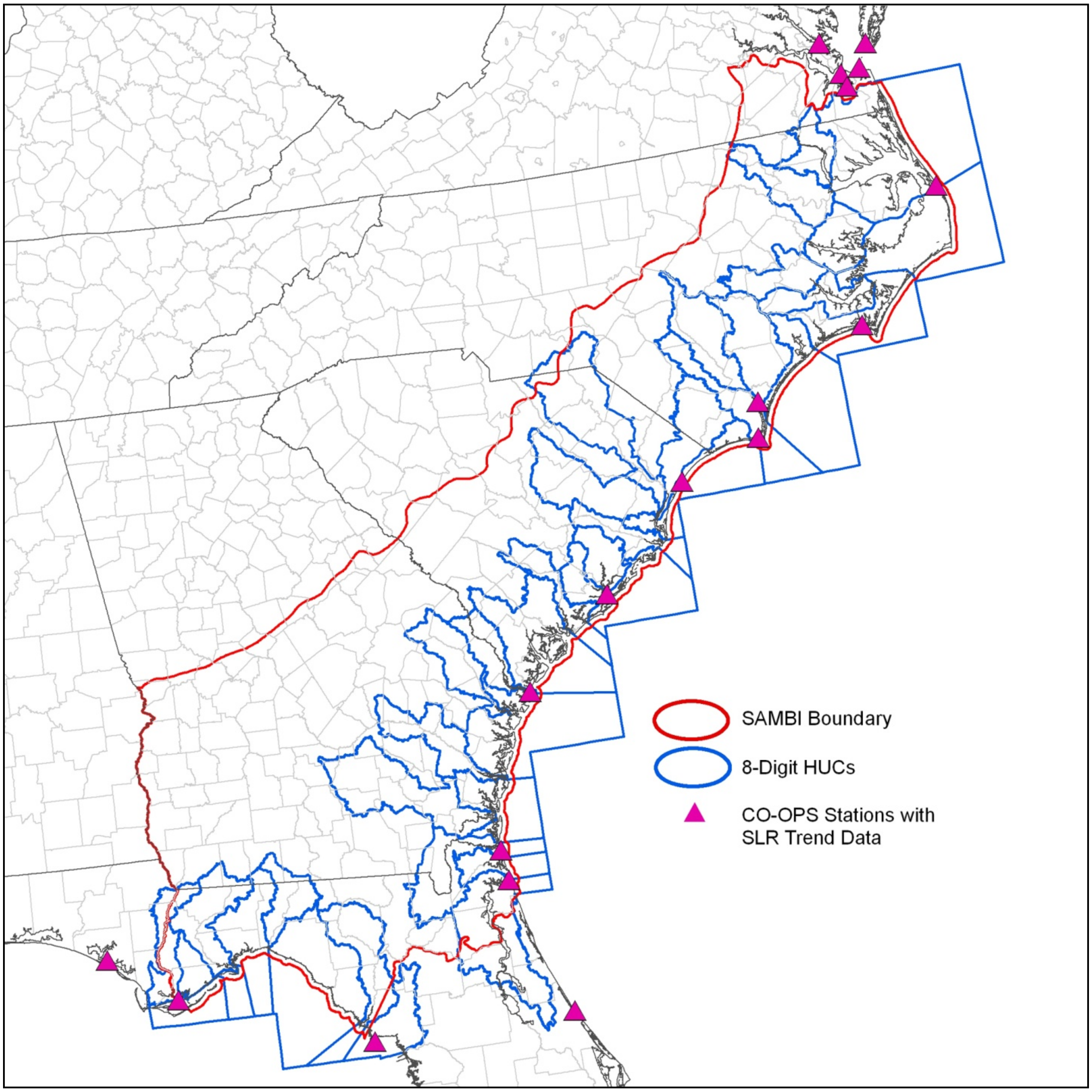


Figure 3. NOS/CO-OPS stations with sea level rise trend data within the SAMBI extent



All other parameter inputs were left as default values with exception of DEM date, NWI photo date, and direction offshore. DEM date was standardized as 2001 per NED data information. NWI photo date was established as 1982 since the majority of data within the SAMBI extent was collected during that year. Offshore direction was determined by visual inspection of each HUC's coastline boundary relative marine waters. Any of the input parameters specified in the site file can be changed by the user with additional site specific information.

SLAMM Data Input & Output Processing

All data layer inputs (NWI, DEM, slope, impervious surface, and dikes) for SLAMM are required to be in ASCII text format. File names for each input must be identical with the exception of the last three characters. These last three charcters are used by SLAMM to differentiate data:

- dem = DEM layer
- dik = dikes/impoundments layer
- imp = impervious surface layer
- nwi = NWI-SLAMM cross-walk layer
- slp = slope layer

For example, the site HUC 03010107 input file names would be as follows:

- Site file = HUC3010107Site.txt
- DEM file = HUC3010107dem.txt
- Dikes file = HUC3010107dik.txt
- Impervious suface file = HUC3010107imp.txt
- NWI-SLAMM cross-walk file = HUC3010107nwi.txt
- Slope file = HUC3010107slp.txt

Data were pre-processed and assembled in a raster environment and then converted to ASCII format using Arc Macro Language (AML) scripting (Appendix 1).

Model runs were automated using a VBScript that interacts with the SLAMM 5.0.1 Windows form interface (Appendix B). Models were run for four climate scenarios (A1B, A2, B1, and A1FI) in 10 year increments for years 2000 (initial conditions) to year 2100 using the “protect developed” option.

Outputs from SLAMM include ASCII raster text files and MS Excel spreadsheets for each climate scenario. To utilize model outputs in GIS, ASCII files were converted to ArcInfo GRID format. AML scripting was implemented to convert outputs to rasters and combine outputs (Appendices C and D). Each of the 39 8-digit HUCs was run separately and then merged to create seamless decadel maps by climate scenario for coastal areas from southern Virginia to northern Florida (Figures 4 and 5).

