EPA
United States
Environmental Protection
Agency

EPA/601/R-14/007 September 2014 www.epa.gov/ord

Sea Level Affecting Marshes Model

New Functionality for Predicting Changes in Distribution of Submerged Aquatic Vegetation in Response to Sea Level Rise

Version 1.0

Office of
Research and Development
National Health and
Environmental Effects
Research Laboratory
Western Ecology Division

EPA/600/R-14/007
September, 2014

Sea Level Affecting Marshes Model (SLAMM) - New Functionality for Predicting Changes in Distribution of Submerged Aquatic Vegetation in Response to Sea Level Rise. Version 1.0

Henry Lee II[1], Deborah A. Reusser[2], Melanie R. Frazier[3], Lee M. McCoy[4], Patrick J. Clinton[1], and Jonathan S. Clough[5]

1: Pacific Coastal Ecology Branch, Western Ecology Division, United States Environmental Protection Agency, Newport, Oregon, 97365

2: Western Fisheries Research Center, U.S. Geological Survey, Newport, Oregon, 97365

3: National Center for Ecological Analysis and Synthesis, 735 State St. Suite 300, Santa Barbara, CA, 93101 USA

4: Agricultural Research Service, United States Department of Agriculture, Newport, OR 97365

5: Warren Pinnacle Consulting, Inc., PO BOX 351, Waitsfield VT, 05673

National Health and Environmental Effects Research Laboratory
Office of Research and Development
U.S. Environmental Protection Agency
Newport, OR 97365

Disclaimer:

The information in this document was partially funded by the U.S. Environmental Protection Agency. This report was subjected to review by the National Health and Environmental Effects Research Laboratory's Western Ecology Division of the EPA, and the Western Fisheries Research Center of U.S. Geological Survey, and is approved for publication. However, approval does not signify that the contents reflect the views of the U.S. EPA. The use of trade, firm, or corporation names in this publication is for the information and convenience of the reader; such use does not constitute official endorsement or approval by the U.S. Department of Interior, the U.S. Geological Survey, or the U.S. EPA of any product or service to the exclusion of others that may be suitable.

Acknowledgements:

We would like to thank Dr. Cheryl Brown (U.S. EPA) for her assistance with developing this report. Special thanks to Rebecca Flitcroft (USDA Forest Service, PNW Research Station) and John Bauer (The Wetland Conservancy) for reviewing this document. Debbie Reusser was partially funded through IAG#DW-14-92252201-0 with the U.S. EPA.

Recommend Citation:

Lee II, H., Reusser, D.A., Frazier, M.R., McCoy, L.M., Clinton, P.J., and Clough, J.S. 2014. Sea Level Affecting Marshes Model (SLAMM) - New Functionality for Predicting Changes in Distribution of Submerged Aquatic Vegetation in Response to Sea Level Rise. Version 1.0. U.S. EPA, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Western Ecology Division. EPA/600/R-14/007.

Table of Contents

Table of Figures

Table of Tables

Introduction

Submerged aquatic vegetation (SAV) is an ecologically important habitat world-wide. In Pacific Northwest (PNW) estuaries, SAV in the lower intertidal and shallow subtidal habitats are dominated by the native seagrass, *Zostera marina* Linnaeus, 1753. Within this report, SAV and seagrass refer to *Z. marina* seagrass beds in PNW estuaries. *Z. marina* provides important habitat for juvenile salmon, dungeness crabs, migratory shore birds, and benthic assemblages (e.g., Philips, 1984; Williamson, 2006; Ferraro and Cole, 2007; Shaughnessy et al., 2012). *Z. marina* typically occurs in a narrow depth range. For example, in Oregon estuaries *Zostera marina* primarily occurs within the depth range of -1 to +1 m relative to Mean Lower Low Water (MLLW) (Young et al. 2012). Because of their narrow depth range, the distribution of these seagrass beds are potentially vulnerable to sea level rise (SLR) through increased water depths and associated reductions in underwater light levels, alterations in tidal variations, altered water movement and wave action, and increased seawater intrusion (Short and Neckles, 1999).

The "Sea-Level Affecting Marshes Model" (SLAMM) is a moderate resolution model used to predict the effects of sea level rise on marsh habitats (Craft et al. 2009). SLAMM has been used extensively on both the west coast (e.g., Glick et al., 2007) and east coast (e.g., Geselbracht et al., 2011) of the United States to evaluate potential changes in the distribution and extent of tidal marsh habitats. However, a limitation of the current version of SLAMM, (Version 6.2) is that it lacks the ability to model distribution changes in seagrass habitat resulting from sea level rise. Because of the ecological importance of SAV habitats, U.S. EPA, USGS, and USDA partnered with Warren Pinnacle Consulting to enhance the SLAMM modeling software to include new functionality in order to predict changes in *Zostera marina* distribution within Pacific Northwest estuaries in response to sea level rise. Specifically, the objective was to develop a SAV model that used generally available GIS data and parameters that were predictive and that could be customized for other estuaries that have GIS layers of existing SAV distribution. This report describes the procedure used to develop the SAV model for the Yaquina Bay Estuary, Oregon, appends a statistical script based on the open source R software to generate a similar SAV model for other estuaries that have data layers of existing SAV, and describes how to incorporate the model coefficients from the site-specific SAV model into SLAMM to predict the effects of sea level rise on *Zostera marina* distributions. To demonstrate the applicability of the R tools, we utilize them to develop model coefficients for Willapa Bay, Washington using site-specific SAV data.

Methods for Creating SAV Model for the Yaquina Bay Estuary

Study Areas

Most Pacific Northwest coastal watersheds are small and steep with limited coastal plains and narrow upstream valleys, leaving very little room for estuarine habitat to migrate. Though sea level is rising, along the Pacific coast it is mitigated in some areas by uplift which can vary from 1 mm to 4 mm per year (Mote et al., 2008). Tide ranges on the Pacific coast generally range about 3 meters, with extreme tides up to 4 meters. We selected the Yaquina Bay Estuary, Oregon (Figure 1) at latitude 44.62, longitude -124.03 to build the *Zostera marina* distribution model due to the large quantity of data available for this estuary. Additional information on the Yaquina Bay Estuary, its SAV habitats, and its associated watershed can be found in Lee and Brown (2009), Young et al. (2012), and Frazier et al. (2013). After generating the Yaquina model, as described below, we evaluated how well it performed in other estuaries by applying the model with the Yaquina coefficients to Willapa Bay, Washington. Willapa Bay is a larger estuary and has a less riverine shape than the Yaquina (Figure 1). However, the model performed poorly in Willapa when using the Yaquina coefficients. Then specific model coefficients were

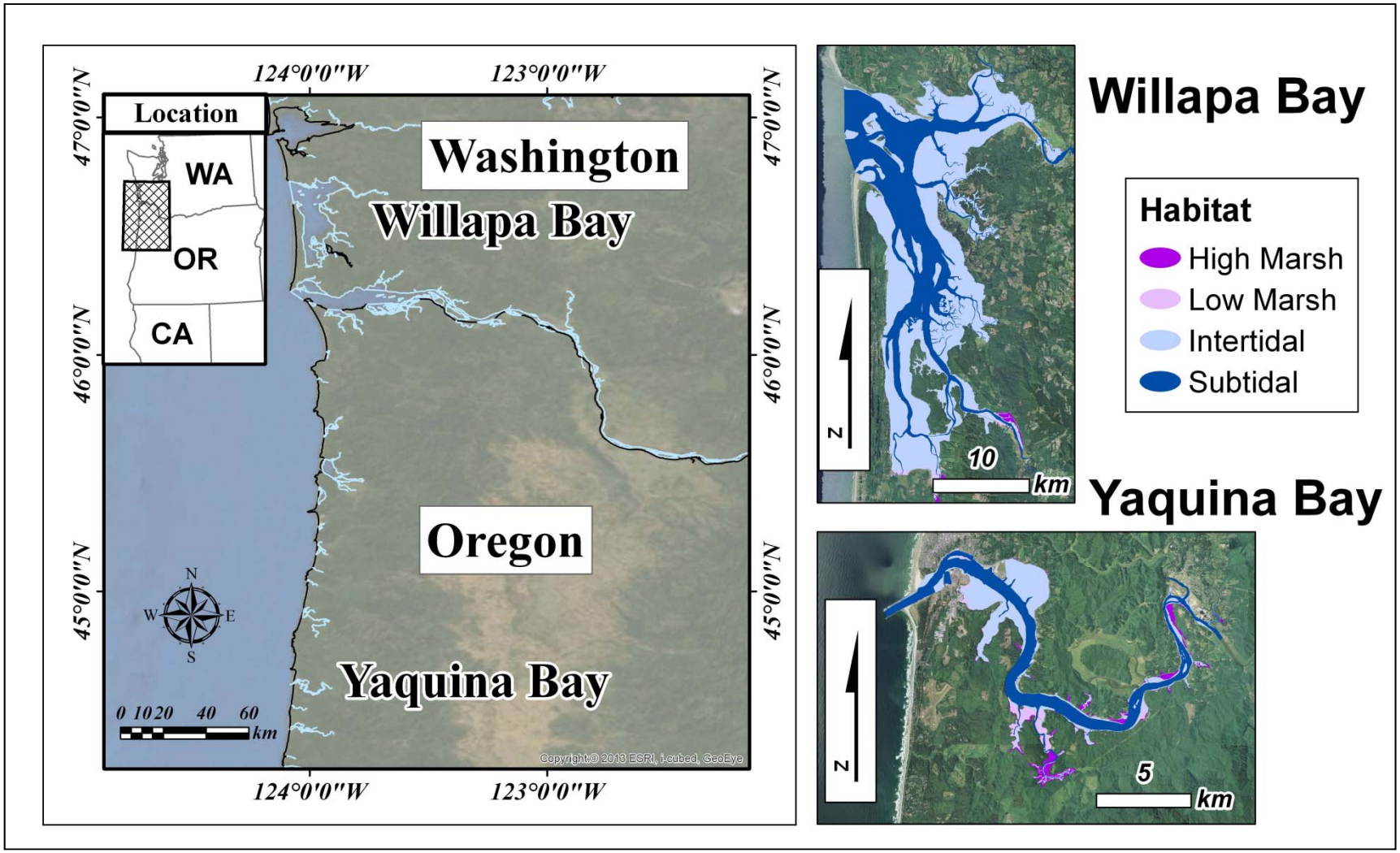


Figure 1: The map on the left shows the location of Willapa Bay, WA and Yaquina Bay, OR on the Northwest Pacific Coast of the U.S. Background data from ESRI and GeoEye. The maps on the right show both estuaries with National Agriculture Imagery Program (NAIP) in the background and the National Wetlands Inventory (NWI) data layer showing general estuarine habitat types in the estuaries. Upper right map background is Washington NAIP 2013 and Lower right map background is Oregon NAIP 2012.

developed for Willapa Bay using a site-specific SAV data layer to evaluate the functionality of the R tools and improve model performance for Willapa Bay.

Data

The GIS layers used to develop the current and predicted distribution model for SAV included five input raster datasets that represented topography, proximity between important features, and the current distribution of seagrass. Specifically, these rasters were:

1) 4 m raster Digital Elevation Model (DEM): A specialized process was used to combine topographic and bathymetric data, and a detailed description of the methods to create this raster layer is found in the following section: Topographic-Bathymetric (Topobathy) Layer.

2) 4 m raster Presence/absence of current distribution of *Zostera marina:* A detailed description of methods is found in the following section: Response Variable-Seagrass Occurrence.

3) 4 m raster distance to mouth – calculated from x to y: A detailed description of the methods for this calculation is included in the following section: Distance to Mouth.

4) 4 m raster Distance to Mean Lower Low Water (MLLW): A detailed description of the methods used to calculate this metric is included in "Appendix A: New functionality for predicting changes in submerged aquatic vegetation" under the section titled "Calculate New GIS Layers.

5) 4 m raster Distance to Mean Higher High Water (MHHW): See above for reference to methods in Appendix A.

For model development, the distance to MLLW and MHHW raster layers were generated within the statistical program R 2.15.3 (R Development Core Team, 2014) assuming MLLW and MHHW is fixed throughout the estuary based on the epochal datum established at South Beach, Oregon (Appendix B). However, within the *Z. marina* occupancy zone in the Yaquina Bay, the epochal datum varies by as much as 0.2 ft. (0.061 m; John Bauer, pers. comm.). Though this is small compared to most SLR projections for Yaquina it may be larger in other estuaries. If the estuary under investigation has significant variation in MLLW/MHHW within the estuary relative to the vertical datum, we suggest a different methodology be used for creating the MLLW/MHHW raster layers for use within the regression model. Within the SLAMM software, new MLLW/MHHW values are generated at each time step. The SLAMM technical reference and user's guide provide directions for incorporating variation in the tidal datum using the National Oceanographic and Atmospheric Administration (NOAA) VDATUM tool within the SLAMM modeling software.

Topographic–Bathymetric (Topobathy) Layer

The most consistently available coastal digital elevation sources in the Pacific Northwest are LIDAR data (http://www.oregongeology.org/sub/projects/olc/). The LIDAR for Yaquina Bay was resampled from ~1 meter (3 international feet) to 4 meters to reduce the number of computations required for the model to process at each time step. LIDAR is used for marsh and land elevations but is unreliable for other intertidal elevations so the LIDAR was erased by National Wetland Inventory (NWI: http://www.fws.gov/wetlands/) open water and intertidal aquatic bed, streambed or unconsolidated shore polygons. The most consistently available PNW coastal bathymetric data are National Oceanic

Atmospheric Administration (NOAA) Tsunami Digital Elevation Models (DEMs: http://www.ngdc.noaa.gov/mgg/inundation/). The DEM for Yaquina Bay was resampled from ~8.37 meters to 4 meters to match the LIDAR data and then masked by the same NWI polygons used to erase the LIDAR. The erased LIDAR and masked Tsunami DEM were then integrated into a seamless topobathy DEM by converting the raster layers to points, merging the points and reconverting the results to a raster (Figure 2). All cell values exceeding a maximum elevation of 20 meters above NAVD88 in the topobathy DEM were converted to NoData to reduce model processing time.

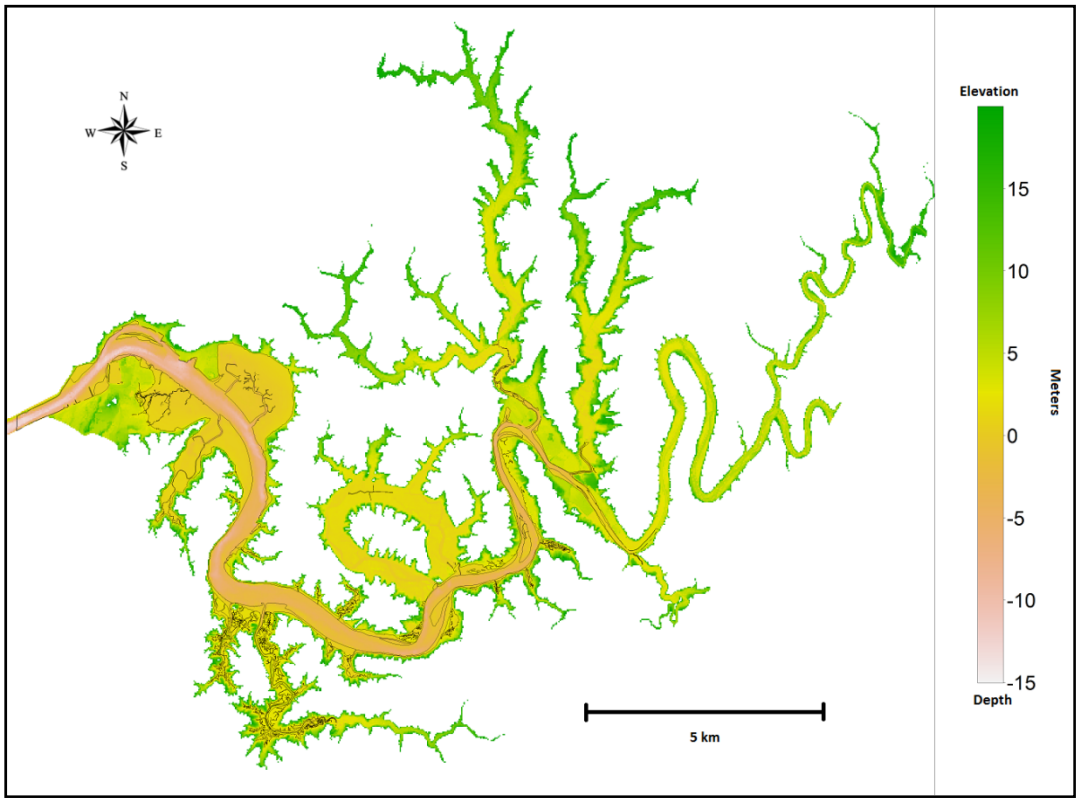


Figure 2: 4-m Topographic-Bathymetric (Topobathy) Digital Elevation Model (DEM) of Yaquina Bay Estuary, Oregon used to develop the SAV model. Vertical datum is NAVD88. Data integrated by U.S. EPA from several data sources (see text). Some areas shown above are behind dikes; these areas were not included in the generation of the SAV model. See section "Creating Random Extraction Points" in Appendix A for an example of where random points for developing the SAV model were taken.

Response Variable–Seagrass Occurrence

The *Zostera marina* presence or absence classification raster was created for the Yaquina Bay Estuary from color infrared aerial photography acquired at extreme low tide in 2007 as described in Clinton et al, 2007. The raster was resampled from 0.25 m resolution to 4 m resolution using the focal statistics tool with the sum option in Spatial Analyst (Figure 3). For each 4 m cell, a 16 by 16 matrix of 0.25 m cells (total of 256) were summed and values greater than or equal to 64 (25%) were considered present, all others were considered absent in the 4 m grid.

