Aufgabenstellung

\*Ziel: \*Interaktives Tool, in dem verschiedene Szenarien an räumlichen Vorhersagemodellen getestet und verglichen werden können. Es soll somit zum verbesserten Verständnis von Herausforderungen räumlicher Vorhersagemodellen und deren Güteabschätzung beitragen.  
  
\*Daten: \*Das Ganze läuft auf simulierten virtuellen Daten: Prädiktoren werden simuliert, anhand von nutzerdefinierten Kriterien (z.B Anzahl Prädiktoren, räumliche Autokorrelation etc.).  
Auch die Response muss simuliert werden damit wir sie flächendeckend (zum Vergleich) zur Verfügung haben. Simulieren könnte man die Response aus einem Subset der Prädiktoren mit Hilfe einer Funktion (Komplexität und Rauschen ist nutzerdefiniert), d.h. wir stellen sicher, dass wir aus den Prädiktoren auf die Response schließen können und Aufgabe des Algorithmus (z.B. RF) ist es das dann später auch zu tun.  
  
Damit wir den RF trainieren können simulieren wir Trainingsdaten (Parameter: N, Verteilung (geclustert, zufällig, regulär),...).  
  
\*Modellierung: \*Mit den Prädiktoren und den Trainingsdaten wird ein Algorithmus (von mir aus erstmal nur Random Forest) trainiert. Tuning & Variablenselektion findet mit & ohne Variablenselektion statt und verschiedene Kreuzvalidierungsmethoden (zufällig und räumlich wobei es für die räumliche optional mehrere Varianten geben könnte) werden nicht nur zum Tuning sondern auch zur Validierung verwendet. Im Anschluss wird die AOA berechnet.  
  
\*Visueller/Quantitativer Output: \*Zufälliger CV Fehler, Räumlicher CV Fehler, realer Fehler, das Ganze mit & ohne Variablenselektion, Karte mit Differenz Real-Vorhersage, Karte mit DI/AOA  
  
\*Wichtig: \*Das Tool soll weiterentwickelt werden können. Deshalb unbedingt von Anfang an darauf achten, dass alles ordentlich dokumentiert wird!\*  
\*  
  
\*Einstiegsliteratur: \*Eine Schwierigkeit ist sicherlich die sinnvolle Simulation der virtuellen Prädiktoren und der Response. Carles hat in seiner Masterarbeit (Anhang) eine virtuelle Landschaft mit Prädiktoren und Response simuliert. Der Code ist auch auf Github. Vielleicht könnte man das sogar genauso übernehmen. Auch im Virtualspecies Paket (<https://onlinelibrary.wiley.com/doi/full/10.1111/ecog.01388>) wird sowas gemacht (wobei hier die Prädiktroren real sind). Im Anhang außerdem ein Artikel von mir in dem mal verschiedene Szenarien an Kreuzvalidierung und Variablenselektion getestet wurden.

Fragen:

* Mein Eindruck und was mir eventuell fehlt:
  + Viel Reproduzieren und korrekt anwenden aber nicht selbst forschen oder eine eigene Fragestellung beantworten
    - Was könnte man für eine eigene Fragestellung noch ergänzen?
    - Ist die Beantwortung einer Frage nicht notwendig? Was wäre dann die Fragestellung?
    - Ich würde gerne Sachen herausfinden und eventuell neue Zusammenhänge erklären.
  + Soll ich die BA auf Englisch verfassen? Auf jeden Fall.
  + Fokus auf interaktives Tool und später nutzen, um eventuell Hinweise zu generieren
* Simulation von Landschaften:
  + Anzahl Prädiktoren (umsetzbar)
    - Was passiert mit den 8 Variablen nicht genutzt werden`? (S.16)
  + Räumliche Autokorrelation
    - Womit? Mit Gauss?
    - “Covariates are 2-dimensional stationary and isotropic Gaussian random fields subject to spatial autocorrelation, expressed with spherical semivariograms”
    - Was genau heißt das? Was hat Carles angenommen? Wieso eignet sich das besonders gut? Wieso nicht eins der anderen?
    - In welcher Range? 5, 10, 20, und 40% Autokorrelation?
    - Wie modelliere ich Komplexität und Rauschen? Random und Spatial Noise?
  + Wie groß soll die Landschaft sein?
  + Was noch?
* Trainingsdaten:
  + N:
    - 50, 100, 150 oder noch mehr?
    - Er hat nur „Punkte“ keine Polygone benutzt ABER geclusterte Punkte repräsentativ für Polygone, oder?
  + Verteilung:
    - (random, regular, clust1, clust2 und nonunif)?
* Model:
  + Welche Modelle? Unterscheidung von räumlichen und Standard-Modellen oder generell andere Algorithmen?
* Validierung:
  + Was ist die normale Surface Validation?
  + Was ist der genaue Unterschied zwischen sbLOOrange und sbLOOndm CV?
  + Welche CV Methode haben wir bereits im Kurs genutzt?
  + Hättest du sie gerne im Tool?
  + Welche Selektionsalgorithmen möchtest du einbetten?
    - RFE, FFS jeweils mit räumlicher und zufälliger Kreuzvalidierung?
* Was ist der reale Fehler?