Figure 4. SLAMM output using climate scenario A1B for year 2000 (initial conditions) within the SAMBI extent.

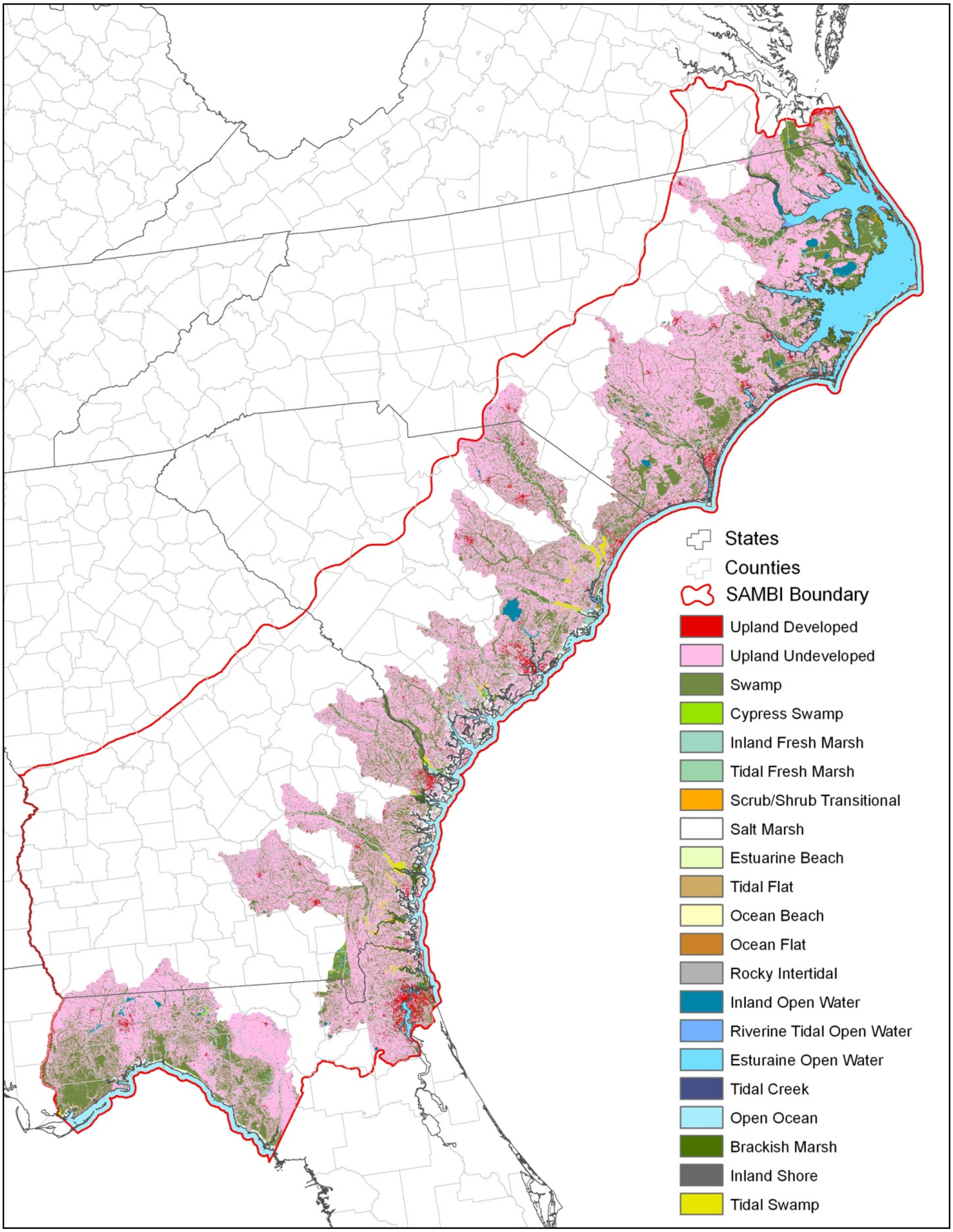
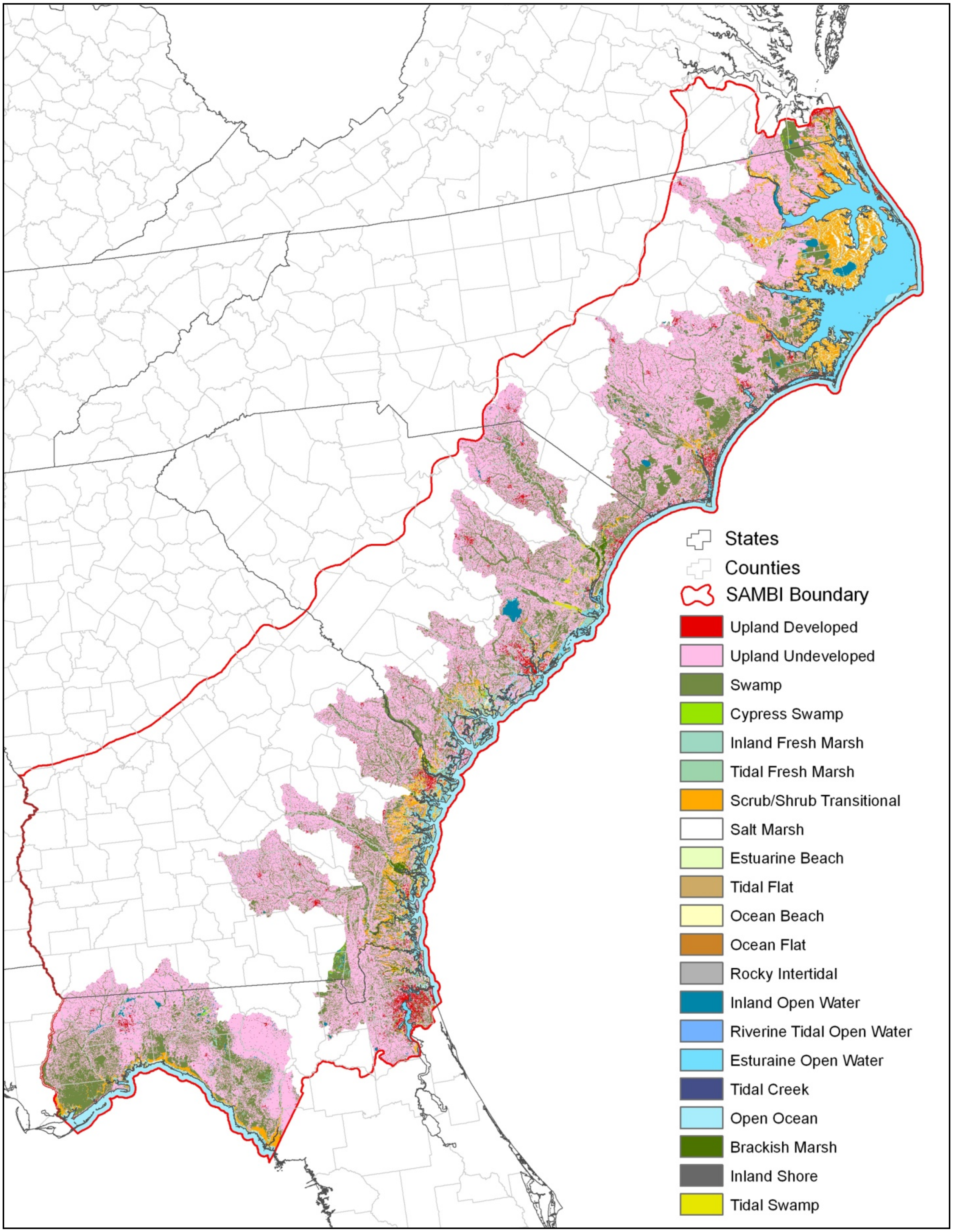