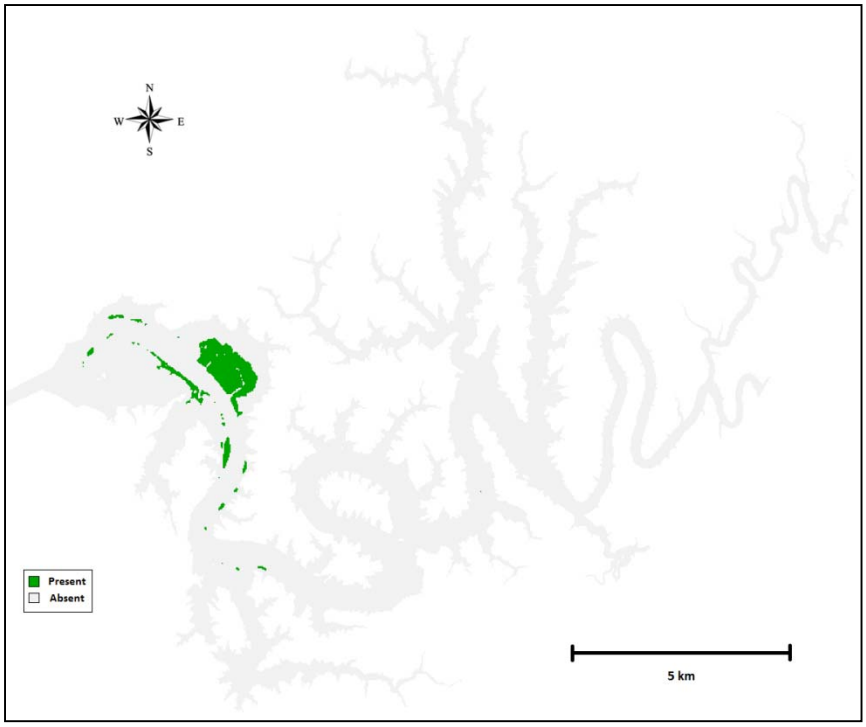


Figure 3: Submerged aquatic vegetation (*Zostera marina)* presence (green)/absence (grey) data for Yaquina Bay Estuary, Oregon used to develop the SAV model. Data from aerial surveys by the U.S. EPA (see Clinton et al., 2007 and Young et al., 2012). Some grey areas shown above are behind dikes; these areas were not included in the generation of the SAV model. See section "Creating Random Extraction Points" in Appendix A for an example of where random points for developing the SAV model were taken.

Distance to Mouth

Two spatial datasets were used to create the distance to mouth 4 m raster; (1) a single point shapefile situated at the mouth of the estuary often referred to as a pour point and (2) a constant raster layer with all cell values set to 1 with the same cell size and extent as the topobathy raster. The Cost Distance Tool (ArcToolbox > Spatial Analyst > Distance) was used to generate the least accumulative cost distance to the pour point at the mouth along the water path. (Cost distance to mouth is the shortest distance to the mouth while being restricted to being in the estuary, and calculates distance around islands, peninsulas, and sinuous bends in the estuary.) The pour point was entered as the source feature data and the constant raster as the cost raster. Cell locations with NoData in the Input cost raster act as barriers in the cost surface (i.e., cells above 20 m in the topobathy DEM). The output was a 4 m raster layer with each grid cell containing the distance to the mouth of the estuary from the centroid of the cell (Figure 4) along the water path from that cell to the ocean.

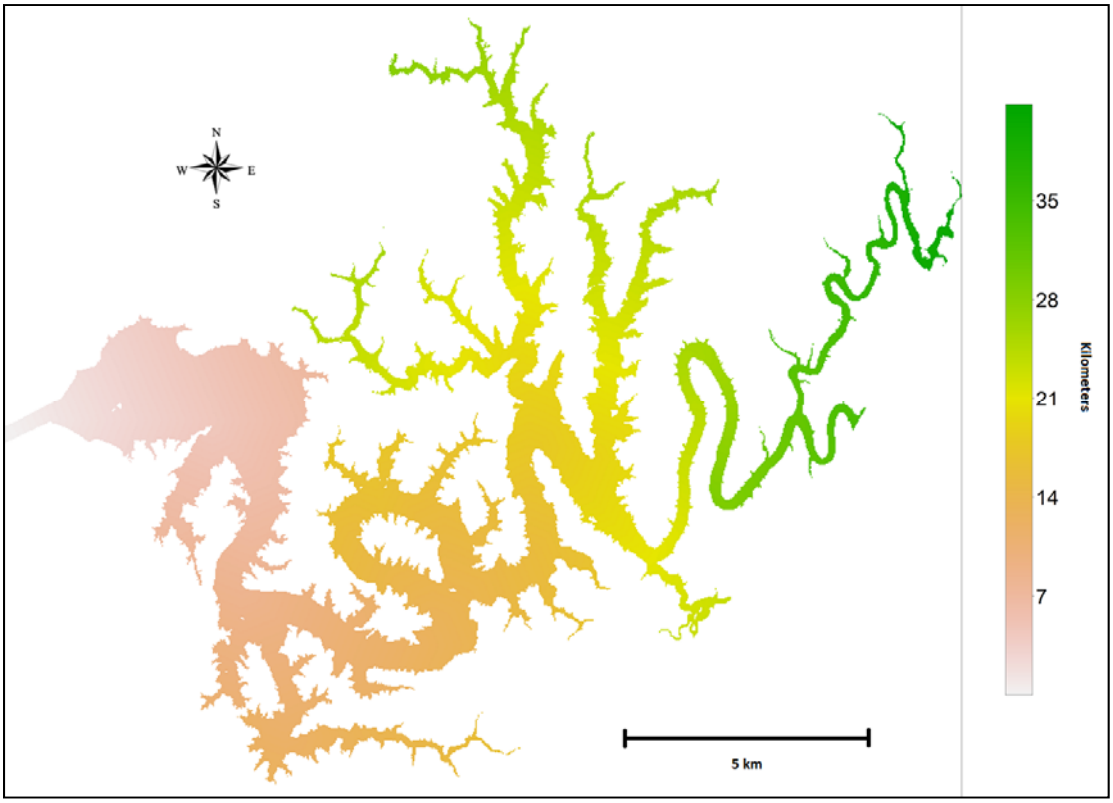


Figure 4: 4-m Cost distance to mouth raster layer used in the development of the SAV model, and the input into the new version of the SLAMM model. Source layer created by Patrick Clinton.

Statistical Methods

Models of varying level of complexity have been developed to model the distribution of seagrass habitat. For example, Kairis and Rybczyk (2010) developed a mechanistic model to simulate changes in seagrass distribution as a function of sea level rise in Padilla Bay, Washington. However, we chose to use a generalized linear model (GLM) as a general modeling approach that could be applied to other estuaries using available GIS parameters. GLMs are flexible generalizations of linear regressions that allow for response variables with non-normal error distribution models. Specifically, we used a logistic regression model, which has been used successfully to model the distribution of *Zostera* in Europe (Bekkby et al., 2008; van der Heide et al., 2009).

The first step in determining which environmental variables are associated with the distribution of *Zostera marina* is to generate a set of random sample points throughout the estuary within the extent of the topobathy raster. The number of points is not important as long as the range of predictor and response variables are adequately represented. For Yaquina Bay, 1000 random sample points were generated where at least 10% of the points included had *Z. marina* present. The range of predictor values were evaluated in R. The selection, creation and evaluation of the 1000 random sample points for Yaquina Bay Estuary are explained in "Appendix A: New functionality for predicting changes in submerged aquatic vegetation" under the sections titled "Creating Random Extraction Points", "Extracting Data from each Parameter Layer", and "Data Confirmation".

Creating the Model for the Yaquina Bay Estuary

GLMs were used to identify the regression relationship between the current distribution of *Zostera marina* in Yaquina Bay using the following explanatory variables: elevation, distance to mouth, distance to mean lower low water (MLLW), and distance to mean higher high water (MHHW). All analyses were performed in R 2.15.3 (R Development Core Team, 2014). Four different models were created using some or all of the explanatory variables. All the models where then compared using the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). In general, the AIC and BIC provide a prediction score with penalties for each parameter used in the model (Burnham and Anderson, 2002). Lower AIC or BIC scores are preferred. When the penalty for a parameter is greater than the benefit of including that parameter, the simpler model is preferred. Complete details of each model and their comparisons can be found in "Appendix A: New functionality for predicting changes in submerged aquatic vegetation" under the section titled "Model Comparison".

Model Validation

Cross validation was used to evaluate model performance. With cross validation, a small portion of the data was held in reserve and parameter estimates were obtained using the remaining data. The model was then applied to the reserve data and the predicted values were compared to the observed values. This was repeated numerous times with different portions of the data being held in reserve to obtain a better estimate of model performance. A complete description and scripts for performing cross validation on the four models evaluated can be found in "Appendix A: New functionality for predicting changes in submerged aquatic vegetation" under the section titled "Model Validation". Of the models explored, one model (Model 2) performed the best in describing the distribution of *Zostera marina* in Yaquina Bay. This model was incorporated into the SLAMM modeling software and is the statistical model (but not coefficients) used for other estuaries.

Yaquina SAV Model

The following is the best model to predict *Zostera marina* distribution in the Yaquina Bay Estuary:

PrSAV = 1/(1+exp(-(***Intercept*** + DEM*(***DEM coefficient***) + DEMsq*(***DEMsq coefficient***) +

D2MLLW*(***D2MLLW coefficient***) + D2MHHW*(***D2MHHW coefficient***) + D2Mouth*(***D2Mouth coefficient***)

+D2Mthsq*(***D2Mthsq coefficient***))))

Where:

PrSAV = the probability a cell has *Zostera marina*;

DEM, DEMsq = DEM and DEM squared with vertical datums of NAVD88 (meters);

D2Mouth, D2mthsq = distance to estuary mouth from the centroid of the cell following the water path and that distance squared in meters. Cost-path methodology was used to calculate the distance;

D2MLLW,D2MHHW = distance to mean lower low water and mean higher high water for each cell as derived by SLAMM in each time step (meters).

Generation of Site-Specific Coefficients for the SAV Model for New Estuaries

The basic statistical model developed for the Yaquina Bay Estuary is used to predict SAV distributions in other estuaries. However, site-specific coefficients for the model need to be generated for each new estuary. The first step in assessing the effects of sea level rise on *Zostera marina* in a new estuary is to select or create the five required GIS data layers as described within this report in the section titled "Data". In most cases, these data will exist in some form but may need some preprocessing to ensure that all layers have the same:

1. Projection
2. Vertical datum
3. Resolution
4. Extent

The data layers must overlay each other exactly with the same number of rows and columns starting at the same location in the upper left corner and ending in the same location in the lower right corner. Then using the R script in "Appendix A: New functionality for predicting changes in submerged aquatic vegetation", generate a logistic model. This site-specific model will have the same structure as the Yaquina Bay model but will have different values for the coefficients as can be seen in Table 1 for Willapa Bay. A GIS layer of SAV distribution in Willapa Bay was available from the USDA from a 2005 aerial survey.

Table 1: Coefficients from logistic Model 2 for Yaquina Bay, OR and Willapa Bay, WA. See Appendix A, Section Model Validation for description of the model. Site-specific SAV distributions for Willapa Bay were available from the USDA.

	Yaquina Bay	**Willapa Bay**
Intercept	-3.47	-2.038
DEM coefficient	-1.54	-1.733
DEMsq coefficient	-1.31	0.1735
D2MLLW coefficient	-0.0126	-0.0001607
D2MHHW coefficient	0.00415	-0.0003253
D2Mouth coefficient	0.00105	0.0002676
D2mthsq coefficient	-0.0000000886	-0.000000006601

The new model will predict the current probability of occurrence of *Z. marina* in each grid cell in the target estuary. In evaluating the model, the user can use different threshold probabilities to indicate presence. Experience in modeling Yaquina Bay and Willapa Bay suggests that very low probabilities (0.01-0.05) can overestimate the distribution of SAV, and probabilities in the neighborhood of 0.25-0.30 often give the best fit (Figure 5 and 6).

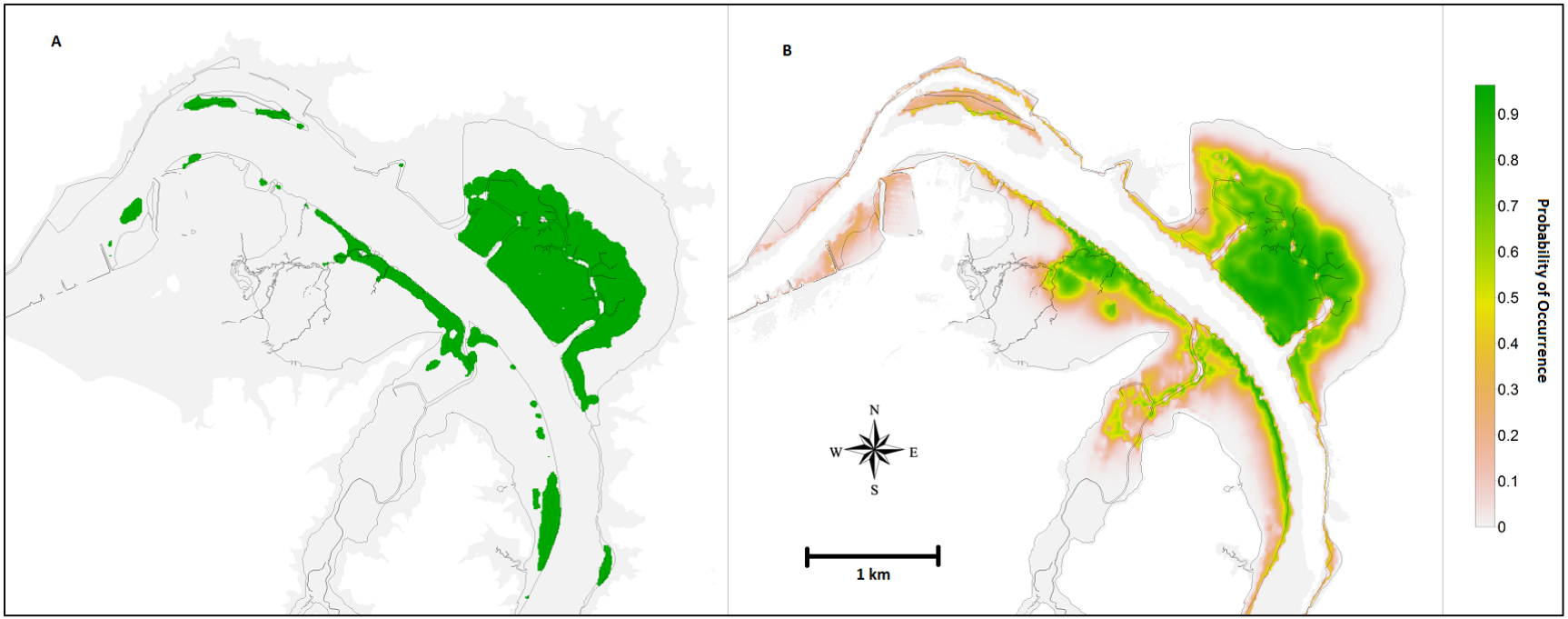


Figure 5: The green area in map A indicates the current distribution of *Z. marina* in the Yaquina Bay Estuary. Map B is the current predicted probability of occurrence of *Z. marina* using Model 2 from Appendix A. Note that all probabilities are mapped and the lighter beige indicates very low probability of *Z. marina* occurrence.

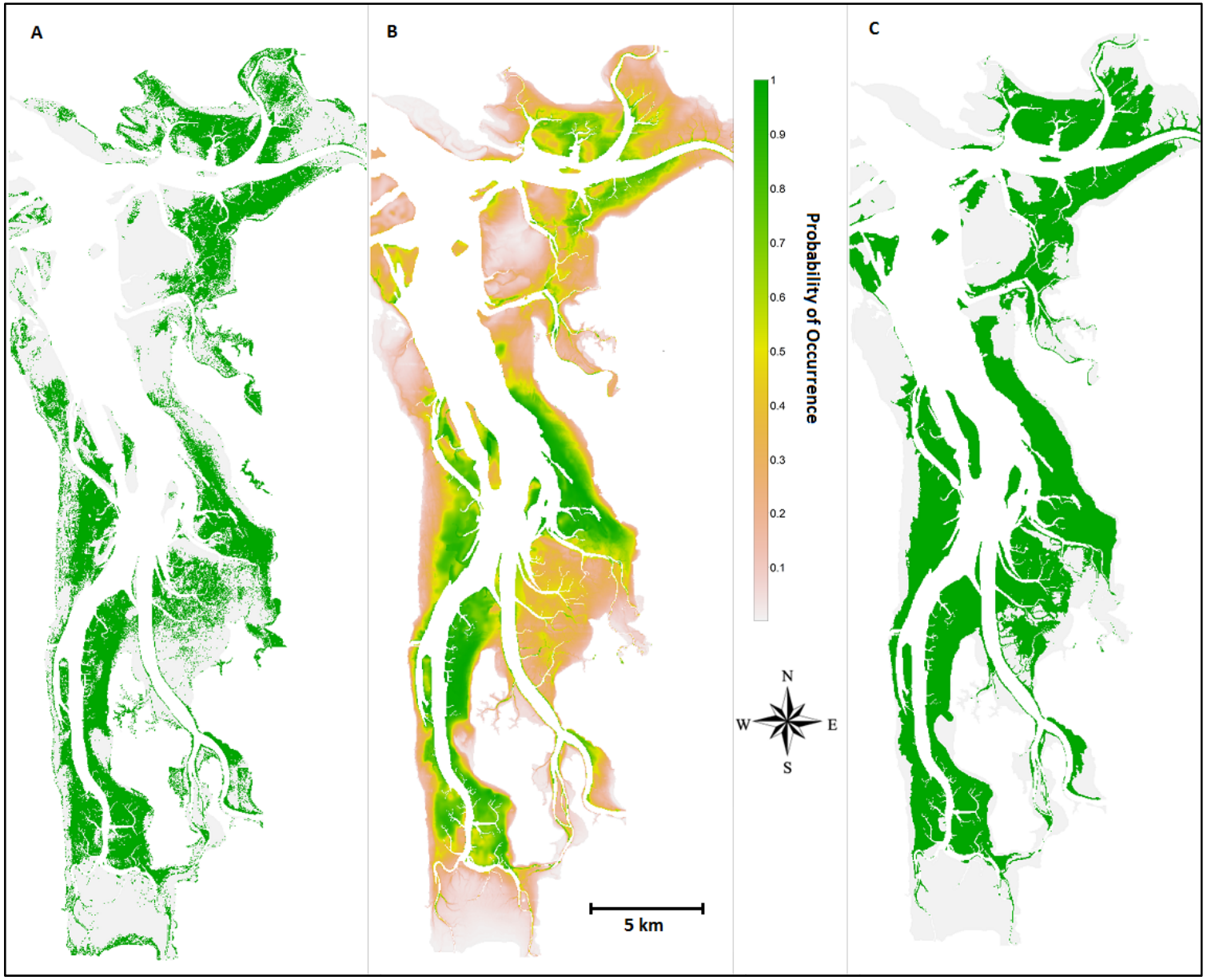


Figure 6: The green area in map A shows current distribution of *Z. marina* in Willapa Bay, WA. The gradient of areas in map B shows predicted probability of occurrence of *Z. marina* presence in Willapa Bay, WA. The green area in map C shows *Z. marina* probabilities =>30% turned to presence in Willapa Bay.

Integration of the SAV Model into the SLAMM Modeling Software

Based on the Yaquina Bay SAV model, Warren Pinnacle Consulting modified the SLAMM software to simulate changes in the distribution of seagrass with rising sea level based on the regression model described above. To access this functionality, a distance to mouth raster layer (as described above) must be identified in the SLAMM model setup form. The specified raster file should contain values for distance to the estuary mouth in meters, generally derived using "cost-path" calculations along a water path from the cell to the mouth of the estuary. When the distance to mouth layer is identified, the "SAV Parameters" option becomes available (Figure 7). The interface requires the user to specify the coefficients of the model, like those shown in Table 1, that describe the relationship between physical site-specific parameters and the probability of SAV in a given cell for a specific estuary. No default parameters are provided as currently new coefficients for a particular estuary must be generated using the R scripts provided in Appendix A before using the SAV model in SLAMM.

The MLLW and MHHW data for every grid cell is re-calculated at each time step within the SLAMM software. While a DEM is required for modeling changes in marsh habitat within SLAMM, an enhanced DEM with topographic and bathymetric data will provide better results for the SAV predictions.

The SLAMM modeling software allows the modeler to select either a particular SLR scenario to model or the expected amount of SLR for a particular place over a given time period. The user can also select the time steps that are calculated and output. So for example, one could model SLR of 1 m between 2000 and 2100 at 25 year time steps. Maps and spreadsheets for 2025, 2050, 2075 and 2100 would be generated as the model runs. The coefficients for SAV remain static, but the elevation and other habitat parameters are modified by SLAMM at each time step outputting a predicted distribution map for each time step and a spreadsheet containing changes in total area for each habitat.

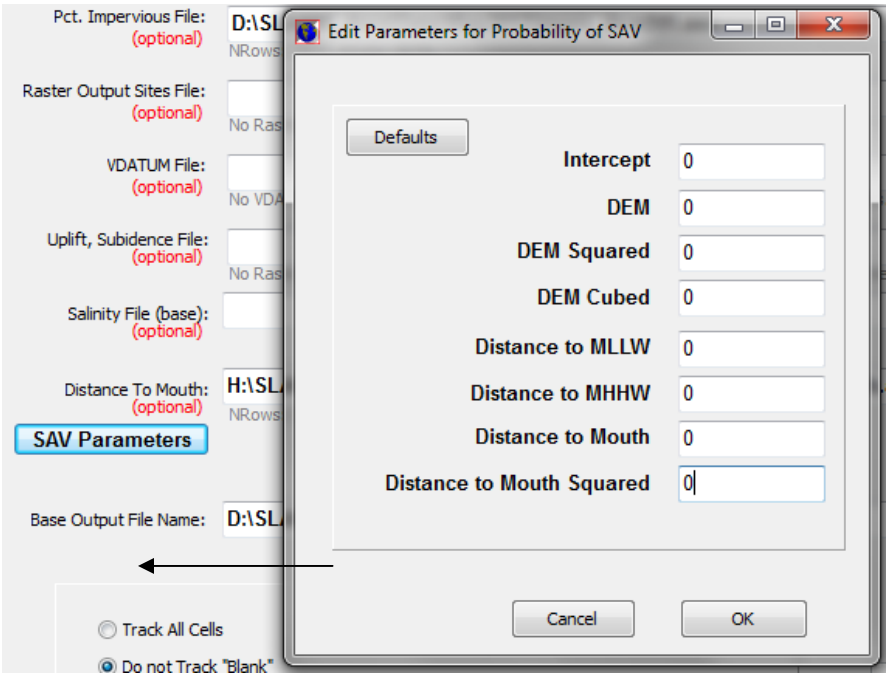


Figure 7: Input screen for the SAV enhancement in the SLAMM software. After a distance to mouth grid is entered, the SAV Parameters button is activated. The user then enters new parameter values based on a site-specific SAV model developed using the R script in "Appendix A: New functionality for predicting changes in submerged aquatic vegetation". Note that in this version, the "Defaults" button is not operational.

SLAMM Software:

SLAMM 6.3 includes the SAV model functionality as described above. The new software installer is located here: http://warrenpinnacle.com/prof/SLAMM6/.

Additional enhancements in SLAMM 6.3 include:

- The option to create SAV prediction maps and the estimation of total square km of SAV in each time step.
- The option to perform an SAV calculation through "Set Map Attributes".
- SAV coverage in square km by year is calculated and included in the output CSV file when the option to predict SAV is selected.
- SAV maps can be output to GIF or MS Word by selecting "Extra Maps" on the "Execute" window.

The SLAMM Technical Documentation and User's Manual have been updated to include the new SAV prediction functionality (Clough, 2012) and are included in the installer, which requires a 64-bit version of Windows. However, a 32-bit version of the installer is located here for those with who don't have 64-bit machines: http://warrenpinnacle.com/prof/SLAMM6/SLAMM6.3_32.exe.

Summary:

Based on known distributions of *Zostera marina* in the Yaquina Bay Estuary, we developed a logistic regression model, a type of GLM, to predict SAV distributions from readily available GIS parameters. This model was added as a new functionality in version 6.3 of SLAMM. The R script available in Appendix A describes how the original SAV model for Yaquina Bay was developed. It also provides a detailed methodology to develop site-specific model coefficients for other estuaries when existing SAV GIS data layers are available. Once the site-specific model coefficients are generated, they can be input into SLAMM to evaluate impacts of sea level rise on SAV distributions under different assumptions (e.g., accretion rates) and/or sea level rise scenarios. This new functionality in SLAMM provides a reasonable first-order approximation of how the distribution of *Zostera marina* will change in Pacific Northwest estuaries in response to sea level rise.

References:

Bekkby, T., Rinde, E., Erikstad, L., Bakkestuen, V., Longva, O., Christensen, O., Isæus, M., and Isachsen, P. E. 2008. Spatial probability modelling of eelgrass (*Zostera marina*) distribution on the west coast of Norway. *ICES Journal of Marine Science* 65:1093–1101.

Burnham, K.P. and Anderson, D.R. (2002), Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach (2nd ed.), Springer-Verlag, ISBN 0-387-95364-7.

Clinton, P.J., Young, D.R., and Specht, D.T. 2007. A guide to mapping intertidal eelgrass and nonvegetated habitats in estuaries of the Pacific Northwest USA. U.S. Environmental Protection Agency, Washington, DC, EPA/600/R-07/062, 2007. (http://cfpub.epa.gov/si/si_public_record_Report.cfm?dirEntryID=172424)

Clough, J. 2012. SLAMM 6.3 User's Manual. Warren Pinnacle Consulting, INC. Waitsfield, VT. http://warrenpinnacle.com/prof/SLAMM6/SLAMM_6.3_Users_Manual.pdf.

Craft, C., Clough, J., Ehman, J., Joye, S., Park, R., Pennings, S., Guo, H., and Machmuller, M. 2009. Forecasting the effects of accelerated sea-level rise on tidal marsh ecosystem services. *Frontiers in Ecology and the Environment* 7:73-78.

Ferraro, S.P. and Cole, F.A. 2007. Benthic macrofauna-habitat associations in Willapa Bay, Washington, USA. *Estuarine, Coastal and Shelf Science* 71:491-507.

Frazier, M.R., Reusser, D.A., Lee II, H., McCoy, L.M., Brown, C., and Nelson, W. 2013. WestuRe: U.S. Pacific Coast estuary/watershed data and R tools. U.S. EPA, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Western Ecology Division. EPA/600/R/13/067.

Geselbracht. L., Freeman, K., Kelly, E., Gordon, D.R., and Putz, F.E.. 2011. Retrospective and prospective model simulations of sea level rise impacts on Gulf of Mexico coastal marshes and forests in Waccasassa Bay, Florida. *Climatic Change* 107:35–57.

Glick, P., Clough, J., and Nunley, B. 2007. Sea-level Rise and Coastal Habitats in the Pacific Northwest: An Analysis for Puget Sound, Southwestern Washington, and Northwestern Oregon, National Wildlife Federation, Seattle, Washington, 94 pp.

Kairis, P.A. and Rybczyk, J.M. 2010. Sea level rise and eelgrass *(Zostera marina*) production: A spatially explicit relative elevation model for Padilla Bay, WA. *Ecological Modelling* 221:1005-1016.

Lee II, H. and Brown, C.A. (editors). 2009. Classification of regional patterns of environmental drivers and benthic habitats in Pacific Northwest estuaries. U. S. EPA, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Western Ecology Division, Newport OR. 298 pp. EPA/600/R-09/140. http://nepis.epa.gov/Adobe/PDF/P1006Q2H.PDF.

Mote, P., Petersen, A., Reeder, S., Shipman, H., and Binder, L.W. 2008. Sea level rise in the Coastal Waters of Washington State. University of Washington Climate Impacts Group & Washington State Department of Ecology. 11 pp.

Phillips, R.C. 1984. The ecology of eelgrass meadows in the Pacific Northwest: A community profile. U.S. Fish Wildl. Serv. FWS/OBS-84/24. 85 pp.

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Shaughnessy, F.J, Gilkerson, W., Black, J.M., Ward, D.H., and Petrie, M. 2012. Predicted eelgrass response to sea level rise and its availability to foraging Black Brant in Pacific coast estuaries. *Ecological Applications* 22: 1741-1763.

Short, F.T. and Neckles, H.A. 1999. The effects of global climate change on seagrasses. *Aquatic Botany* 63: 169-196.

Van der Heide, T., Peeters, E.T.H.M, Hermus, D.C.R., van Katwijk, M.M., Roelofs, J.G.M., and Smolders, A.J.P. 2009. Predicting habitat suitability in temperate seagrass ecosystems. *Limnology and Oceanography* 54:2018-2024.

Williamson, K.J. 2006. Relationships between eelgrass (*Zostera marina*) habitat characteristics and juvenile Dungeness crab (*Cancer magister*) and other invertebrates in southern Humboldt Bay, California. MS thesis, Humboldt State University, Arcata, California. 56 pp.

Young, D., Clinton, P., Specht, D., Mochon Colura, T.C., and Lee II, H. 2012. Determining bathymetric distributions of the eelgrass Zostera marina L. in three turbid estuaries on the eastern North Pacific coast. *Botanica Marina* 55:229-240.

SLAMM - New Functionality for Predicting Changes in Submerged Aquatic Vegetation

Appendix A

Authors: Lee McCoy [1], Melanie Frazier [2], Debbie Reusser [3], Henry Lee [4], Pat Clinton [4], Erin Graham [1], Brett Dumbauld [1], Jonathan S. Clough [5]

Institutions:

1. Agricultural Research Service, United States Depatment of Agriculture, Newport, OR 97365
2. National Center for Ecological Analysis and Synthesis, Santa Barbara, CA
3. Western Fisheries Research Center, U.S. Geological Survey, Newport, Oregon, 97365
4. Pacific Coastal Ecology Branch, Western Ecology Division, United States Environmental Protection Agency, Newport, Oregon, 97365
5. Warren Pinnacle Consulting, Inc.,Warren, VT 05674

Date: 2014-09-18

R version 3.1.1 (2014-07-10)

Script Version: 1.0

Instructions for Creating Coefficients and Confirming Parameters

This document describes the selection of the model parameters and coefficients that were incorporated into SLAMM to predict how the distribution of *Zostera marina* will change in response to sea level rise for Yaquina Bay, Oregon. This information can also be used to refine the Yaquina model as more data becomes available or to develop models for new estuarine systems.

Specifically, this document provides detailed instructions on how to use the statistical programming language R to:

1. create new coefficients
2. confirm or develop new parameters

Set the Working Directory

The working directory is the default location where R will search for existing files and save generated files. For this exercise the file structure should consist of a main folder **SeaLevelRise**, which will be set as the working directory, and a sub-folder, **YaquinaGIS** where original and generated GIS files are stored.

This structure is only for convenience. Files can be stored and created in different folders, but the file path needs to be explicitly defined for each file that is not in the working directory folder in any read/write command. More on this later.

Folders need to be either separated with two backslashes \ \ or a single forward slash /. Valid working directory formats are:
setwd("C:\ \TopFolder\ \SubFolder\ \SubSubFolder")
setwd("C:/TopFolder/SubFolder/SubSubFolder")

```
setwd("C:\\Users\\Projects\\SeaLevelRise")
```

Install or Load Packages

Packages are extra code that extend the functionality of R. There are thousands of different packages - many of them with overlapping functionality. For this exercise, two packages are used.

- rgdal : adds GIS functionality
- raster : adds visualization/analysis functionality for raster files

Packages only need to be installed on a computer once but must be loaded for each script in which they will be used. Since installing only needs to be done once, these lines of code have been commented out below (by putting a # in front of each line of code). If these packages have not been installed, simply remove the # and run each line. Alternately, the **Packages** tab and **Install Packages** can also be used in the lower right hand pane in RStudio.

Installing Packages

```
#  install.packages("rgdal")
#  install.packages("raster")
```

Loading Packages

```
  library(rgdal)
  library(raster)
```

Required GIS Layers

To develop the model, several GIS files are necessary. There are two main types of GIS files - vector files (points, lines, polygons) and raster files (images and single layer grid data). It is useful to know the difference between these and there are a number of excellent explanations online.

Polygon Files

A polygon file that defines the area of the estuary is the only vector file used in developing this model.

The following code creates a variable called **estuary** that contains the polygon shapefile **YBarea**. If this polygon file is not in the **YaquinaGIS** sub-folder of the working directory, change dsn="filepath" to indicate the location of the file. Both of the following lines of code do the same thing. The second one only works if the working directory has been correctly designated.

Use one or the other of these to read in the estuary polygon:

```
  #designating the entire file path
    estuary <- readOGR(dsn="C:\\Users\\Projects\\SeaLevelRise\\YaquinaGIS",
                       layer="YB_area")
```

```
  #designating the path relative to the working directory
    estuary <- readOGR(dsn="YaquinaGIS", layer="YB_area")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "YaquinaGIS", layer: "YB_area"
## with 1 features and 1 fields
## Feature type: wkbPolygon with 2 dimensions
```

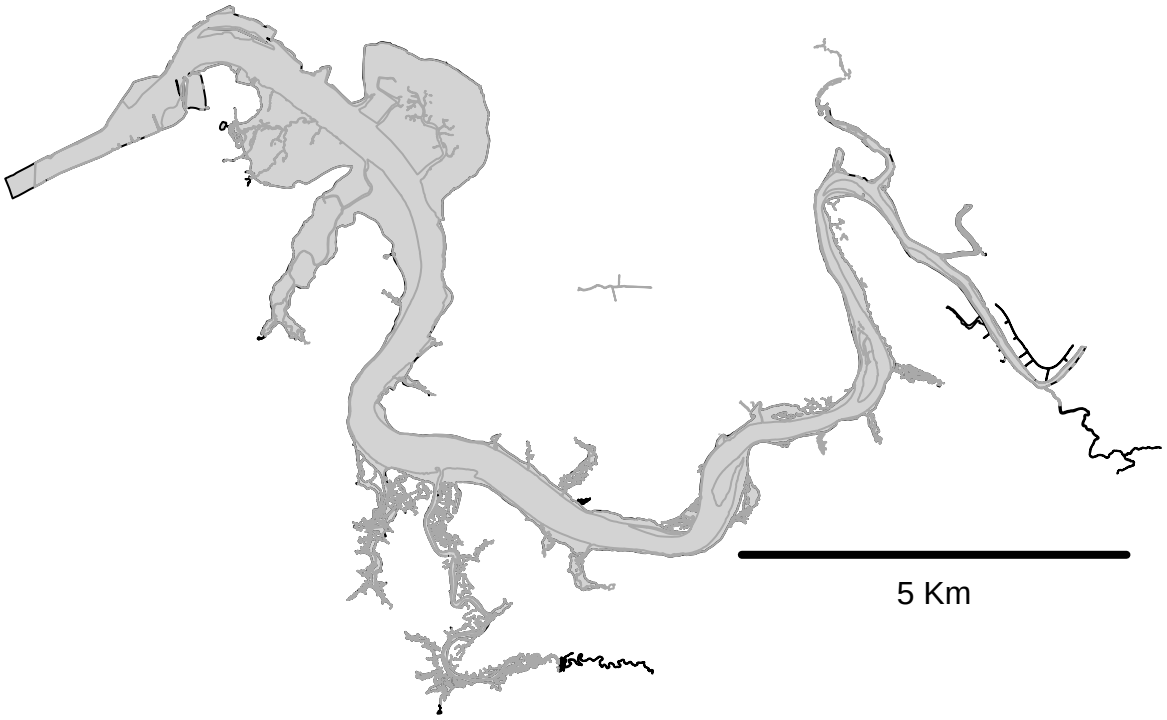
Bring in a tideflat layer. This layer isn't necessary for the analysis, but it helps for visualization.

```
tideflat <- readOGR(dsn=dsn="C:\\Users\\Projects\\SeaLevelRise\\YaquinaGIS",
                    layer="YB_intertidal")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\leemccoy\GIS\Yaquina\Background\YB_Intertidal", layer: "YB_intertidal"
## with 460 features and 39 fields
## Feature type: wkbPolygon with 2 dimensions
```

Plot the estuary layer and the tideflat:

```
    par(mar=c(0,0,0,0))
    plot(estuary, col="lightgray")
    plot(tideflat, border="darkgray", add=T)
          segments(998000, 321000, 1003000, 321000, lwd=4)
          text(1000500, 320500, "5 Km")
```



Raster Files

Raster files are pixel based data in which the size of each pixel defines the resolution (e.g., 4 X 4 meter pixels) and the value of each pixel denotes the level (e.g., color, elevation, temperature, salinity, distance from something else). Resolution is important because the entire area depicted by a pixel has the same value (e.g., a 4 X 4 meter pixel can only have one elevation value even though micro-topographical features may vary within this area, such as a small channel, small patch of eelgrass, freshwater seep, etc). Higher resolution (smaller pixel size) allows for finer features to be defined but creates larger files for a given area. Lower resolution (larger pixel size) creates coarser features and smaller files for the same area. In general, the pixel size for the analysis will be restricted to the resolution of the lowest or coarsest raster. For this tool, all of the rasters need to be in the same coordinate system (e.g., UTM), and have the same extent and resolution (e.g., 4 meter by 4 meter pixels). One technique for confirming that the rasters match will be demonstrated below. The techniques for transforming, standardizing, and reprojecting rasters is beyond the scope of this exercise.