Figure 5. SLAMM output using climate scenario A1B for year 2100 within the SAMBI extent.



LITERATURE CITED

Clough, J. S. 2008. SLAMM 5.0.1. Technical documentation and executable program downloadable from http://www.warrenpinnacle.com/prof/SLAMM/index.html.

Cowardin, L. M. V. Carter, F. C. Golet, and E. T. LaRoe. 1979. Classification of wetlands and deepwater habitats of the United States. U.S. Dep. Interior, Fish and Wildl. Serv. FWS/OBS - 79/31.

Homer, C. C. Huang, L. Yang, B. Wylie and M. Coan. 2004. *Development of a 2001 National Landcover Database for the United States. Photogrammetric Engineering and Remote Sensing*, Vol. 70, No. 7, July 2004, pp. 829-840.

National Aeronautics and Space Administration. 2009. Jet Propulsion Laboratory, Califorinia Institute of Technology. Shuttle Radar Topography Mission. Available online, URL: http://www2.jpl.nasa.gov/srtm/.

North Carolina Floodplain Mapping Program. 2009. North Carolina Floodplain Mapping Information System. Available online, URL: http://www.ncfloodmaps.com/default_swf.asp?r=2.

Park, R. A., T. V. Armentano, and C. L. Cloonan. 1986. Predicting the Effects of Sea Level Ris on Coastal Wetlands. Pages 129-152 in J. G. Titus, ed. *Effects of Changes in Stratospheric Ozone and Global Climate*, *Vol. 4: Sea Level Rise*. U.S. Environmental Protection Agency, Washington, D.C.

Seaber, P.R., Kapinos, F.P., and Knapp, G.L., 1987, Hydrologic Unit Maps: U.S. Geological Survey Water-Supply Paper 2294, 63 p.

U.S. Geological Survey. 2003. National Mapping Division EROS Data Center. National Elevation Dataset. Available online, URL: http://ned.usgs.gov/.

U.S. Geological Survey. 2009. National Mapping Division EROS Data Center. Digital Line Graphs. Available online, URL: http://eros.usgs.gov/products/map/dlg.html.

APPENDICES

ARC MACRO LANGUAGE (AML), AND VB SCRIPT AUTOMATION OF SLAMM MODELING

The following are scripts written by the Biodiversity and Spatial Information Center to automate processing of SLAMM, required input and output data sets. The majority are written to be performed using ESRI's ARC Workstation command line product. Knowledge of programming and scripting in AML and VBScript is essential to replicate any of the automation procedures detailed herein. Cutting and pasting the scripts as written and editing in any standard text editor should be done only if the user is familiar with AML, and VBScript program flow and syntax, various GIS data formats, GIS terminology and basic concepts, and standard data storage and retrieval using Windows operating systems.

APPENDIX A. SLAMMDataPrepLiDARfloat.aml

This AML script is used to create input datasets for SLAMM modeling including NWI-SLAMM crosswalk, DEMs, slope, and impervious surface in ASCII text format. It uses an 8-digit HUC polygon coverage to iterate processing.

```
/**********************************************************************
/*
/*                              SLAMMDataPrepLiDARfloat.aml
/*
/*                    Use this AML to create ASCII text files for input to SLAMM
/*                    modeling procedures using LiDAR DEM derived data in NC.
/*                    It creates inputs for the SAMBI area on a HUC-by-HUC
/*                    basis using a specified HUC coverage.
/*
/*          The following datasets are necessary to run SLAMM models:
/*                    NWI
/*                    DEM
/*                    Slope
/*                    Impervious Surface
/*          Additionally, the following could be used to augment datasets:
/*                    Dikes- to prevent water level rises from inundating certain locations
/*
/* INPUTS:
/*                    1) The following grids located in the working directory:
/*                              NWI (Adjusted with land cover data to identify urban)
/*                              DEM  made using LiDAR
/*                              {Note: Slope grids will be made with this AML}
/*                              Imp (impervious surface)
/*                              Dikes (Dike layer based on NWI attributes)
/*                    2) A HUC coverage for the NC portion of the SAMBI area
/*                    3) Directories for each of the HUCs named as follows:
/*                              HUC<8-digit HUC minus the leading 0> ie. HUC3130013
/* OUTPUTS:
/*                    ASCII text files in HUC numbered directories.
/*
/*                    Also, a "site" file needs to be created that sets various model
/*                    parameters and points to data input file name and locations.
/*                    These have been created and placed in each of the separate
/*                    HUC numbered directories. See SLAMM documentation
/*                    for more info.
/*
/*  MJR 10 April 2009
/**********************************************************************
&terminal 9999
/* Start Timer
&sv initialize = [show &pt time]

/* -------------------------------Set Variables ------------------------------------------
/* Setting variables
&sv DirWS = [show &workspace]
&sv snap = %DirWS%\snapgrid
&sv dem = %DirWS%\output_float
&sv imp = %DirWS%\imp
&sv nwi = %DirWS%\nwi_slamm_sam
&sv dikes = %DirWS%\dikes
&sv l = HUClist.txt
&sv pre_huc = 0
&sv i = 0

/*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> User Inputs <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
&ty
&sv selcov = [getcover * -poly 'Select the input HUC Coverage:']
&sv huccov = [subst %selcov% %DirWS%\ '']
&ty
/*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
/* Go into TABLES and get a list of HUCs from the coverage
/* and output it to a text file for reading and looping

&sv ok := [delete HUClist.txt -file]
&if [exists huc_numbers.fil -info] &then killinfo huc_numbers.fil

&if ^ [show program] = 'TABLES' &then TABLES

/* First, make sure there are no decimals
/* in the 'HUC' field by replacing it with
/* data calculated without decimals

additem %huccov%.pat hucn 10 10 i
select %huccov%.pat
calculate hucn = huc
dropitem %huccov%.pat huc
additem %huccov%.pat huc 10 10 i
select %huccov%.pat
calculate huc = hucn
dropitem %huccov%.pat hucn

copy %huccov%.pat huc_numbers.fil
sel huc_numbers.fil
sort huc
unload HUClist.txt huc delimited init

&if [show program] = 'TABLES' &then QUIT

/*
/*-----------------------START LOOP---------------------------------
/*
/***************************************************
&sv fileunit [open %l% openstatus -read]
&if %openstatus% ne 0 &th ~
&return &warning Error opening List file %l%
&label next_cover
/*
/*** Get current HUC number from HUClist.txt
/*
&sv rhuc [read %fileunit% readstatus]
&if %readstatus% ne 0 &then &goto no_more_covers

&if %rhuc% < 10000000 and %rhuc% > 0 &then
&sv huc := 0%rhuc%
&else
&sv huc := %rhuc%


&if %huc% ne 0 &then
&do
&ty %huc% %pre_huc%
&if %huc% eq %pre_huc% &then &goto next_cover
&sv i := %i% + 1

/* Create variables for output directories
&sv HName = HUC[substr %huc% 2 7]
&sv OutDir = %DirWS%\input\%HName%

/* Go into arcedit to get individual HUC polygons
AE
edit %huccov%
editfeature polygon
mapextent default
select huc = %huc%
put h%huc%
edit h%huc%
build
QUIT yes

/* Delete ASCII files in the output directory if they exist
```

```
&sv ok := [delete %OutDir%\%HName%dem.txt -file]
&sv ok := [delete %OutDir%\%HName%slp.txt -file]
&sv ok := [delete %OutDir%\%HName%nwi.txt -file]
&sv ok := [delete %OutDir%\%HName%imp.txt -file]
&sv ok := [delete %OutDir%\%HName%dik.txt -file]

&if ^ [show program] = 'GRID' &then GRID
/* Create HUC mask from selected HUC polygon
mg%huc% = polygrid(h%huc%, #, #, #, 30)
&if [exists h%huc% -grid] &then  kill h%huc% all

/* Set window and mask to HUC grid
setwindow mg%huc% %snap%
setmask mg%huc%

/* Clip each dataset to the selected HUC boundary
%OutDir%\%HName%dem = %dem%
tmpslope = slope(%OutDir%\%HName%dem)
tmpnwi = %nwi%
tmpimp = %imp%
tmpdikes = %dikes%

&if [show program] = 'GRID' &then QUIT

/* Create ASCII raster datasets in the output directory

gridascii %OutDir%\%HName%dem %OutDir%\%HName%dem.txt
gridascii tmpslope %OutDir%\%HName%slp.txt
gridascii tmpnwi %OutDir%\%HName%nwi.txt
gridascii tmpimp %OutDir%\%HName%imp.txt
gridascii tmpdikes %OutDir%\%HName%dik.txt

/**************** Clean Up *****************
&severity &error &routine PauseCleanUp
&messages &on

/* Delete temporary grids
&do grd &list tmpdem tmpslope tmpnwi ~
tmpimp tmpdikes mg%huc%
&if [exists %grd% -grid] &then kill %grd% all
&end
/* Delete temporary coverages
&do cov &list h%huc%
&if [exists %cov% -cover] &then kill %cov% all
&end

&sv pre_huc := %huc%

&end

&goto next_cover
&label no_more_covers
&if [close %fileunit%] ne 0 &th [close -all]
/************************************************************

&ty +++++++++++++++++++++++++++++++++++++++++
&ty    Successfully Created SLAMM Inputs
&ty     using HUCs in Polygon Coverage
&ty            "%huccov%"
&ty +++++++++++++++++++++++++++++++++++++++++


/* End Timer
&sv .timesec = [show &pt time]
&r I:/Docs/Scripts/AML/Time/TimeStamp.aml

&return

/*******************************************
&routine PauseCleanUp
```

```
/********************************************
&ty !Error encountered trying to delete temp outputs!
&pause Pausing 5 seconds... &seconds 5
/* Delete temporary grids
&do grd &list tmpdem tmpslope tmpnwi ~
tmpimp mg%huc%
&if [exists %grd% -grid] &then kill %grd% all
&end
/* Delete temporary coverages
&do cov &list h%huc%
&if [exists %cov% -cover] &then kill %cov% all
&end

/* Delete windows tmp directories with the system
&if [exists tmpdem -directory] &then
&do
&system rmdir /s/q tmpdem
&end

&if [exists tmpslope -directory] &then
&do
&system rmdir /s/q tmpslope
&end

&if [exists tmpnwi -directory] &then
&do
&system rmdir /s/q tmpnwi
&end

&if [exists tmpimp -directory] &then
&do
&system rmdir /s/q tmpimp
&end

&if [exists tmpdikes -directory] &then
&do
&system rmdir /s/q tmpdikes
&end

&if [exists mg%huc% -directory] &then
&do
&system rmdir /s/q mg%huc%
&end

&if [exists h%huc% -directory] &then
&do
&system rmdir /s/q h%huc%
&end

&return

/********************************************
```