Three original raster files are necessary for developing this model:

1. **Digital elevation model (DEM)** of the bathymetry of the estuary. This layer was created by merging the intertidal bathymetry with a LiDAR digital elevation model of the adjacent uplands to 20 meters. The bathymetry layer, which was referenced to mean lower low water (MLLW) had to be adjusted to NAVD88 (the vertical scale for upland data) before being merged with the upland DEM.
2. **Cost distance to mouth** is the shortest distance to the mouth while being restricted to being in the estuary. This allows proper calculation of distance around objects such as islands, peninsulas, and sinuous bends in the estuary. This is the same as the distance a fish would have to travel to get from the mouth of the bay to each pixel.
3. **Current *Zostera marina*** is a presence/absence raster interpreted from color infrared aerial photography.

Several additional raster files are generated from these base files including squared and cubed variations of elevation and distance to mouth, and the distance to mean lower low water (MLLW) and mean higher high water (MHHW). These files are created in this script. It is possible that generating a new statistical model incorporating additional raster layers (e.g., salinity, velocity) could improve model performance, though we note that GIS data layers of these parameters are often not available and their inclusion would reduce the generality of the model.

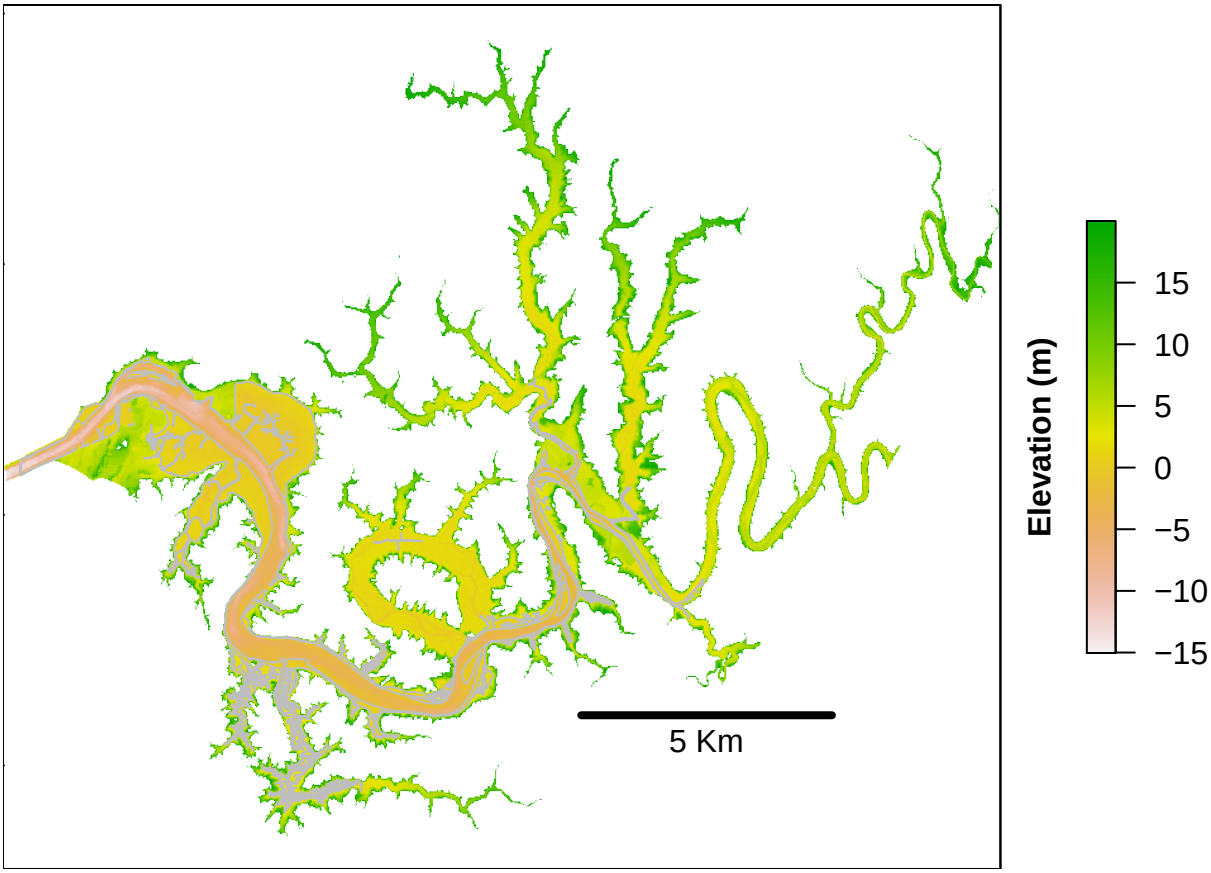
As mentioned earlier, these raster GIS files are in the **YaquinaGIS** sub-folder of the working directory. The filepath from the working directory has to be explicitly defined by adding "YaquinaGIS\ \" before the file name.

Load the required raster files:

```
#elevation
  dem <- raster("YaquinaGIS\\yaqslammdemG")
#distance to mouth
  d2m <- raster("YaquinaGIS\\dist2mouth")
#current zostera marina
  zm <- raster("YaquinaGIS\\zmarinanow")
```

Plot the elevation layer:

```
  plot(dem, legend=FALSE)
    plot(tideflat, border="gray", add=T)
    plot(dem, legend.only=T, legend.width=1, legend.mar=3,
         axis.args=list(at=seq(-15, 15, 5), labels=seq(-15, 15, 5), cex.axis=1),
         legend.args=list(text="Elevation (m)", side=4, cex=1, line=-2.5, font=2))
          segments(1000000, 321000, 1005000, 321000, lwd=4)
          text(1002500, 320500, "5 Km")
```



Explore the elevation layer:

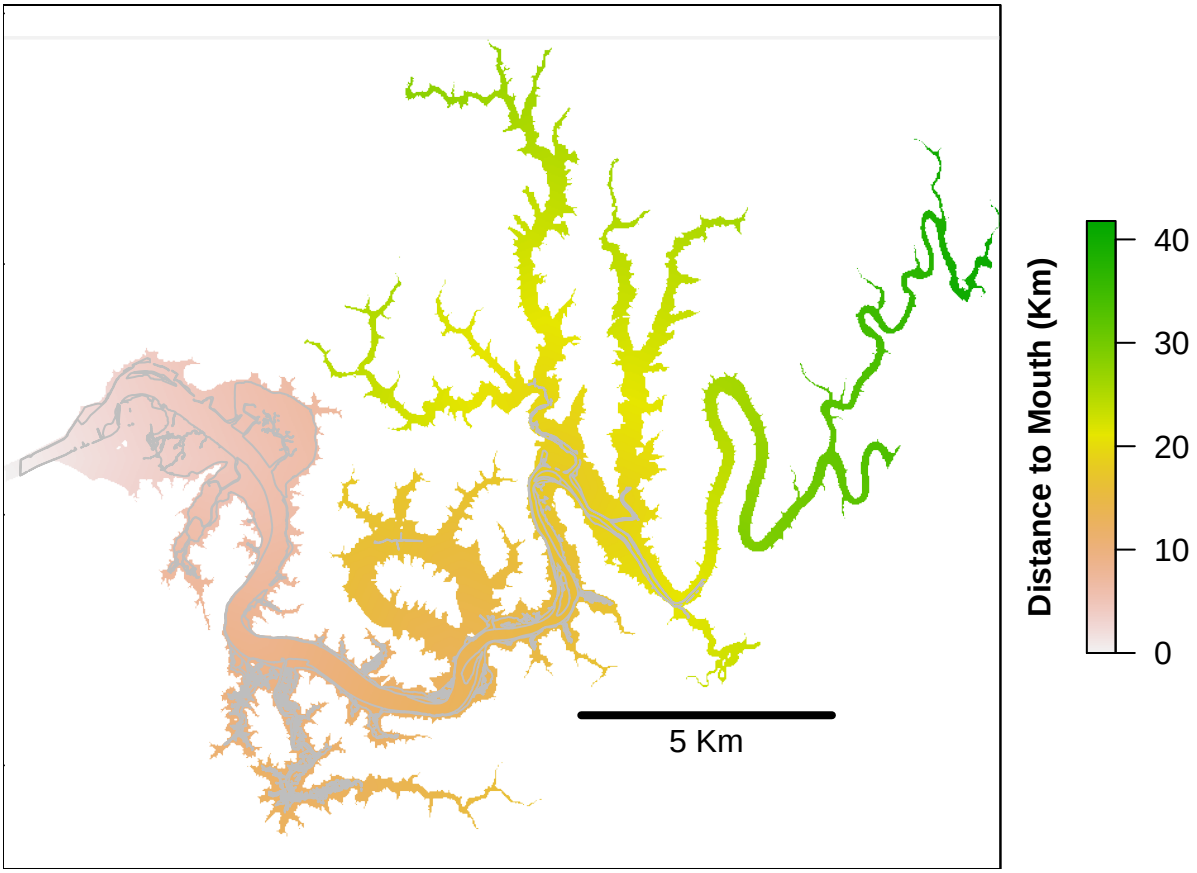
```
dem
```

```
## class       : RasterLayer
## dimensions  : 3993, 4974, 19861182  (nrow, ncol, ncell)
## resolution  : 4, 4  (x, y)
## extent      : 988465, 1008361, 318565, 334537  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=lcc +lat_1=43 +lat_2=45 +lat_0=41.75 +lon_0=-120 +x_0=1312335.958005249
 +y_0=0 +ellps= ## data source : C:\Users\leemccoy\Projects\Stuff4Others\SLR\YaquinaGIS\yaqslammdemG
## names       : yaqslammdemG
## values      : -15.51, 31.79  (min, max)
```

This shows the number of rows (3993) and columns (4974) and the total number of cells (1.9861×107) in this raster, as well as the resolution (4, 4 meters), extent, coordinate system (lat/long with a bunch of other stuff), file path, and range of values (in meters relative to MLLW).

Plot the distance to mouth layer:

```
plot(d2m, legend=FALSE)
    plot(tideflat, border="gray", add=T)
    plot(d2m, legend.only=T, legend.width=1, legend.mar=3,
         axis.args=list(at=seq(0, 40000,10000), labels=seq(0, 40, 10), cex.axis=1),
         legend.args=list(text="Distance to Mouth (Km)", side=4, cex=1, line=-2.5,
                          font=2))
          segments(1000000, 321000, 1005000, 321000, lwd=4)
          text(1002500, 320500, "5 Km")
```



Explore the distance to mouth layer:

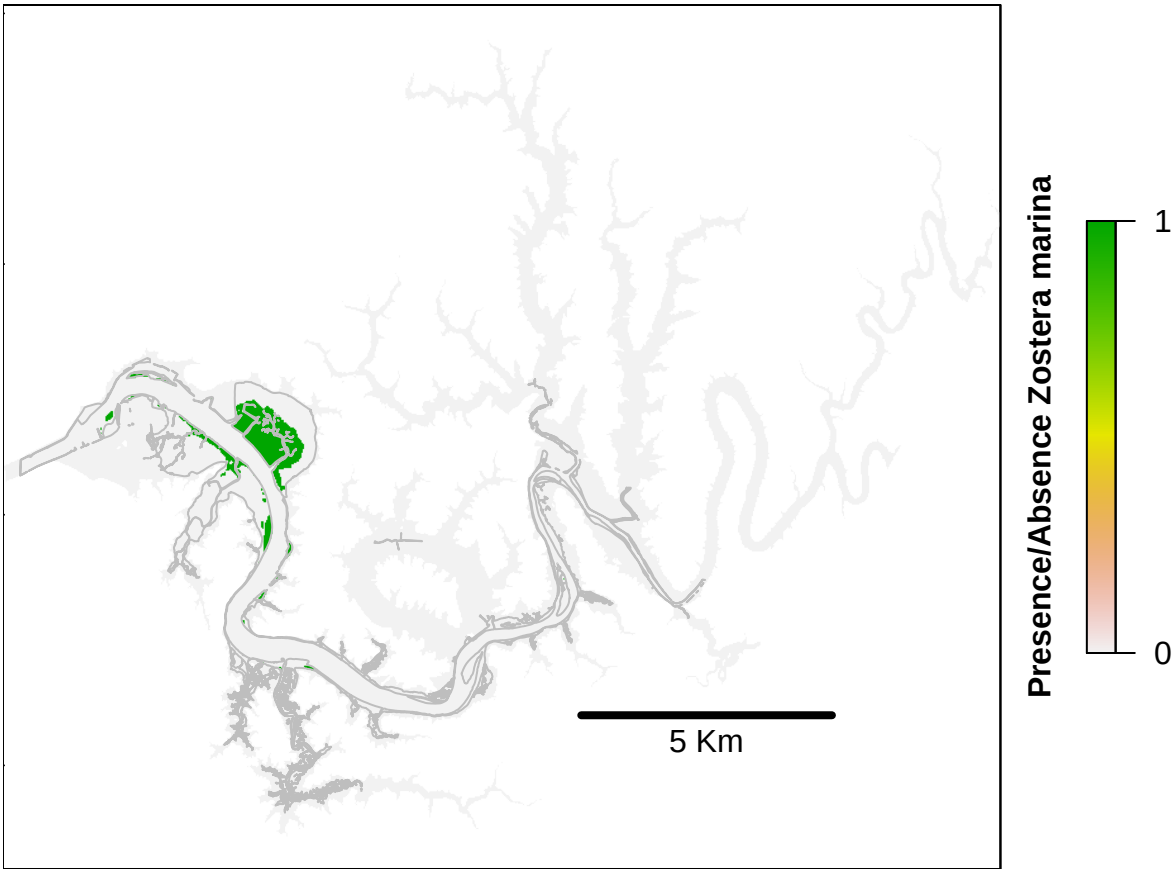
```
d2m
```

```
## class       : RasterLayer
## dimensions  : 3993, 4974, 19861182  (nrow, ncol, ncell)
## resolution  : 4, 4  (x, y)
## extent      : 988465, 1008361, 318565, 334537  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=lcc +lat_1=43 +lat_2=45 +lat_0=41.75 +lon_0=-120 +x_0=1312335.958005249
 +y_0=0 +ellps= ## data source : C:\Users\leemccoy\Projects\Stuff4Others\SLR\YaquinaGIS\dist2mouth
## names       : dist2mouth
## values      : 0, 41878  (min, max)
```

Confirm that the number of rows, columns, extent, and coordinate system match the elevation model.

Plot the current *Zostera marina* layer:

```
plot(zm, legend=FALSE)
    plot(tideflat, border="gray", add=T)
    plot(zm, legend.only=T, legend.width=1, legend.mar=3,
         axis.args=list(at=c(0,1), labels=c(0,1), cex.axis=1),
         legend.args=list(text="Presence/Absence Zostera marina", side=4, cex=1,
                          line=-2.5, font=2))
          segments(1000000, 321000, 1005000, 321000, lwd=4)
          text(1002500, 320500, "5 Km")
```



Explore the current *Zostera marina* layer:

```
zm
```

```
## class       : RasterLayer
## dimensions  : 3993, 4974, 19861182  (nrow, ncol, ncell)
## resolution  : 4, 4  (x, y)
## extent      : 988465, 1008361, 318565, 334537  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=lcc +lat_1=43 +lat_2=45 +lat_0=41.75 +lon_0=-120 +x_0=1312335.958005249
  +y_0=0 +ellps= ## data source : C:\Users\leemccoy\Projects\Stuff4Others\SLR\YaquinaGIS\zmarinanow
## names       : zmarinanow
## values      : 0, 1  (min, max)
```

Confirm that the number of rows, columns, extent, and coordinate system match the elevation model.

Calculate New GIS Layers

Calculate distance to MHHW and MLLW

Two of the parameters used in this model are the distance from each cell to the mean higher high water level (MHHW) and the mean lower low water level (MLLW). MHHW and MLLW values can be determined for an estuary from the National Geodetic Service tidal benchmarks.

See the end of this script for complete tidal datum information

```
 MEAN HIGHER HIGH WATER                    MHHW   =  2.542
```

```
MEAN HIGH WATER                          MHW    =  2.330
MEAN TIDE LEVEL                          MTL    =  1.376
MEAN SEA LEVEL                           MSL    =  1.358
MEAN LOW WATER                           MLW    =  0.421
North American Vertical Datum            NAVD88 =  0.225
MEAN LOWER LOW WATER                     MLLW   =  0.000
```

These values are used to create a contour for MHHW and MLLW. Two new rasters are created that reflect the distance from each pixel to each of these contours. This does not have to be done in R, it can be done in ArcGIS or any other GIS program, but the resolution, extents, and coordinate system must match the other raster files.

Remember to confirm that these values are in the same units as the DEM (e.g., meters or feet).