APPENDIX B. SLAMMRunsA1B.vbs

This VB script is used to run the SLAMM interface for multiple time step iterations for multiple model regions – 8-digit NHD HUCs in this instance. Separate scripts are required to run the other 3 climate scenarios – A1FI, A2, and B1.

```
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'
'		SLAMM Model Execution VBScript
'
'				Use this script to run SLAMM programmatically. It will open the
'				application and move through the interface by using the
'				SendKeys method in WSH.
'
'				The input and output directory structure must be as follows:
'						..\input\HUC<8-digit HUC with no leading 0>
'						..\output\HUC<8-digit HUC with no leading 0>
'
'				The SLAMM site parameter files must use the naming convention:
'						HUC<8-digit HUC with no leading 0>Site.txt and be placed in
'						the input directory with all other data inputs
'
'				This script must be placed in and run from the ..\input directory
'
'				It makes a list of all the HUC<8-digit HUC with no leading 0> directories
'						and loops on that list
'
'				It runs twice for each HUC - once with a time step of 5 years until 2010
'						to get 2010 output, and once with 10 year steps until the year 2100
'
'				Edits: Changed the script to check for existing output files for years 2010
'							or 2100. If these outputs exist, it will skip that HUC iteration and
'								move to the next HUC.
'
'				Set the input root directory (e.g. I:\Proj2\MSCGP\SLAMM\) on line 200
'
'		NOTE: It is highly recommended that you CLOSE ALL WINDOWS when running
'		this script because it uses the SendKeys method to direct control of the
'		SLAMM application interface. If something goes awry, these key stroke
'		directives may interact with other open applications and screw them up.
'		Simply open a DOS shell, navigate to the directory where the script is
'		located, and type the name of the file.vbs and hit enter.
'
'		For a list of SendKeys representations, see the MS website:
'		http://www.microsoft.com/technet/scriptcenter/guide/sas_wsh_hilv.mspx?mfr=true
'
'  MJR 24-27 April 2009
'  Edited MJR 5 May 2009
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


'#####################################################################
'
' Function to run SLAMM programatically
'
'  Parameters passed to the function include the input and output directories
'  where this is being run, the time step (i.e. 5 years/10 years etc...) and the
'  end year for simulations to stop
'

Function RunSLAMM(InDir, OutDir, timestep, endyear)

		Dim Program, oShell, Return
		Dim win1, Success

		Program = """C:\Program Files\SLAMM\SLAMM5.exe"""

```

```

	Set oShell = CreateObject("WScript.Shell")


	' Start the SLAMM program and make sure it's active
	' by using AppActivate and referencing the SLAMM app
	' using the text in the title of the window

	WScript.Sleep 1500
	Return = oShell.Run(Program, 1, False)
	WScript.Sleep 300
	win1 = oShell.AppActivate("SLAMM v5.0.1   July 2008")


	' Now start moving through the interface and populate
	' the various fields in the form using the SendKeys method

		'Move to the Input Data Files text box
		WScript.Sleep 300
		oShell.SendKeys "{TAB}"
		WScript.Sleep 300
		oShell.SendKeys InDir & "Site.txt"

		'Move to the Output Directory text box
		WScript.Sleep 300
		oShell.SendKeys "{TAB}{TAB}"
		oShell.SendKeys OutDir

		'Enter a simulations end year
		WScript.Sleep 300
		oShell.SendKeys "{TAB}"
		WScript.Sleep 300
		oShell.SendKeys endyear
		WScript.Sleep 300

		' Move to the GIS Output Options radio button
		oShell.SendKeys "{TAB 21}"
		WScript.Sleep 300
		oShell.SendKeys "{DOWN}"

		' Move to the Display Maps-No Maps radio button
		WScript.Sleep 300
		oShell.SendKeys "{TAB}{DOWN}"

		' Move to the Time Step text box
		oShell.SendKeys "{TAB 5}"
		WScript.Sleep 300
		oShell.SendKeys timestep

		' Move to the Additional File Opt. check box
		' and check it using " " as the key sent
		oShell.SendKeys "{TAB 5}"
		WScript.Sleep 300
		oShell.SendKeys " "


		' Finally, move to the Execute button and run SLAMM
		WScript.Sleep 300
		oShell.SendKeys "{TAB 3}"
		oShell.SendKeys "{ENTER}"
		WScript.Sleep 3000

		' Test to see if SLAMM has finished and then click
		' the OK button on the Simulations Complete dialog
		Do Until Success = True
			Success = oShell.AppActivate("Information")
			WScript.Sleep 10000
		Loop
		oShell.SendKeys "{ENTER}"