DEM has been adjusted to be relative to NAVD88, so MLLW and MHHW values will need to be adjusted to match

The table above shows that MLLW is at 0.000. NAVD88 zero is at 0.225 relative to MLLW. So MLLW is -0.225 relative to NAVD88.

```
#set MLLW
  MLLW=-0.225
#create a contour shapefile line feature from DEM at MLLW
  MLLWcontour <- rasterToContour(dem, levels=MLLW, maxpixels=1000000)
```

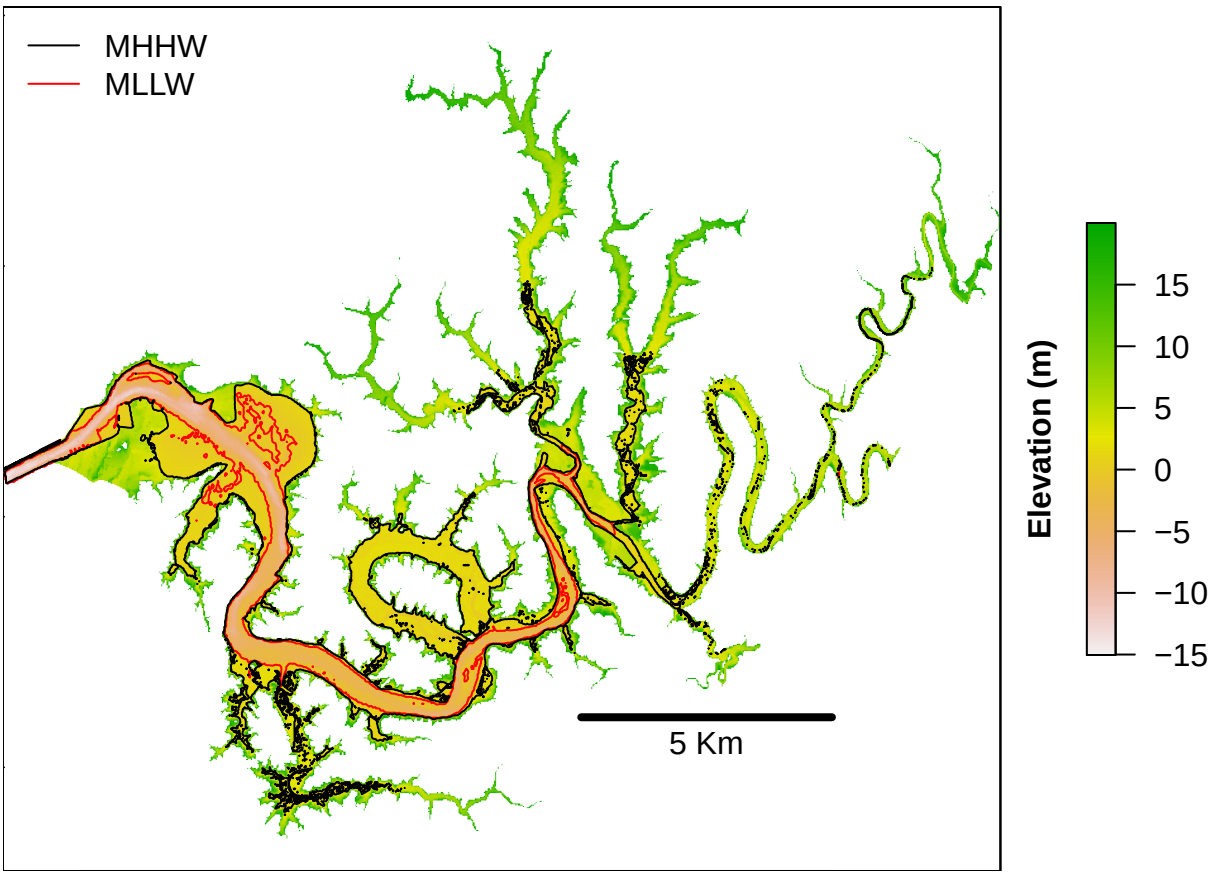
The table above shows that MHHW is at 2.542 relative to MLLW. Since NAVD88 zero is at 0.225 relative to MLLW, it is necessary to subtract 0.225 from 2.542 to get MHHW relative to NAVD88.

The DEM used here covers the area immediately surrounding the estuary up to ~30 meters NAVD88. Pixels outside of this area are blank (value=NA). The rasterToContour command used here wraps around this upper edge and tries to estimate the MHHW line on the outside of the area of interest. This doesn't fundamentally change the results since pixels in the intertidal will still be the same distance from the correct portions of the MHHW line, but it does make for an incorrect MHHW line. To correct this, a temprary DEM file is created where the NA's are replaced with a value far outside of the range of interest - 100 meters NAVD88 in this case.

```
#set MLLW
  MHHW=2.317
  #create the temporary dem with NA cells replaced with 100 meters NAVD88
    dem4mhhw <- calc(dem, fun=function(x) ifelse(is.na(x),100,x))
    MHHWcontour <- rasterToContour(dem4mhhw, levels=MHHW, maxpixels=1000000)
    #remove the temporary dem
      rm(dem4mhhw)
```

Plot these to make sure they look reasonable.

```
plot(dem, legend=F)
plot(MLLWcontour, col="red", add=T)
plot(MHHWcontour, col="black", add=T)
```



Create a raster of the MLLW contour with NA for all cells that do not touch the contour and the value of MLLW for all of the cells that do touch the countour.

To accomplish this, it is necessary to provide a raster layer with the proper extent, resolution, and projection. A blank raster can be produced from scratch, but it is easier to use one of the existing rasters as a template. Any of the rasters from above can be used (zm is used here).

Note that these processes can take a significant amount of time depending on the size of your estuary, the resolution of your rasters and your computer processing capability. For this example, it was taking ~15 minutes to create the distance to MLLW raster. A progress bar has been built in, but sometimes it takes a while for the progress bar to report as well. Please be patient.

```
#create a raster of MLLW contour
  rasterMLLWcontour <- rasterize(MLLWcontour, zm, field='level', fun='last',
                                 background=NA, filename="rasterMLLWcontour",
                                 overwrite=T, progress='text')
#create a raster of MHHW contour
  rasterMHHWcontour <- rasterize(MHHWcontour, zm, field='level', fun='last',
                                 background=NA, filename="rasterMHHWcontour",
                                 overwrite=T, progress='text')
```

NOTE: **rasterize()** in the previous section is a function. A number of arguments were provided to **rasterize** (including field, fun, background, etc.). To see a list of available arguments and what they do for any function, use **?function_name**. This will pop up a description of the function and define the arguments.

```
?rasterize
```

Use the distance function to calculate the distance from each NA cell in **rasterMLLWcontour** and **rasterMHHWcontour** to the nearest non-NA cell (the cells that represent the MLLW or MHHW contour line). This process takes the longest and only sporadically reports the progress.

```
  d2MLLW <- distance(rasterMLLWcontour, filename="YaquinaGIS\\d2MLLW", overwrite=T,
                     progress='text')
  d2MHHW <- distance(rasterMHHWcontour, filename="YaquinaGIS\\d2MHHW", overwrite=T,
                     progress='text')
```

Calculating the Square and Cube of Elevation

Two other important parameters are the square and cube of the elevation. It is possible to create these files on the fly and store them in R. However, these files are rather large and it will be faster to create them once and then access them when needed.

```
#create the square of DEM and save as YB_dem_square
  dem_square <- calc(dem, fun=function(x) {x ^ 2},
                     filename="YaquinaGIS\\YB_dem_square", progress='text')
#create the cube of DEM and save as YB_dem_cube
  dem_cube <- calc(dem, fun=function(x) {x ^ 3},
                   filename="YaquinaGIS\\YB_dem_cube", progress='text')
```

Calculating the Square of Distance to Mouth

Another factor which ends up being useful is the square of distance to mouth.

```
#create the square of dem and save as YB_dem_square
  d2m_square <- calc(d2m, fun=function(x) {x ^ 2},
                     filename="YaquinaGIS\\YB_d2mouth_square", progress='text')
```

Creating Random Extraction Points

The first step of determining which environmental variables are associated with the distribution of *Zostera marina* is to generate random sample points throughout the estuary. The number of points selected is not important as long as it is sufficient to adequately represent the ranges of our predictor and response variables. For this exercise, 1000 points are randomly distributed throughout Yaquina Bay, Oregon. The points are randomly distributed inside of the estuary polygon layer.

Note: A new set of points is created each time the following line of code is run, which means a whole new set of data will be created which will vary from the previously created data. Only create new sample points and extract the data from each of the layers if this is necessary. Otherwise load the previously extracted data.

```
  samplepoints <- spsample(estuary, 1000, type="random")
```

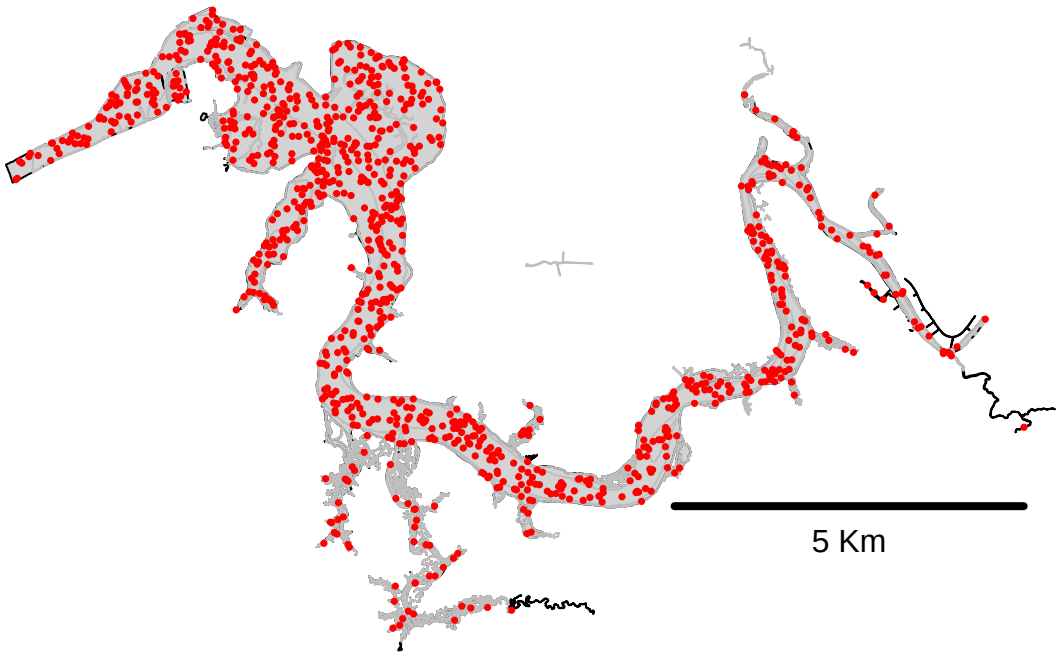
It is necessary to convert these spatial sample points into a file type that can hold data. This requires a spatial points data frame.

```
#merge the sample points with a dataframe holding the point coordinates
  samplepoints <- SpatialPointsDataFrame(samplepoints,
                                         as.data.frame(coordinates(samplepoints)))
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "YaquinaGIS", layer: "ExtractedValuesAtSamplePoints"
## with 1000 features and 10 fields
## Feature type: wkbPoint with 2 dimensions
```

Plot these points:

```
par(mar=c(0,0,0,3))
plot(estuary, col="lightgray")
    plot(tideflat, border="gray", add=T)
          segments(998000, 321000, 1003000, 321000, lwd=4)
          text(1000500, 320500, "5 Km")
points(samplepoints, col="red", pch=16, cex=0.5)
```



Extracting Data from Each Parameter Layer

The value of each predictor raster and the *Zostera marina* presence/absence raster is extracted at each of the sample points and added to the samplepoints dataframe.

```
#extract the elevation at each sample point
  samplepoints$dem <- extract(dem, samplepoints)
#extract the distance to the mouth at each sample point
  samplepoints$d2mouth <- extract(d2m, samplepoints)
#extract the squared distance to the mouth at each sample point
```

```
  samplepoints$d2mouth_sq <- extract(d2m_square, samplepoints)
#extract the distance to mean higher high water at each sample point
  samplepoints$d2mhhw <- extract(d2MHHW, samplepoints)
#extract the distance to mean lower low water at each sample point
  samplepoints$d2mllw <- extract(d2MLLW, samplepoints)
#extract the squared elevation at each sample point
  samplepoints$dem_square <- extract(dem_square, samplepoints)
#extract the cubed elevation at each sample point
  samplepoints$dem_cube <- extract(dem_cube, samplepoints)

#extract the current presence/absence of zostera marina
  samplepoints$zm <- extract(zm, samplepoints)
```

Save this file of samplepoints with the extracted data.

```
  writeOGR(samplepoints, dsn="YaquinaGIS", layer="ExtractedValuesAtSamplePoints",
           driver="ESRI Shapefile", overwrite=T)
```

Data Confirmation

Before proceeding, it is useful to confirm that all of the points overlap usable data for each of the factors. Using summary allows confirmation of the range of values and shows how many points have NA for each of the factors. Check that the range and mean make sense for each of the layers. If there are a large number of NA's, it is worth checking to make sure the layers are aligned and all layers have complete data for the areas of interest.

```
  summary(samplepoints)
```

```
## Object of class SpatialPointsDataFrame
## Coordinates:
##              min     max
## coords.x1 988569 1003004
## coords.x2 319254  328102
## Is projected: TRUE
## proj4string :
## [+proj=lcc +lat_1=43 +lat_2=45 +lat_0=41.75 +lon_0=-120
## +x_0=1312335.958005249 +y_0=0 +datum=NAD83 +units=m +no_defs
## +ellps=GRS80 +towgs84=0,0,0]
## Number of points: 1000
## Data attributes:
##        x                y              dem             d2mouth
##  Min.   : 988569   Min.   :319254   Min.   :-13.560   Min.   :  147
##  1st Qu.: 992495   1st Qu.:322531   1st Qu.: -3.436   1st Qu.: 4773
##  Median : 993758   Median :324947   Median : -0.426   Median : 6606
##  Mean   : 994471   Mean   :324558   Mean   : -1.595   Mean   : 8450
##  3rd Qu.: 995935   3rd Qu.:326502   3rd Qu.:  0.678   3rd Qu.:11685
##  Max.   :1003004   Max.   :328102   Max.   : 10.222   Max.   :22744
##                                     NA's   :1         NA's   :1
##    d2mouth_sq          d2mhhw          d2mllw       dem_square
##  Min.   :2.17e+04   Min.   :  0.0   Min.   :   0   Min.   :  0.00
##  1st Qu.:2.28e+07   1st Qu.: 45.3   1st Qu.:  36   1st Qu.:  0.39
```

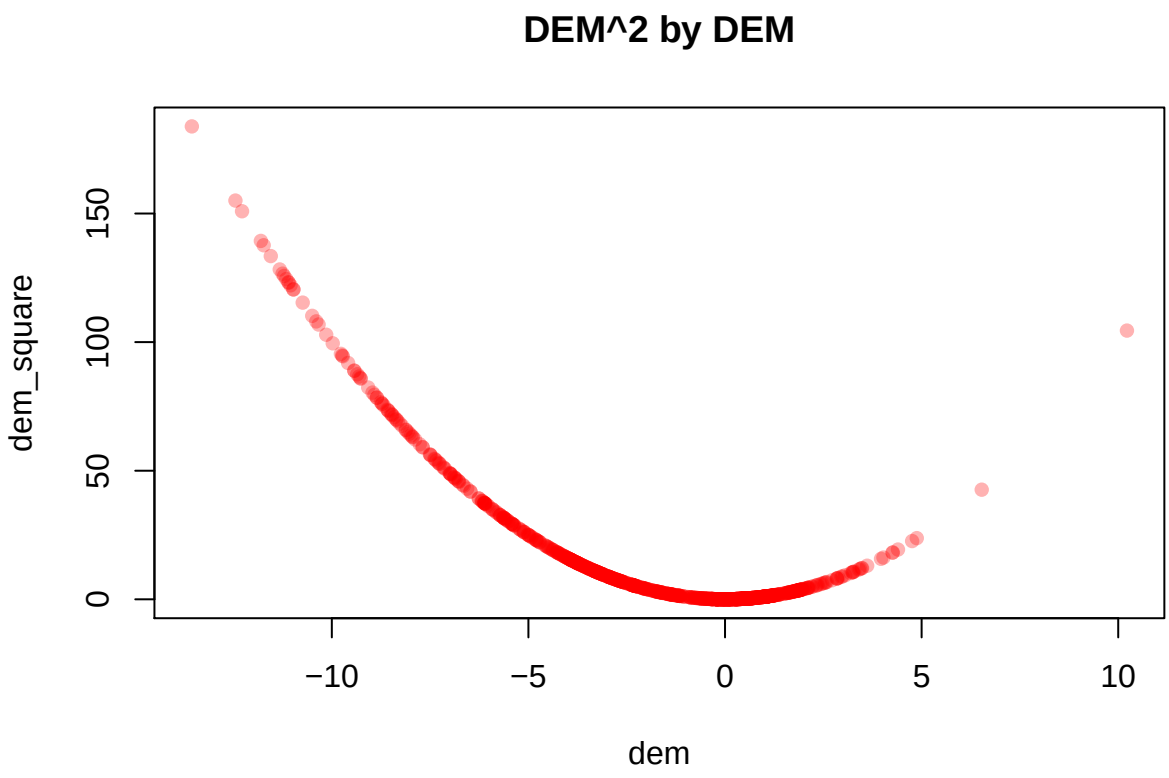
```
##  Median :4.36e+07   Median :113.0   Median :  87   Median :  2.74
##  Mean   :9.55e+07   Mean   :152.4   Mean   : 210   Mean   : 12.86
##  3rd Qu.:1.37e+08   3rd Qu.:208.2   3rd Qu.: 184   3rd Qu.: 12.53
##  Max.   :5.17e+08   Max.   :787.6   Max.   :3259   Max.   :183.88
##  NA's   :1                                         NA's   :1
##     dem_cube             zm
##  Min.   :-2493.4   Min.   :0.0000
##  1st Qu.:  -40.6   1st Qu.:0.0000
##  Median :   -0.1   Median :0.0000
##  Mean   :  -84.7   Mean   :0.0911
##  3rd Qu.:    0.3   3rd Qu.:0.0000
##  Max.   : 1068.1   Max.   :1.0000
##  NA's   :1         NA's   :1
```

After confirming that the number of NA values is reasonable, remove the points with NA values in any column.