```

```
                    'Now re-activate the SLAMM window and then close it
                    WScript.Sleep 1000
                    oShell.AppActivate "SLAMM v5.0.1   July 2008"
                    WScript.Sleep 300
                    oShell.SendKeys "%{F4}"
                    'WScript.Echo OutDir
                    WScript.Sleep 10000

End Function
'####################################################
'++++++++++++++++++++++++++++++++++++++++++
' Function to read in a list of files in a given directory

Function ShowFileList(rtdir)

  Dim fso, f, f1, fc, s, txtname, indir, outdir
  Set fso = CreateObject("Scripting.FileSystemObject")
  Set f = fso.GetFolder(rtdir)
  Set fc = f.Files
  For Each f1 in fc
    s = s & f1.name '& VbCrLf
    txtname = s
  Next
  ShowFileList = txtname

End Function
'++++++++++++++++++++++++++++++++++++++++++

'++++++++++++++++++++++++++++++++++++++++++
' Function created for a regular expression to test
' a file name pattern

Function RegExpTest(patrn, strng)

  ' Create variables
  Dim regEx, Match, Matches

  ' Create a regular expression
  Set regEx = New RegExp

  regEx.Pattern = patrn   ' Set pattern
  regEx.IgnoreCase = True   ' Set case insensitivity
  regEx.Global = True   ' Set global applicability

  Set Matches = regEx.Execute(strng)   ' Execute search
   For Each Match in Matches   ' Iterate Matches collection
     RetStr = RetStr & Match.Value' & vbCRLF
   Next
  RegExpTest = RetStr

End Function
'++++++++++++++++++++++++++++++++++++++++++
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

' Define the variables to pass as parameters then run
' the RunSLAMM function using passed parameters/variables

Dim rootdir, cFileNames, strFileName
Dim i, id, od, Step, EndYear
Dim oFS, fld, x, sf, a, InName
Dim strInputDir, strOutputDir, EndDate, EndTime

rootdir = "S:\SLAMM\"
id = rootdir & "input"
od = rootdir & "output"

' Run SLAMM twice for each HUC
' One with a time step of 5 years to get output for 2010
' and one with a time step of 10 years until the year 2100
```

```
i = 0
For i = 1 to 2

        If i = 1 Then
          Step = "5"
          EndYear = "2010"
        Else
          Step = "10"
          EndYear = "2100"
        End If

                ' Get a list of HUC folders in the SLAMM\input directory
                ' and loop based on that folder list

                Set oFS = CreateObject("Scripting.FileSystemObject")
                Set fld = oFS.GetFolder(id)
                Set sf = fld.SubFolders
                For Each x in sf

                        ' Set variables
                        a = x.name
                        InName = a
                        strInputDir = fld & "\" & InName & "\" & InName
                        strOutputDir = rootdir & "output\" & InName & "\"

                        ' Get a list of the files in each directory using the
                        ' ShowFileList function
                        cFileNames = ShowFileList(strOutputDir)

                        ' Run SLAMM only if output doesn't already exist.
                        ' Use the RegExpTest function to find the file name
                        ' matching the specified pattern for year 2010 or
                        ' year 2100 output

                        If i = 1 Then
                          strFileName = RegExpTest("HUC\w*_2010\w*\.\w*\.txt", cFileNames)
                          If strFileName <> "" Then
                            ' Output file for year 2010 exists. Move on to the next HUC.
                          Else
                            'Run SLAMM using the RunSLAMM function created above
                                WScript.Sleep 1000
                                RunSLAMM strInputDir, strOutputDir, Step, EndYear
                          End If
                        Else
                          strFileName = RegExpTest("HUC\w*_2100\w*\.\w*\.txt", cFileNames)
                          If strFileName <> "" Then
                            ' Output file for year 2100 exists. Move on to the next HUC.
                          Else
                            'Run SLAMM using the RunSLAMM function created above
                                WScript.Sleep 1000
                            RunSLAMM strInputDir, strOutputDir, Step, EndYear
                          End If
                        End If

                Next

Next
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

EndDate = Date
EndTime = Time
WScript.Echo "<<< SLAMM Simulations Complete for all HUCs. >>>" & _
        VbCrLf & VbCrLf & _
         "      Script Ended on " & EndDate & " at " & EndTime
WScript.Quit
```