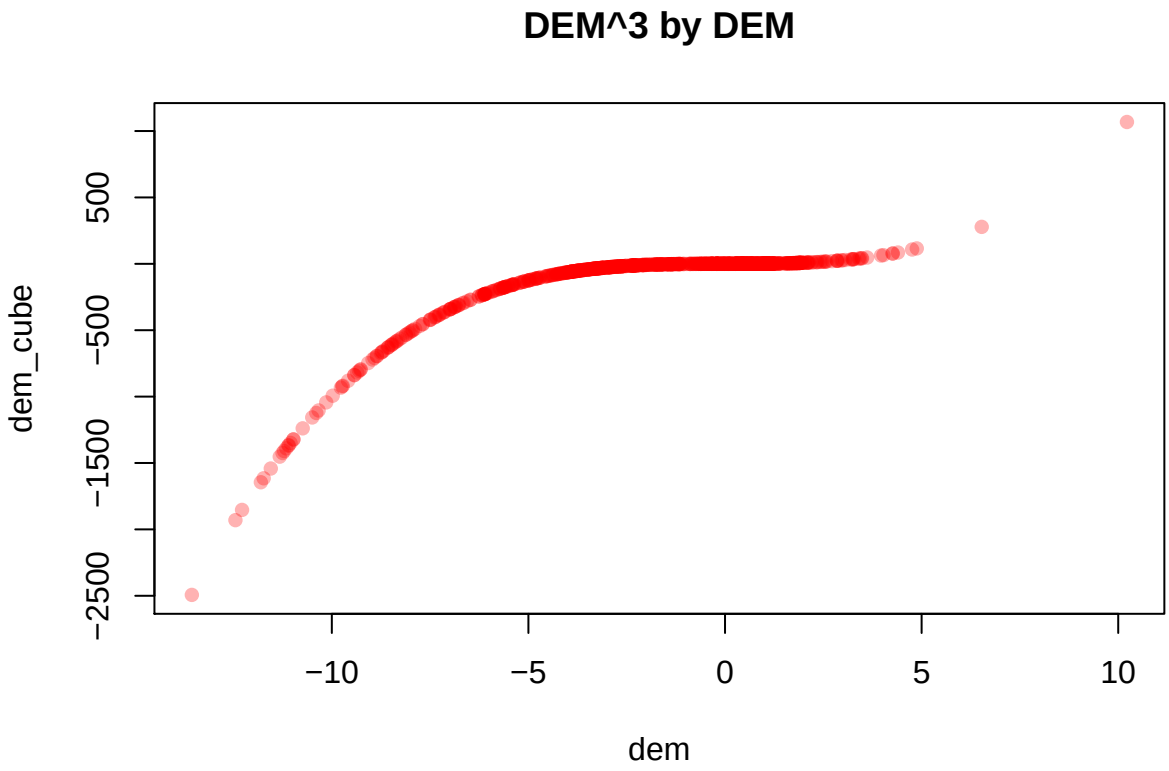
```
#removes any row that contains an NA (not complete)
  samplepoints <- samplepoints[complete.cases(samplepoints@data),]
```

It is also worthwhile to plot the derived values against their origins to confirm that they conform to expectation.

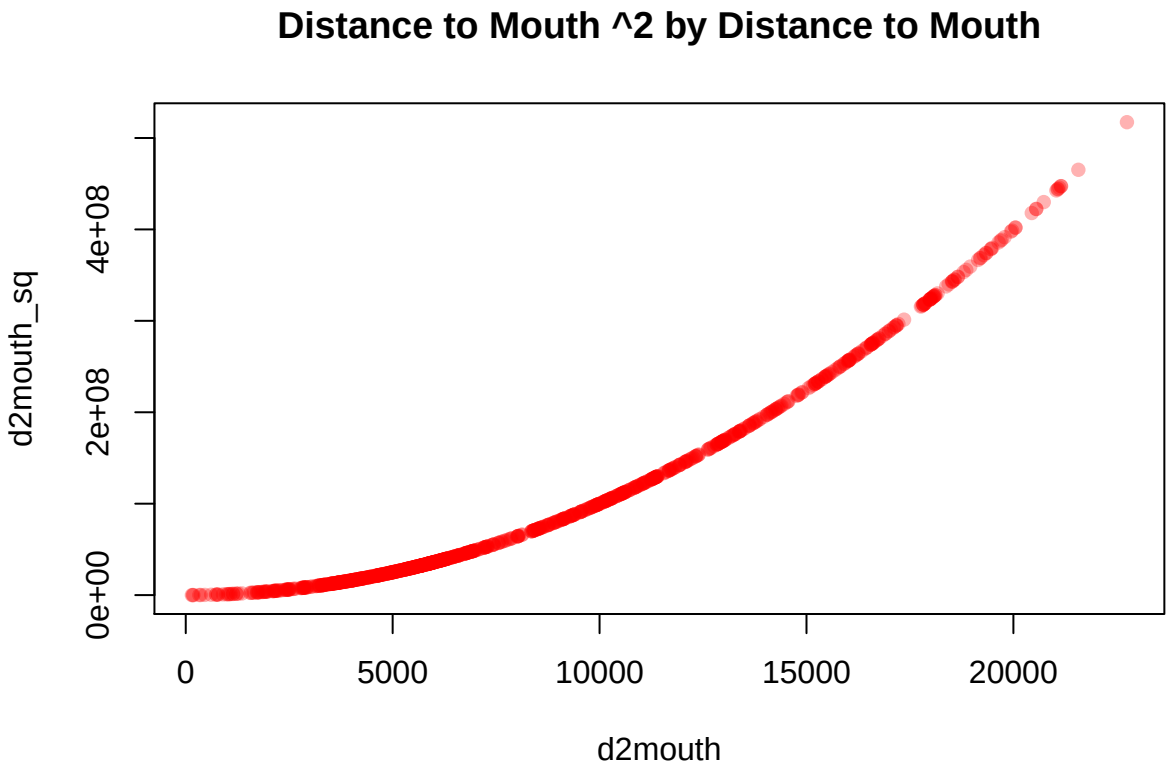
```
  plot(dem_square~dem, data=samplepoints, col=rgb(1,0,0,0.3), pch=16,
       main="DEM^2 by DEM")
```



```
plot(dem_cube~dem, data=samplepoints, col=rgb(1,0,0,0.3), pch=16,
     main="DEM^3 by DEM")
```



```
plot(d2mouth_sq~d2mouth, data=samplepoints, col=rgb(1,0,0,0.3), pch=16,
     main="Distance to Mouth ^2 by Distance to Mouth")
```



And it is useful to confirm that there are a reasonable number of sites where *Zostera marina* is present (the general guideline is at least 10 points with *Zostera marina* present for each predictor variable).

There are 908 sample points without *Zostera marina* and 91 sample points with *Zostera marina* present.

These numbers will change each time a new set of "sample points" are generated and the values extracted, so it is impossible to address specific examples. However, the main points are to confirm that the range and mean of values for each factor are reasonable and confirm that a reasonable number of points have valid data for all of the extracted parameters.

Model Creation

A binomial generalized linear model with a logit link is used because these models predict presence/absence data and return easily interpreted coefficients that can be integrated into SLAMM.

One of the challenges of developing a predictive model is selecting the parameters that accurately predict the response variable, without over fitting the data. The goal here is to develop the simplest model that does the best job of predicting the distribution of *Zostera marina* without overparameterizing the model. The following code is used to compare the performance of four potential models, comprised of different combinations of parameters.

There are two common errors/warnings that are reported with these models:

1. Warning: glm.fit: algorithm did not converge

2. Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

There are several excellent descriptions online of what these errors mean and why they occur in logistic regression.

In this case, the first error is probably due to having too many parameters relative to the number of samples that actually have *Zostera marina*, and/or fairly high multicollinearity between parameters. You should definitely read the documentation if this error occurs, but in summary, this error occurs because the prediction gets so close to 0 or 1 that the number becomes in effect 0 or 1, preventing the algorithm from converging. One way to deal with this is to increase the number of iterations within the glm function so the algorithm can converge (see the documentation for glm). The number of parameters included in the model can also be reduced (which will be done in mod2).

The second error is due to how *Zostera marina* is distributed relative to the predictor variables and is to be expected in this case. See Example A below.

Due to the relatively large sample size of 1000 random points, many parameters will have significant p-values (<0.05) even though they may not actually improve the model fit. As the sample size increases, very small and biologically insignificant differences in values will be significant. Predictor variables that are not significant, however, are probably not useful predictors. The point is, don't get too excited about the p-values reported in the summary. They are basically meaningless and are only useful for determining which variables to remove.

Model 1

Create a model with all the available parameters.

```
#Logistic regression model using all available values
  mod1 <- glm(zm ~  dem + dem_square +dem_cube + d2mllw + d2mhhw + d2mouth +
              d2mouth_sq, data=samplepoints, family=binomial(link=logit))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(mod1)
```

```
##
## Call:
## glm(formula = zm ~ dem + dem_square + dem_cube + d2mllw + d2mhhw +
##     d2mouth + d2mouth_sq, family = binomial(link = logit), data = samplepoints)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8593  -0.0833  -0.0013   0.0000   2.8105
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.53e+00   1.27e+00   -2.77  0.00557 **
## dem         -1.57e+00   4.75e-01   -3.30  0.00098 ***
## dem_square  -1.12e+00   4.67e-01   -2.40  0.01655 *
## dem_cube     1.18e-01   2.25e-01    0.52  0.60078
## d2mllw      -1.26e-02   3.63e-03   -3.48  0.00049 ***
## d2mhhw       4.20e-03   1.03e-03    4.08  4.6e-05 ***
## d2mouth      1.05e-03   4.02e-04    2.61  0.00905 **
## d2mouth_sq  -8.88e-08   3.05e-08   -2.91  0.00363 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.50  on 998  degrees of freedom
## Residual deviance: 240.19  on 991  degrees of freedom
## AIC: 256.2
##
## Number of Fisher Scoring iterations: 16
```

Model 2

In model 2, dem_cube is removed since it is not significant in mod1.

```
  mod2 <- glm(zm ~  dem + dem_square + d2mllw + d2mhhw + d2mouth +
              d2mouth_sq, data=samplepoints, family=binomial(link=logit))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
  summary(mod2)
```

```
##
## Call:
## glm(formula = zm ~ dem + dem_square + d2mllw + d2mhhw + d2mouth +
##     d2mouth_sq, family = binomial(link = logit), data = samplepoints)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8612  -0.0825  -0.0017   0.0000   2.8254
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.47e+00   1.27e+00   -2.73  0.00627 **
## dem         -1.54e+00   4.79e-01   -3.22  0.00129 **
## dem_square  -1.31e+00   3.40e-01   -3.86  0.00011 ***
## d2mllw      -1.26e-02   3.64e-03   -3.45  0.00056 ***
## d2mhhw       4.15e-03   1.03e-03    4.05  5.2e-05 ***
## d2mouth      1.05e-03   4.03e-04    2.59  0.00949 **
## d2mouth_sq  -8.86e-08   3.06e-08   -2.89  0.00383 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.50  on 998  degrees of freedom
## Residual deviance: 240.31  on 992  degrees of freedom
## AIC: 254.3
##
## Number of Fisher Scoring iterations: 12
```

Model 3

In model 3, the square of distance to mouth is removed for comparison.

```
  mod3 <- glm(zm ~  dem + dem_square + d2mllw + d2mhhw + d2mouth,
              data=samplepoints, family=binomial(link=logit))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
  summary(mod3)
```

```
##
## Call:
## glm(formula = zm ~ dem + dem_square + d2mllw + d2mhhw + d2mouth,
##     family = binomial(link = logit), data = samplepoints)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0703  -0.1187  -0.0033   0.0000   3.1114
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.22e-01   5.03e-01    0.64  0.52248
## dem         -1.79e+00   4.94e-01   -3.62  0.00029 ***
## dem_square  -1.59e+00   3.73e-01   -4.25  2.1e-05 ***
## d2mllw      -1.22e-02   3.60e-03   -3.40  0.00069 ***
## d2mhhw       5.11e-03   1.01e-03    5.07  3.9e-07 ***
## d2mouth     -2.41e-04   5.94e-05   -4.06  4.9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.50  on 998  degrees of freedom
## Residual deviance: 256.31  on 993  degrees of freedom
## AIC: 268.3
##
## Number of Fisher Scoring iterations: 12
```

Model 4

In model 4, a very simple model with only elevation is used for comparison.

```
  mod4 <- glm(zm ~  dem, data=samplepoints, family=binomial(link=logit))
```

```
  summary(mod4)
```

```
##
## Call:
## glm(formula = zm ~ dem, family = binomial(link = logit), data = samplepoints)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.038  -0.500  -0.411  -0.321   2.244
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.1343     0.1123  -19.01  < 2e-16 ***
## dem           0.1757     0.0436    4.03  5.6e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.50  on 998  degrees of freedom
## Residual deviance: 590.21  on 997  degrees of freedom
## AIC: 594.2
##
## Number of Fisher Scoring iterations: 5
```

Model Comparison

When building models, it is often a delicate balance between including all of the relevant predictors without making an unnecessarily complicated model that includes predictors that statistically benefit the model but do not meaningfully contribute to describing the pattern of the dependent variable. The most accurate way to test a model and determine if it is "overfit" is with cross validation, which will be covered further down. However, a quick way to compare models is to use Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC). Complete definitions can be found online, but the basic idea is to compare models by quantifying how well they do at predicting while penalizing for each parameter included. When the penalty is greater than the additional benefit from a parameter, one would select the simpler model. AIC and BIC attempt to provide this simple comparison. AIC and BIC differ in that BIC has a higher penalty for additional parameters and is therefore more conservative. When comparing the AIC and BIC score of different models, the model with the lower AIC or BIC score is preferred. Since model exploration can often include numerous parameters and interactions, having a quick way to compare models is useful for determining the two or three best models to further compare with cross validation.

AIC

```
AIC(mod1,mod2,mod3, mod4)
```

```
##      df   AIC
## mod1  8 256.2
## mod2  7 254.3
## mod3  6 268.3
## mod4  2 594.2
```

For Yaquina Bay, Oregon, mod1 and mod2 have much lower AIC scores than mod3 and mod4 and will be the preferred models. mod2 has a lower AIC score than mod1, but the difference is relatively small.

BIC

```
BIC(mod1,mod2,mod3, mod4)
```

```
##      df   BIC
## mod1  8 295.4
## mod2  7 288.7
## mod3  6 297.7
## mod4  2 604.0
```

For Yaquina Bay, Oregon, mod2 has a much lower BIC score than either mod1, mod3, or mod4. Using BIC to select would result in mod2 being selected as the preferred model.

The AIC and BIC comparison between models identifies mod2 is the best performing model for the Yaquina estuary.

Model Validation

One of the easiest ways to confirm the performance of a model is to use cross validation. A small portion of the data is held in reserve (~10-20%) and the parameter estimates are obtained using the remaining data (~80-90%). The model is then applied to the reserve data and the predicted values are compared to the observed values. This is repeated numerous times with different portions of the data being held in reserve to obtain a better estimate of model performance. Cross validation is used to evaluate the parameters included in the model, but does not evaluate the coefficients values.

There are several pre-written functions for cross validating data. What follows is a description of how to manually perform cross validation.

Start by adding a column defining sub portions of the data. Here each row of data is assigned to one of ten groups.

```
#add a column assigning each row to one of ten groups
rowid <- sample(seq(1,dim(samplepoints)[1],1), dim(samplepoints)[1], replace=F)
group <- rep(seq(1,10,1),100)

#setup a column to hold the sub-group
  samplepoints$sub <- NA
#assign the group value to samplepoints$sub at each rowid
  for(i in 1:length(rowid)){
    samplepoints$sub[rowid[i]] <- group[i]
  }
#see how many lines were assigned to each group
  table(samplepoints$sub)
```

```
##
##   1   2   3   4   5   6   7   8   9  10
## 100 100 100 100 100 100 100 100 100  99
```

Set aside each sub-group of data, create a model using the remaining data, predict the values for the sub-group using this new model, and compare the observed to predicted values of the sub-group. All of the information from each sub-group is stored in a dataframe to show how well each validation run does.

To accomplish this for different models it is easier to create a function that takes the model being tested and performs the cross validation. This function does not do anything until it is called further down the script.

```
#create a function that takes a model and calculates a 10 fold validation
  modelvalidate <- function(modelname){
    #create a dataframe to hold the data
      validationvalues <- data.frame(
        subgroup=c(1:10),
        numpoints=NA,
        NoZm=NA,
        FalsePos=NA,
        FalseNeg=NA,
```

```
        YesZm=NA
      )

  #for each subgroup
  for(i in 1:10){
    #temporarily assign one of the subsections of data to reserve
    reserve <- samplepoints@data[samplepoints$sub == i,]
  #temporarily assign the other subsections to tempdata
    tempdata <- samplepoints@data[samplepoints$sub != i,]
  #create a model using tempdata
    tempmodel <- glm(formula(modelname), data=tempdata,
                     family=binomial(link=logit), na.action='na.omit')
  #using tempmodel, predict the zm level for the data held in reserve
  #note that this produces probabilities between 0 and 1
    reserve$predict <- round(predict(tempmodel, newdata=reserve,
                                     type="response"),0)
  #put the values into validationvalues data frame - predict across the top of the table
    validationvalues$numpoints[i] <- dim(reserve)[1]
    validationvalues$NoZm[i] <- sum(reserve$zm == 0 & reserve$predict== 0)
    validationvalues$FalsePos[i] <- sum(reserve$zm == 0 & reserve$predict== 1)
    validationvalues$FalseNeg[i] <- sum(reserve$zm == 1 & reserve$predict== 0)
    validationvalues$YesZm[i] <- sum(reserve$zm == 1 & reserve$predict== 1)
  }

#calculate the validation values error and correct
  validationvalues$error <- round((validationvalues$FalsePos
                                   +validationvalues$FalseNeg)
                                  /validationvalues$numpoints,3)
  validationvalues$correct <- round((validationvalues$YesZm
                                     +validationvalues$NoZm)
                                    /validationvalues$numpoints,3)

  #have the function return validationvalues data frame
    return(validationvalues)
}
```

Call this function for each model that needs to be compared and look at the results.

Model 1: Full model

```
#cross validate model 1
  mod1val <- modelvalidate(mod1)
#look at the validation results of model 1
  mod1val
```

```
##    subgroup numpoints NoZm FalsePos FalseNeg YesZm error correct
## 1         1       100   92        0        3     5 0.030   0.970
## 2         2       100   88        5        2     5 0.070   0.930
## 3         3       100   86        4        0    10 0.040   0.960
## 4         4       100   83        3        8     6 0.110   0.890
## 5         5       100   89        1        3     7 0.040   0.960
## 6         6       100   88        5        4     3 0.090   0.910
## 7         7       100   90        0       10     0 0.100   0.900
## 8         8       100   88        2        7     3 0.090   0.910
```

```
## 9         9       100   88        4        2     6 0.060   0.940
## 10       10        99   89        3        3     4 0.061   0.939
```

Model 2: Full model with parameter demcube removed

```
#cross validate model 2
  mod2val <- modelvalidate(mod2)
#look at the validation results of model 1
  mod2val
```

```
##    subgroup numpoints NoZm FalsePos FalseNeg YesZm error correct
## 1         1       100   92        0        4     4 0.040   0.960
## 2         2       100   88        5        2     5 0.070   0.930
## 3         3       100   88        2        5     5 0.070   0.930
## 4         4       100   83        3        8     6 0.110   0.890
## 5         5       100   89        1        3     7 0.040   0.960
## 6         6       100   88        5        4     3 0.090   0.910
## 7         7       100   88        2        3     7 0.050   0.950
## 8         8       100   86        4        3     7 0.070   0.930
## 9         9       100   89        3        2     6 0.050   0.950
## 10       10        99   89        3        3     4 0.061   0.939
```

Model 4: Simple model with only elevation

```
#cross validate model 4
  mod4val <- modelvalidate(mod4)
#look at the validation results of model 1
  mod4val
```

```
##    subgroup numpoints NoZm FalsePos FalseNeg YesZm error correct
## 1         1       100   92        0        8     0 0.080   0.920
## 2         2       100   93        0        7     0 0.070   0.930
## 3         3       100   90        0       10     0 0.100   0.900
## 4         4       100   86        0       14     0 0.140   0.860
## 5         5       100   90        0       10     0 0.100   0.900
## 6         6       100   93        0        7     0 0.070   0.930
## 7         7       100   90        0       10     0 0.100   0.900
## 8         8       100   90        0       10     0 0.100   0.900
## 9         9       100   92        0        8     0 0.080   0.920
## 10       10        99   92        0        7     0 0.071   0.929
```

Calculate the mean and standard deviation of the number of points correctly and incorrectly identified in each model.

```
#calculate the mean and sd of the number of points correctly predicted
  mod1_correct_mean <- mean(mod1val$correct)
  mod1_correct_sd <- sd(mod1val$correct)
#calculate the mean and sd for the number of points incorrectly predicted
  mod1_error_mean <- mean(mod1val$error)
  mod1_error_sd <- sd(mod1val$error)
#calculate the mean and sd of the number of points correctly predicted
  mod2_correct_mean <- mean(mod2val$correct)
```

```
  mod2_correct_sd <- sd(mod2val$correct)
#calculate the mean and sd for the number of points incorrectly predicted
  mod2_error_mean <- mean(mod2val$error)
  mod2_error_sd <- sd(mod2val$error)
#calculate the mean and sd of the number of points correctly predicted
  mod4_correct_mean <- mean(mod4val$correct)
  mod4_correct_sd <- sd(mod4val$correct)
#calculate the mean and sd for the number of points incorrectly predicted
  mod4_error_mean <- mean(mod4val$error)
  mod4_error_sd <- sd(mod4val$error)

data.frame(model = c(1,2,4),
           meancorrect = c(mod1_correct_mean,
                           mod2_correct_mean,
                           mod4_correct_mean)*100,
           sdcorrect = round(c(mod1_correct_sd,
                               mod2_correct_sd,
                               mod4_correct_sd)*100,2),
           meanerror = c(mod1_error_mean,
                         mod2_error_mean,
                         mod4_error_mean)*100,
           sderror = round(c(mod1_error_sd,
                             mod2_error_sd,
                             mod4_error_sd)*100,2))
```

```
##   model meancorrect sdcorrect meanerror sderror
## 1     1       93.09      2.76      6.91    2.76
## 2     2       93.49      2.22      6.51    2.22
## 3     4       90.89      2.17      9.11    2.17
```

One of the interesting things is how well mod4 (using only elevation as the predictor) performs as a model. mod4 gets 91% of the points correct and only 9% wrong. mod4val shows that this is accomplished by never predicting any point to have *Zostera marina*. On the other hand, the other two models contain false positives, indicating that they predict the presence of *Zostera marina* at some locations where it does not occur. Overall though, model 2 is better at predicting *Zostera marina* in the relatively small areas where it does occurs.

To understand how well model 2 is doing, it is useful to know how much *Zostera marina* is currently present. There are two ways to do this:

1) count the total number of pixels in the zm layer with and without zm,
2) count the number of sample points with and without zm.

The total area of the zm layer is much greater than the area from which the sample points were selected, so these can be very different values. This can be fixed by masking the zm layer by the estuary polygon.

```
#get the percent cover of zm currently in the zm layer
  ybzm <- as.data.frame(freq(zm))
  ybzmperc <- ybzm$count[ybzm$value %in% 1]
              /(ybzm$count[ybzm$value %in% 0]
              +ybzm$count[ybzm$value %in% 0])

#mask the zm layer
  zmmask <- mask(zm, estuary, filename="YaquinaGIS\\ZmMaskByBay")
```

```
#get the percent cover of zm in the masked layer
  ybzmmask <- as.data.frame(freq(zmmask))
  ybzmmaskperc <- ybzmmask$count[ybzmmask$value %in% 1]
                  /(ybzmmask$count[ybzmmask$value %in% 0]
                  +ybzmmask$count[ybzmmask$value %in% 1])

#get the percent cover of zm based on the sample points
  sum(samplepoints$zm, na.rm=T)/length(samplepoints$zm)
```

- *Zostera marina* makes up 1.5472 percent of the original zm layer.
- *Zostera marina* makes up 9.6955 percent of the zm layer masked to the sample area.
- *Zostera marina* makes up 9.1091 percent of the 1000 sample points.

Conclusion

Of the models we explored, **Model 2** performs the best in describing the distribution of *Zostera marina* in Yaquina Bay, Oregon. The coefficients from the summary of **Model 2** would be used in SLAMM to allow prediction of the distribution of *Zostera marina* with sea level rise.

```
  summary(mod2)
```

```
##
## Call:
## glm(formula = zm ~ dem + dem_square + d2mllw + d2mhhw + d2mouth +
##     d2mouth_sq, family = binomial(link = logit), data = samplepoints)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8612  -0.0825  -0.0017   0.0000   2.8254
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.47e+00   1.27e+00   -2.73  0.00627 **
## dem         -1.54e+00   4.79e-01   -3.22  0.00129 **
## dem_square  -1.31e+00   3.40e-01   -3.86  0.00011 ***
## d2mllw      -1.26e-02   3.64e-03   -3.45  0.00056 ***
## d2mhhw       4.15e-03   1.03e-03    4.05  5.2e-05 ***
## d2mouth      1.05e-03   4.03e-04    2.59  0.00949 **
## d2mouth_sq  -8.86e-08   3.06e-08   -2.89  0.00383 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.50  on 998  degrees of freedom
## Residual deviance: 240.31  on 992  degrees of freedom
## AIC: 254.3
##
## Number of Fisher Scoring iterations: 12
```

In this case, the values to use in SLAMM would be as follows:

- Intercept = -3.4748
- DEM = -1.5405
- DEM Squared = -1.3139
- Dem Cubed = 0
- Distance to MLLW = -0.0126
- Distance to MHHW = 0.0042
- Distance to Mouth = 0.001
- Distance to Mouth Squared = -8.8573 × 10-8

Predicting *Zostera marina* using Model 2

Create a stack of the raster layers for predicting the distribution of *Zostera marina*.

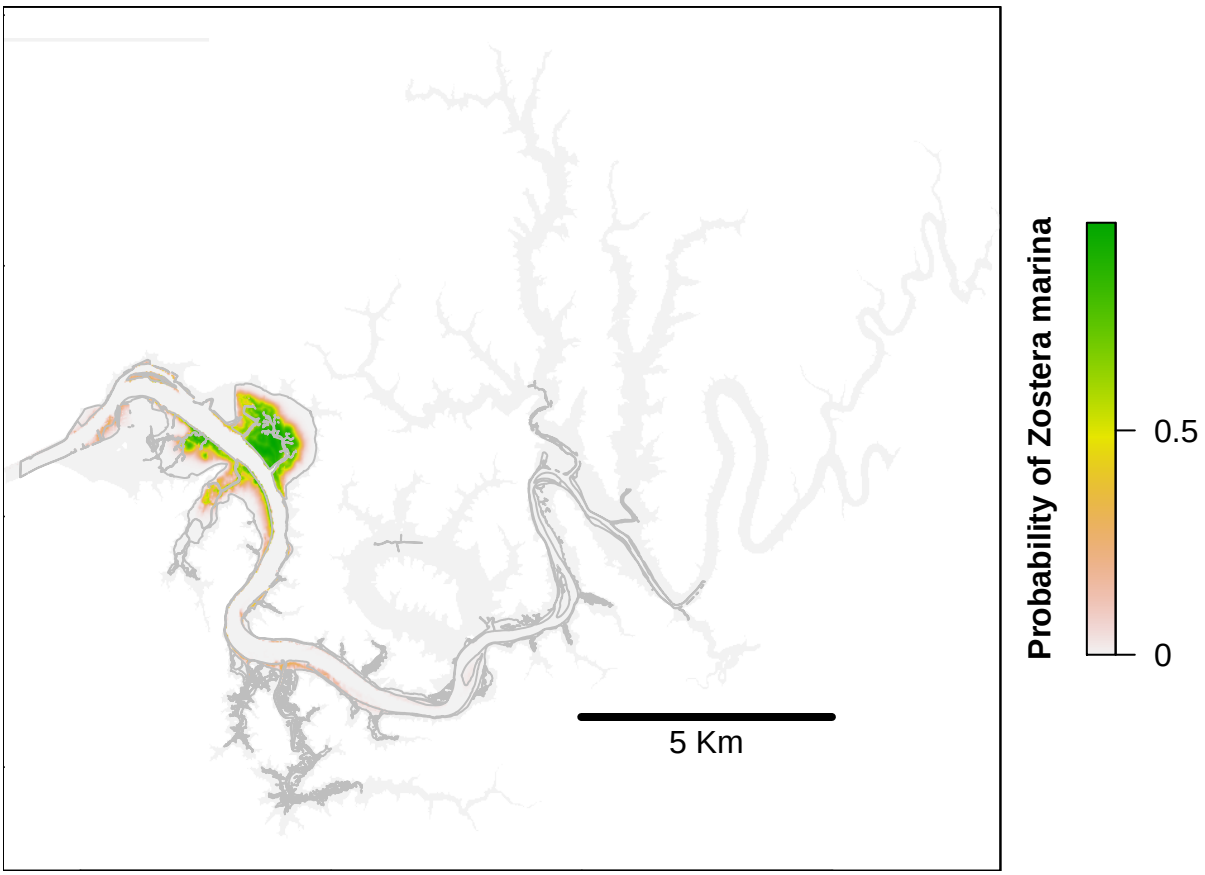
```
  temp <- stack(dem, dem_square, d2m, d2m_square, d2MLLW, d2MHHW)

#need to change the layer names to match
  names(temp) <- c("dem",
                   "dem_square",
                   "d2mouth",
                   "d2mouth_sq",
                   "d2mllw",
                   "d2mhhw")
  names(temp)

#predict zm levels
  prediction <- predict(temp, model=mod2,
                        filename="YaquinaGIS\\CurrentZostera_Prediction_Model2",
                        type="response", format="raster", overwrite=T)
```

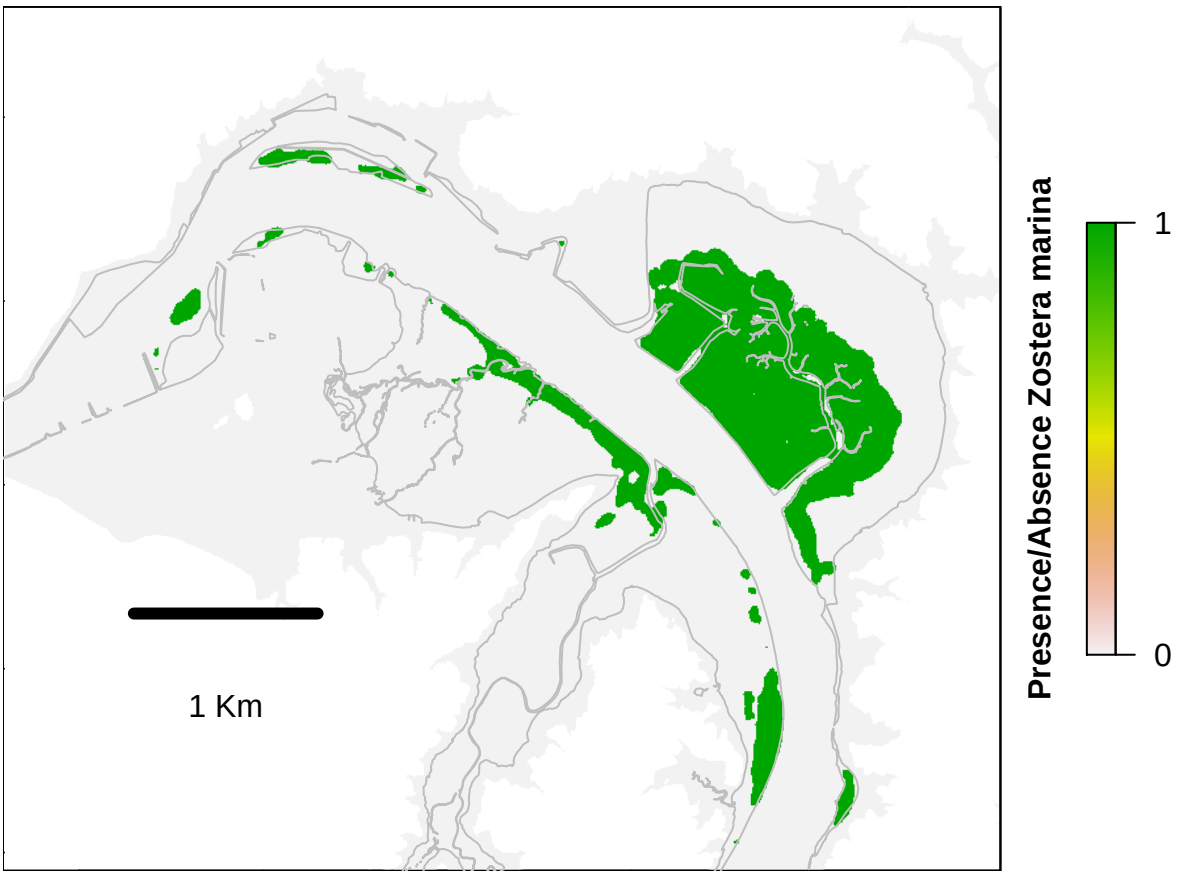
Plot the predicted *Zostera marina* probabilities.

```
par(mar=c(0,0,0,3))
plot(prediction, legend=FALSE)
  plot(tideflat, border="gray", add=T)
  plot(prediction, legend.only=T, legend.width=1, legend.mar=3,
       axis.args=list(at=c(0,0.5,1), labels=c(0,0.5,1), cex.axis=1),
       legend.args=list(text="Probability of Zostera marina", side=4, cex=1,
                        line=-2.5, font=2))
        segments(1000000, 321000, 1005000, 321000, lwd=4)
        text(1002500, 320500, "5 Km")
```



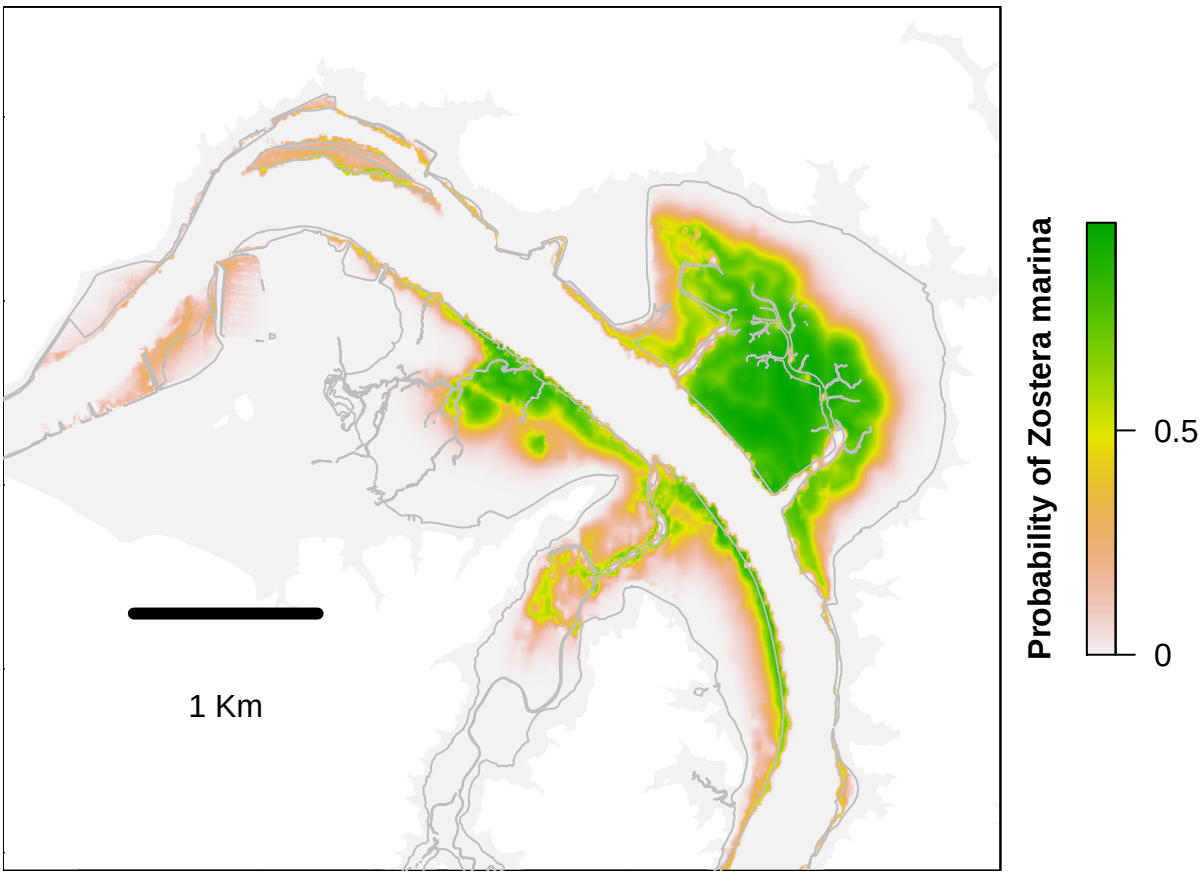
Plot the actual amounts of *Zostera marina*.

```
par(mar=c(0,0,0,3))
plot(zm, xlim=c(989600, 995000), ylim=c(323900, 328600), legend=F)
  plot(tideflat, border="gray", add=T)
  plot(zm, legend.only=T, legend.width=1, legend.mar=3,
       axis.args=list(at=c(0,1), labels=c(0,1), cex.axis=1),
       legend.args=list(text="Presence/Absence Zostera marina", side=4, cex=1,
                        line=-2.5, font=2))
        segments(990300, 325300, 991300, 325300, lwd=6)
        text(990800, 324800, "1 Km")
```



Plot the predicted amounts of *Zostera marina*.

```
par(mar=c(0,0,0,3))
plot(prediction, xlim=c(989600, 995000), ylim=c(323900, 328600), legend=F)
  plot(tideflat, border="gray", add=T)
  plot(prediction, legend.only=T, legend.width=1, legend.mar=3,
       axis.args=list(at=c(0,0.5,1), labels=c(0,0.5,1), cex.axis=1),
       legend.args=list(text="Probability of Zostera marina", side=4, cex=1,
                        line=-2.5, font=2))
        segments(990300, 325300, 991300, 325300, lwd=6)
        text(990800, 324800, "1 Km")
```



There are a number of ways to calculate the relative amount of the estuary occupied by *Zostera marina*. Further up, it is calculated that there are 1,014,639 4X4 meter pixels in the estuary. Instead of recalculating this, I am going to use that value here.