APPENDIX C. ConvertSLAMMOutput.aml

This AML script is used to convert SLAMM modeling output from ASCII to raster format. It uses an 8-digit HUC polygon coverage to iterate processing. Use the AML MosaicSLAMMLoop.aml to combine outputs in 10 year increments for years 2000-2100.

```
/*************************************************************************************
/*          ConvertSLAMMOutput.aml
/*
/*                    Use this to convert directories containing ASCII output from
/*                    SLAMM model runs into grids and define an Albers projection.
/*                    It creates a list of HUC output directories in the SLAMM\Output
/*                    directory and writes a file called WSList. It loops on this list,
/*                    going into each of the HUC directories to create another list called
/*                    ASCIIList. It then loops on ASCIIList to create grids from each
/*                    of the ASCIIs in that particular HUC directory.
/*
/*
/*  NOTE:
/*   Need to merge these HUC grids together after each individual
/*   grid has been created to create SAMBI-wide SLAMM output.
/*
/* MJR 14 April 2009
/*************************************************************************************
&terminal 9999

/* Start Timer...
&sv initialize := [show &pt time]

/*-------------------------------------------------------------------
/* Setting variables
&sv DirWS = [show &workspace]

&sv replace = [locase [response ~
'Replace SLAMM output grids if they exist? (yes | no)' yes]]

/* Turn off warning that directory is not a workspace
&severity &warning &ignore
&messages &off &all

/*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
/*    Create a list of output HUC directories and loop on that list
/* +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
&if [exists WSList -file] = .true. &then ~
&sv status [delete WSList -file]
&sv ndirs := [filelist HUC* WSList -directory]
&sv file = [open WSList ok -read]
&do i = 1 &to %ndirs%

&sv HUCDir = [read %file% ok]
&sv HUC = 0[substr %HUCDir% 4 7]
&if %ndirs% <= 0 &then
&do
&ty
&ty No Output HUC Directories Exist
&ty
&goto skip
&end
&else
&do
w %HUCDir%
&ty ###############################################
&ty Working on HUC %HUC%...
&sv count := [filelist HUC*.txt ASCIIList -file]
&sv unit := [open ASCIIList ok -read]
/* ---------------------------------------------------------------------------
```

```
/* Now loop on the list of ASCIIs in this HUC directory
/* ---------------------------------------------------------------------------
&if %count% <= 0 &then
&do
&ty No SLAMM Output in Directory %HUCDir%; &ty
&end
&else
&do j := 1 &to %count%
&sv ASCII := [read %unit% ok]
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/* Set variables for file names
&sv fn1 = [substr %ASCII% 4 3]          /* 1st part of file name variable
&sv fn2 = [substr %ASCII% 8 3]          /* 2nd part of file name variable
&sv year = [substr %ASCII% 12 4]        /* Year variable
&sv gn = h%fn1%%fn2%_%year%             /* Unique grid name variable

&if [exists %gn% -grid] = .true. &then
&do
&if %replace% = yes OR %replace% = y  &then
&do
&ty SLAMM output grid %gn% exists.
&ty Replacing existing output...
kill %gn% all
&call ConvertASCII
&end
&else
&do
&ty -------------------------------------------
&ty SLAMM output %gn% exists.
&ty Not replacing existing output.
&end
&end
&else
&do
&call ConvertASCII
&end
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
&end
/* close and delete ASCIIList file.
&sv ok := [close %unit%]
&sv ok := [delete ASCIIList -file]
w %DirWS%
&end
&label skip

&end
/* close and delete WSList file.
&ty Exiting WSList ...; &ty
&sv ok := [close %file%]
&sv ok := [delete WSList -file]

/* End the HUC directory list loop
/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

&messages &on

/*------------------------------------------------------------------
/* End Timer....
&sv .timesec = [show &pt time]
&r I:\Proj\SEGap\Vert\Model\misc\TimeStamp.aml
/*------------------------------------------------------------------

&return

/****************************************************
&routine ConvertASCII
/****************************************************
&ty -------------------------------------------
&ty Working on ASCII Output for Year %year% ...
/* ----------------------------------------------------------
```

```
/* Convert ASCII to a Raster
/* -----------------------------------------------------------
&ty Converting ASCII to a Raster Grid ...
asciigrid %ASCII% %gn% int
/* -----------------------------------------------------------
/* Define Grid Projection
/* -----------------------------------------------------------
&ty Defining Grid Projection ...
projectdefine grid %gn%
projection albers
units meters
datum nad83
spheroid grs1980
parameters
29 30  0.000 /* 1st standard parallel
45 30  0.000 /* 2nd standard parallel
-96 0  0.000 /* central meridian
23 0  0.000 /* latitude of projection's origin
/* false easting (meters)
/* false northing (meters)
&return
```