Calculate the actual and predicted percentage of the estuary with *Zostera marina*.

```
totaltf <- calc(zm, fun=function(x) ifelse(is.na(x),0,1))
zmactualperc <- (cellStats(zm, sum)/1014639)*100
zmpredictperc <- (cellStats(prediction, sum)/1014639)*100
```

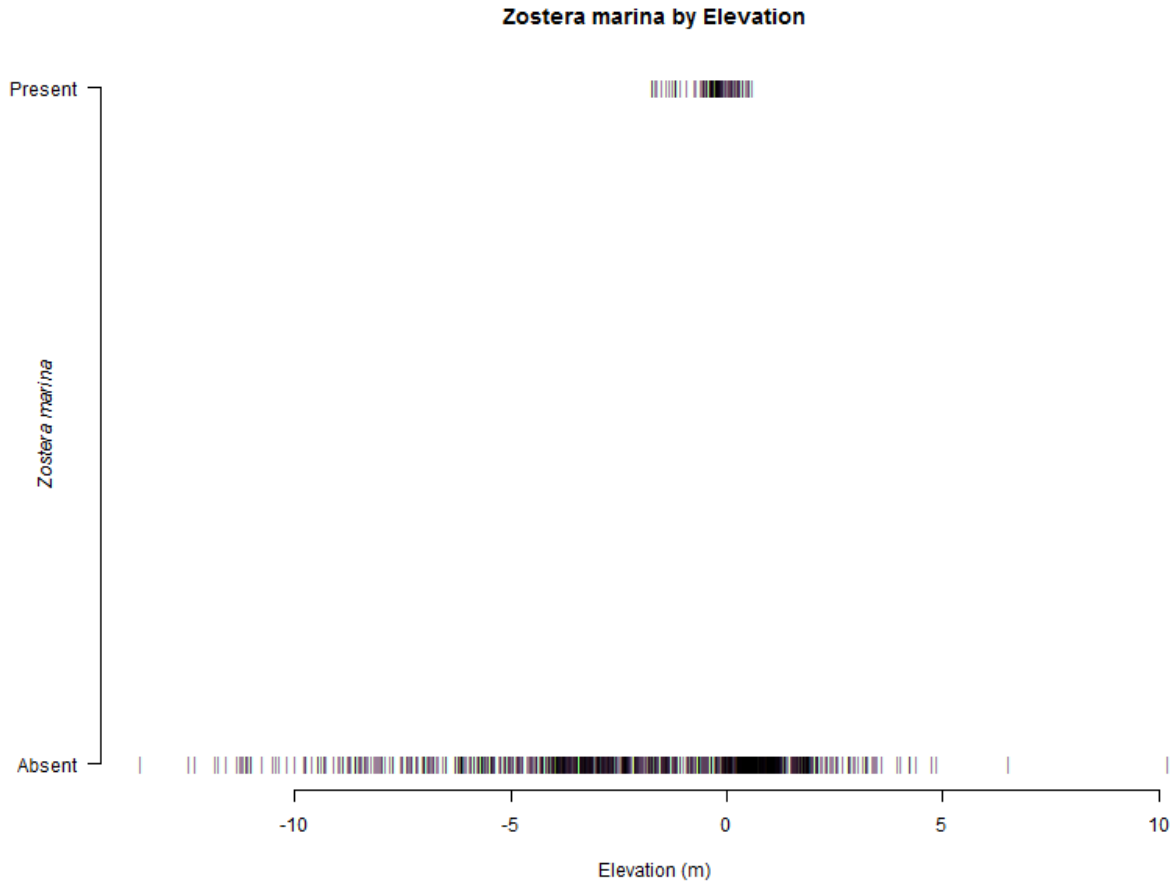
9.7157 percent of the estuary is actually occupied by *Zostera marina*.

10.4503 percent of the estuary is predicted to be occupied by *Zostera marina*.

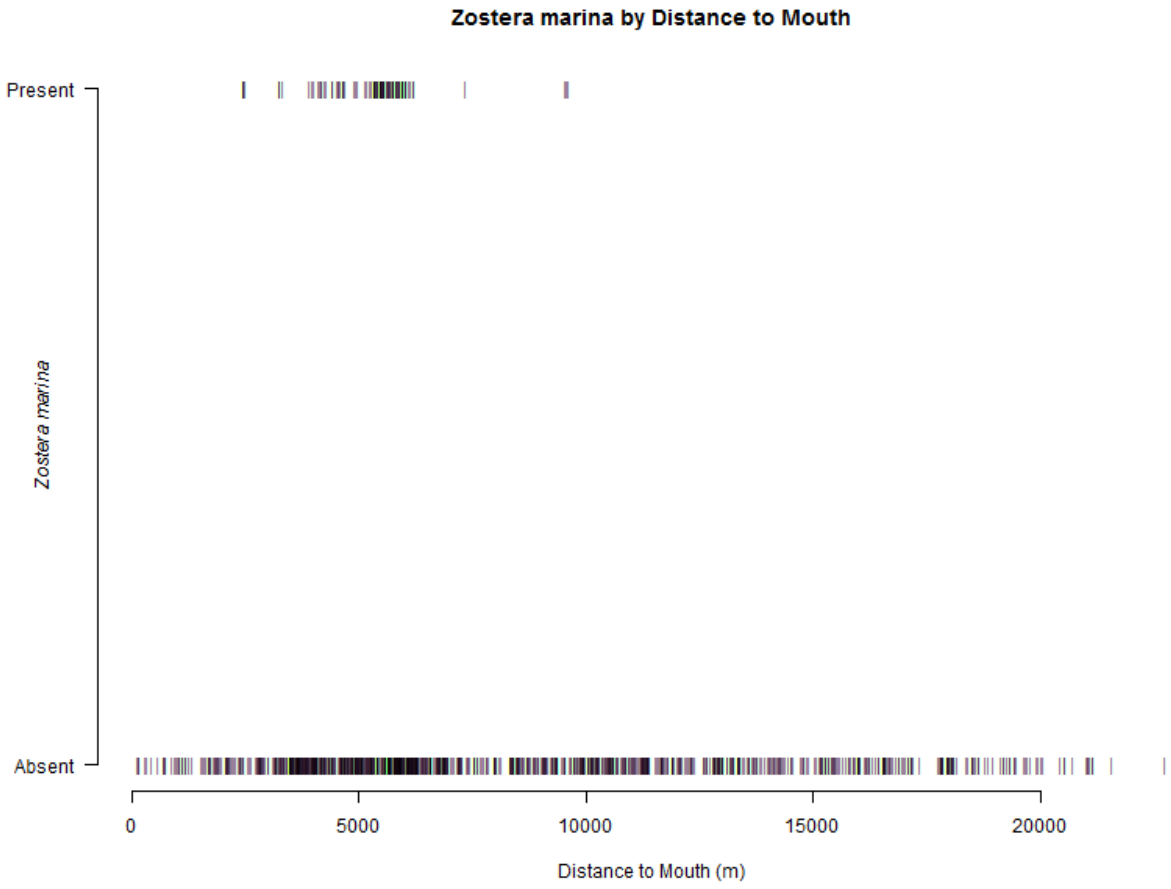
Example A

Zostera marina is not distributed randomly relative to elevation, distance to mouth, or distance to MLLW. This will lead to the warning glm.fit: fitted probabilities 0 or 1 occurred.

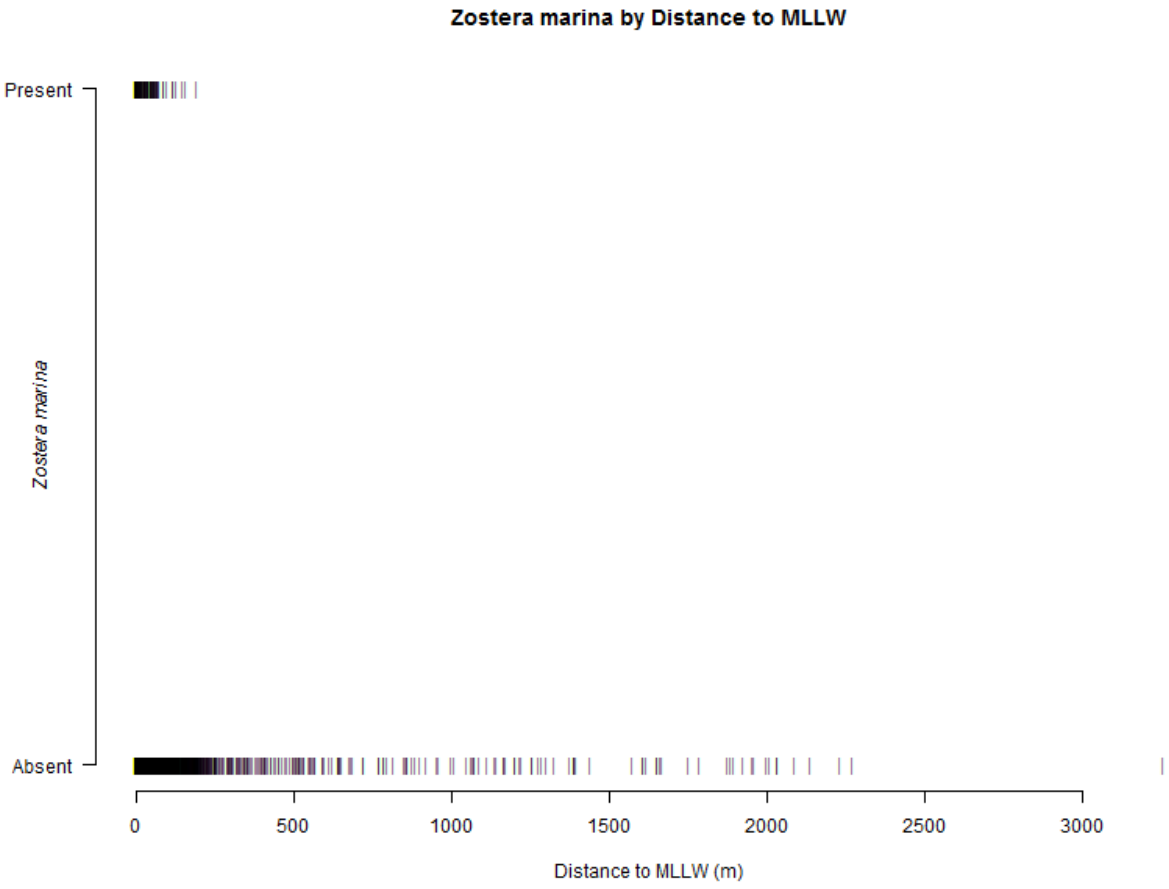
```
par(mar=c(5,6,4,2))
  plot(zm~dem, data=samplepoints, pch="|", xlab="Elevation (m)",
     ylab="", yaxt="n", frame=F, col=rgb(0,0,0,0.5),
     main="Zostera marina by Elevation")
axis(2,at=c(0,1), labels=c("Absent","Present"), las=2)
mtext("Zostera marina", 2, 2, font=3)
```



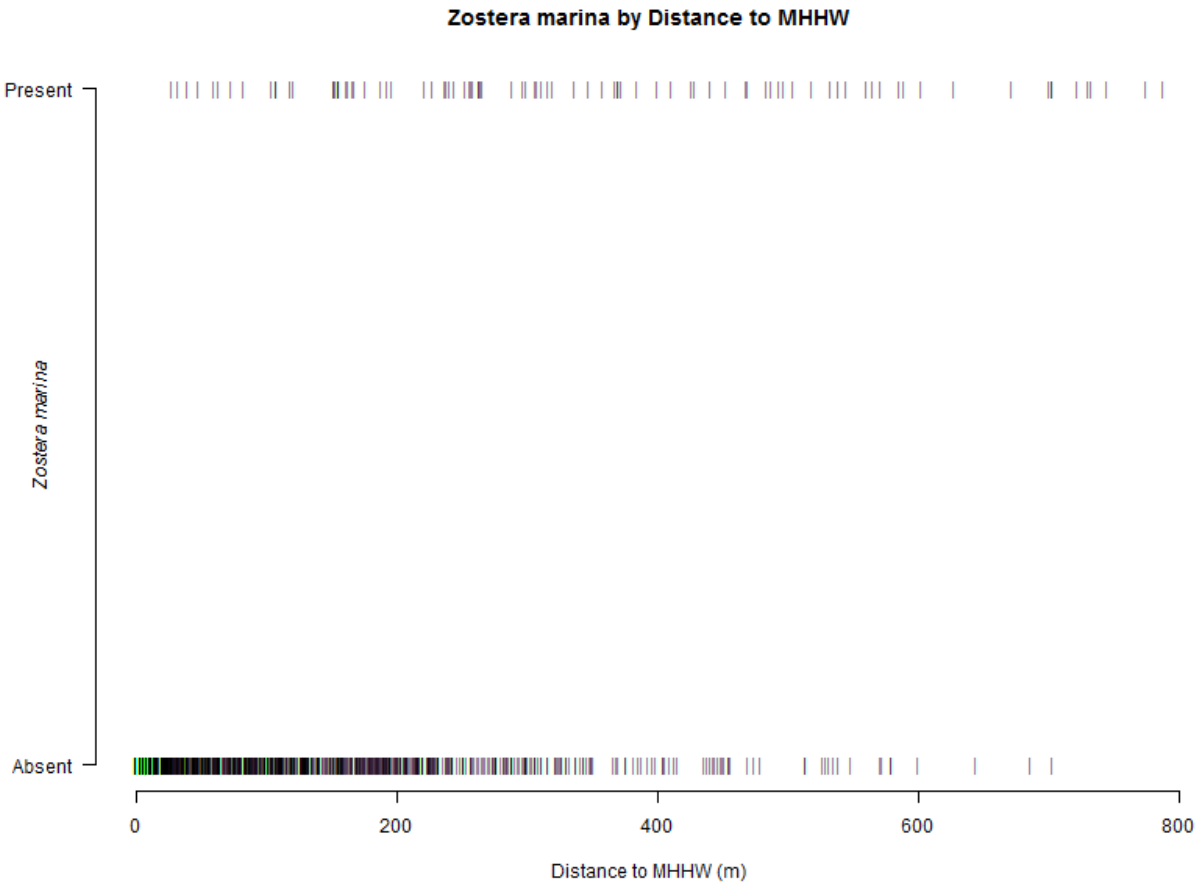
```
plot(zm~d2mouth, data=samplepoints, pch="|", xlab="Distance to Mouth (m)",
     ylab="", yaxt="n", frame=F, col=rgb(0,0,0,0.5),
     main="Zostera marina by Distance to Mouth")
axis(2,at=c(0,1), labels=c("Absent","Present"), las=2)
mtext("Zostera marina", 2, 2, font=3)
```



```
plot(zm~d2mllw, data=samplepoints, pch="|", xlab="Distance to MLLW (m)",
     ylab="", yaxt="n", frame=F, col=rgb(0,0,0,0.5),
     main="Zostera marina by Distance to MLLW")
axis(2,at=c(0,1), labels=c("Absent","Present"), las=2)
mtext("Zostera marina", 2, 2, font=3)
```



```
plot(zm~d2mhhw, data=samplepoints, pch="|", xlab="Distance to MHHW (m)",
     ylab="", yaxt="n", frame=F, col=rgb(0,0,0,0.5),
     main="Zostera marina by Distance to MHHW")
axis(2,at=c(0,1), labels=c("Absent","Present"), las=2)
mtext("Zostera marina", 2, 2, font=3)
```



Appendix B

TIDAL DATUM INFORMATION FOR YAQUINA BAY, OR

Conversion from MLLW to NAVD88 for Yaquina Bay, OR

U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
National Ocean Service

Station ID: 9435380 PUBLICATION DATE: 04/21/2003 Name: SOUTH BEACH, YAQUINA RIVER OREGON NOAA Chart: 18581 Latitude: 44Â° 37.5' N USGS Quad: NEWPORT SOUTH Longitude: 124Â° 2.6' W

T I D A L D A T U M S

Tidal datums at SOUTH BEACH, YAQUINA RIVER based on:

LENGTH OF SERIES:	19 Years
TIME PERIOD:	January 1983 - December 2001
TIDAL EPOCH:	1983-2001
CONTROL TIDE STATION:	

Elevations of tidal datums referred to Mean Lower Low Water (MLLW), in METERS:

HIGHEST OBSERVED WATER LEVEL (12/11/1969)		=	3.734
MEAN HIGHER HIGH WATER	MHHW	=	2.542
MEAN HIGH WATER	MHW	=	2.330
MEAN TIDE LEVEL	MTL	=	1.376
MEAN SEA LEVEL	MSL	=	1.358
MEAN LOW WATER	MLW	=	0.421
North American Vertical Datum	NAVD88	=	0.225
MEAN LOWER LOW WATER	MLLW	=	0.000
LOWEST OBSERVED WATER LEVEL (06/01/1973)		=	-1.073

North American Vertical Datum (NAVD88)

Bench Mark Elevation Information In METERS above:

Stamping or Designation	MLLW	MHW
C 590 1965	4.746	2.416
5380 D 1981	4.416	2.086
5380 E 1981	4.075	1.745
S 743 1988	4.394	2.064
5380 G 1997	4.559	2.229
5380 H 1997	4.644	2.314
5380 J 1997	5.191	2.861
5380 K 1997	6.349	4.019
5380 L 1997	7.749	5.419

Citations

R: R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

raster : Robert J. Hijmans (2014). raster: Geographic data analysis and modeling. R package version 2.2-31. http://CRAN.R-project.org/package=raster

rgdal : Roger Bivand, Tim Keitt and Barry Rowlingson (2014). rgdal: Bindings for the Geospatial Data Abstraction Library. R package version 0.8-16. http://CRAN.R-project.org/package=rgdal

EPA
United States
Environmental Protection
Agency

PRESORTED STANDARD
POSTAGE & FEES PAID
EPA
PERMIT NO. G-35

Office of Research and Development (8101R)
Washington, DC 20460

Official Business
Penalty for Private Use
$300