APPENDIX D. MosaicSLAMMLoop.aml

This AML script is used to combine SLAMM modeled output raster datasets in 10 year increments from 2000 to 2100. It uses an 8-digit HUC polygon coverage to iterate processing.

```
1   /*************************************************************************************
2   /*************************************************************************************
3   /*
4   /* MosaicSLAMMLoop.aml
5   /*
6   /*  Use this to mosaic SLAMM grid outputs by year. It also uses
7   /*  a focal mean function to eliminate any NoData values between adjacent
8   /*  grids.
9   /*
10  /*  NOTE: This AML runs an iterative process to mosaic grids
11  /*        create on a HUC-by-HUC basis by year. Therefore, it takes
12  /*        an increasingly lengthy amount of time to process.
13  /*
14  /*  Inputs: 1) A list of HUC directories for looping
15  /*                          It is set interactively by the user on Line 33
16  /*
17  /*              2) SLAMM output grids by year in their own HUC directory
18  /*
19  /*  *******************************************************************************
20  /*
21  /*  Original scripting MJR 11 May 2009
22  /*  Edited to include varying climate scenario runs MJR 26 August 2009
23  /*
24  /*************************************************************************************
25  /*************************************************************************************
26
27  &terminal 9999
28  /* Start Timer
29  &sv initialize = [show &pt time]
30
31  /*-------------------------------------------------------------------
32  /* Select the text file list
33  &sv HUCList = [getfile 'HUCList*.txt' -file 'What HUC list do you want to run? ~
34  (HUCList#.txt)']
35  &if [exists %HUCList% -file] = .false. &then
36   &do
37    &ty; &ty \ Error - Text file does not exist.
38         &ty \ Stopping execution.
39         &return
40    &end
41
42  /* Get the climate scenario from user input
43  &sv scenario = [upcase [response 'Enter the Climate Scenario (A1B|A1FI|A2|B1)']]
44  &if %scenario% = A1B &then &sv s = 1
45    &else; &if %scenario% = A1FI &then &sv s = 2
46    &else; &if %scenario% = A2 &then &sv s = 3
47    &else; &if %scenario% = B1 &then &sv s = 4
48    &else
49      &do
50        &ty; &ty !! Error !!
51        &ty You Must Enter a Climate Scenario Abbreviation
52        &return
53      &end
54
55  /*-------------------------------------------------------------------
56  /* Setting variables
57  &sv DirWS = [show &workspace]
58  &sv SLAMMDir = I:\Proj2\MSCGP\SLAMM
59  &sv snap = %SLAMMDir%\snapgrid
60  &sv HUCsMask = %SLAMMDir%\HUCsMask
61  &sv SAMBIBnd = %SLAMMDir%\sambibnd
62  &sv pre_HUC = NA
```

```
&sv i = 0
&sv pre_i = 0
&sv skip = 0

/* ------------------------------------------------------------------------------------------
/*   Loop Year-by-Year using a list of years for which grids
/*   were made during SLAMM output runs
/*************************************************

&do yr &list init 2010 2020 2030 2040 2050 ~
            2060 2070 2080 2090 2100
/*************************************************
&ty ############################################
&ty Working On SLAMM Output for Year %yr%...; &ty

/* ---------------------------------------------------------------------------------
/* Loop on HUC List to mosaic grids in HUC directories
/* ---------------------------------------------------------------------------------
&sv fileunit = [open %HUCList% openstatus -read]
&if %openstatus% ne 0 &then ~
 &return &warning Error opening List file %HUCList%
&label next_HUC

/*** Get current HUC from HUCList.txt
&sv HUC = [read %fileunit% readstatus]
&if %readstatus% ne 0 &th &goto no_more_HUCs

            /*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            /* Set variables for file names
            &sv HUCDir = %DirWS%\%HUC%
            &sv fn1 = [substr %HUC% 4 3]  /* 1st part of file name variable
            &sv fn2 = [substr %HUC% 8 3]  /* 2nd part of file name variable
            &sv year = [substr %HUC% 12 4]            /* Year variable
            &sv gn = h%fn1%%fn2%_%yr%%s%         /* Unique grid name variable

             &if %HUC% ne '' &then
             &do
                        &ty %HUC% %pre_HUC%
                        &if %HUC% eq %pre_HUC% &then &goto next_HUC
                        &sv i := %i% + 1

                        &if [exists %HUCDir%\%gn% -grid] = .false. &then
                         &do
                                    &ty; &ty Grid HUC %gn% doesn't exist. Moving on ...; &ty
                                    &sv i = %pre_i%
                                    &goto next
                         &end
                        &if %i% eq 1 &then
                         &do
                                     copy %HUCDir%\%gn% s%i%_%yr%%s%_mg
                                    &sv pre_i := %i%
                         &end
                        &else
                         &do
                                    &if ^ [show program] = 'GRID' &then GRID
                                     /* Iteratively mosaic HUCs
                                     &ty
                                     &ty Mosaicking grids HUC-by-HUC....
                                     &ty

                                     s%i%_%yr%%s%_mg = mosaic(s%pre_i%_%yr%%s%_mg,  %HUCDir%\%gn%)

                                                /**************** Clean Up Grids *****************
                                                &severity &error &routine PauseCleanUp
                                                &messages &on

                                                /* Delete temporary grids
                                                &do grd &list fm s%pre_i%_%yr%%s%_mg
                                                            &if [exists %grd% -grid] &then kill %grd% all
                                                &end
```

```

                                &if [show program] = 'GRID' &then QUIT
                                &sv pre_i := %i%
                        &end

            &end
            &label next
            &sv skip = 0
            &sv pre_HUC := %HUC%

&goto next_HUC
&label no_more_HUCs
&if [close %fileunit%] ne 0 &th [close -all]

/* Clip Mosaicked Year Output to the SAMBI Boundary
&if ^ [show program] = 'GRID' &then GRID
  &ty Clipping Final Output for %yr% to SAMBI...
  setwindow %SAMBIBnd% %snap%
  setmask %SAMBIBnd%
  %scenario%_%yr% = s%i%_%yr%%s%_mg

  &if [exists s%i%_%yr%%s%_mg -grid] &then ~
                                kill s%i%_%yr%%s%_mg all
&if [show program] = 'GRID' &then QUIT

&if [exists %scenario%_%yr% -grid] = .true. &then
  &do
        &ty ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    &ty    Successfully Mosaicked SLAMM HUCs for year %yr%
        &ty ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  &end
&else
  &do
        &ty !-------------------------------------------!
        &ty   Problem creating output for year %yr%
        &ty !-------------------------------------------!
  &end

&sv i = 0
&sv pre_i = 0

&end

/*************************************************************
/* Finish by filling in NoData HUC edge cells using
/* NIBBLE function
/*
&do yr &list 1982 2010 2020 2030 2040 2050 ~
        2060 2070 2080 2090 2100
        &call FinishWithNibble
        &ty Finished nibble function for year %yr%; &ty
&end


&ty ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
&ty    Mosaicked SLAMM HUCs for Years 1982 - 2100
&ty ++++++++++++++++++++++++++++++++++++++++++++++++++++++++

/* End Timer
&sv .timesec = [show &pt time]
&r I:/Docs/Scripts/AML/Time/TimeStamp.aml

&return


/*********************************************
&routine PauseCleanUp
/*********************************************
&ty !Error encountered trying to delete temp outputs!
&pause Pausing 5 seconds... &seconds 5
```

```
/* Delete temporary grids
&do grd &list fm s%pre_i%_%yr%%s%_mg
    &if [exists %grd% -grid] &then kill %grd% all
&end

/* Delete windows tmp directories with the system
&if [exists fm -directory] &then
  &do
    &system rmdir /s/q fm
  &end

&if [exists s%pre_i%_%yr%%s%_mg -directory] &then
  &do
    &system rmdir /s/q s%pre_i%_%yr%%s%_mg
  &end

&return
/*******************************************
/*******************************************
&routine FinishWithNibble
/*******************************************
&if ^ [show program] = 'GRID' &then GRID
  setwindow %HUCsMask% %snap%
  setmask %HUCsMask%

  tmp = isnull(%scenario%_%yr%)
  nibmask = con(tmp == 1, setnull(tmp), tmp)
  ingrid = con(isnull(%scenario%_%yr%), 0, %scenario%_%yr%)
  nibble1 = nibble(ingrid, nibmask)
  &if [exists %scenario%_%yr% -grid] &then kill %scenario%_%yr% all
  rename nibble1 %scenario%_%yr%

  /* Delete temporary grids
  &do grd &list tmp nibmask ingrid nibble1
      &if [exists %grd% -grid] &then kill %grd% all
  &end
&if [show program] = 'GRID' &then QUIT

&return
```