**Aufbau der App**

* Auswahl der Prädiktoren:
  + Parameter: Anzahl, Landschaftsmodell (NLM), räumliche Autokorrelation?
* Simulieren der Response
  + Simulieren könnte man die Response aus einem Subset der Prädiktoren mit Hilfe einer Funktion (Komplexität und Rauschen ist nutzerdefiniert), d.h. wir stellen sicher, dass wir aus den Prädiktoren auf die Response schließen können und Aufgabe des Algorithmus (z.B. RF) ist es das dann später auch zu tun
* Simulieren der Trainingsdaten
  + Damit wir den RF trainieren können simulieren wir Trainingsdaten
  + Parameter: N, Verteilung (random, regular, weak clustering, strong clustering, uniform)
* Modellierung
  + Mit den Prädiktoren und den Trainingsdaten wird ein Algorithmus (Random Forest) trainiert
  + Tuning & Variablenselektion finden mit & ohne Variablenselektion statt und verschiedene Kreuzvalidierungsmethoden (zufällig und räumlich, wobei es für die räumliche optional mehrere Varianten geben könnte) werden nicht nur zum Tuning, sondern auch zur Validierung verwendet
  + Berechnung der AOA
* Visueller/Quantitativer Output
  + \*Zufälliger CV Fehler, Räumlicher CV Fehler, realer Fehler, das Ganze mit & ohne Variablenselektion, Karte mit Differenz Real-Vorhersage, Karte mit DI/AOA

Wichtig: Das Tool soll weiterentwickelt werden können. Deshalb unbedingt von Anfang an darauf achten, dass alles ordentlich dokumentiert wird!

Größte Unklarheiten bei Programmierung:

* Wie lasse ich die NLM auswählen?
* Wie modelliere ich konkret die Response und wie füge ich Rauschen bzw. Noise hinzu?
* Implementieren verschiedener Auswahlmöglichkeiten für Anzahl und Verteilung
* Räumliche CV/ Zufällige CV
* Wie mach ich räumliche Kreuzvalidierung, wenn ich keine Polygone habe?

**2. Fragestunde (09.03.22)**

Zeigen, was ich bisher gemacht habe:

* Erste Erfahrungen mit RShiny gesammelt
* Prototyp der App entwickelt
  + Erstes Layout der App
  + Bereits einige Funktionalitäten implementiert

NLMs:

* Anzahl lässt sich über ausgewählte NLMs bestimmen
* Nur 10 verschiedene NLMs in App integriert aufgrund von Problemen (lässt sich beheben)
* Viele Parameter können bei der Erzeugung der einzelnen NLMs übergeben werden aber aktuell sind sie nicht einstellbar (siehe Funktion generate\_predictors()) (Experten-Modus)
* Soll das eventuell geändert werden?
* Sollen vom gleichen NLM-Typen mehrere NLMs generiert werden können? Hätte das einen Nutzen? (Experten-Modus)
* Keine Legenden
* Maß für Korrelation der simulierten Ausgabe?
* Virtual species R-Package
* Nicht alle Prädiktoren mit in die Berechnung der Ausgabe einbringen
* 🡪 Auswahl ermöglichen
* Anzahl Variablen und dann random generieren oder specify

Sample Points:

* Sind die Variablen so in Ordnung?
* Sinn für N-Cluster? Cluster Dichte?
* Vielleicht den Output Plot etwas verändern und nur sichtbar machen, sobald man an den Parametern herumspielt

Simulated outcome:

* Funktion, die aus allen Prädiktoren einen Ausdruck generiert, sodass die verschiedenen Prädiktoren-Layer anschließend miteinander verrechnet werden können
* Wird jetzt zufällig generiert, aber soll es immer die gleiche sein? Welche Operanden soll ich mit einbringen?
* Noch kein Rauschen/Noise eingefügt
* Kann erst berechnet werden, wenn vorher Prädiktoren generiert wurden
* Skalieren des Outcomes
* Noise auswählbar (random und spatially correlated noise)
  + Wie viel Noise? Einfach addieren ist nicht so gut, weil es zu viel Noise wird (Kurzfristige Lösung: Mit 0.2 multipliziert)
* Knopf für Set seed -> ABER immer der gleiche seed (Eine bestimmte Auswahl lässt sich bisher nicht speichern)

Prediction:

* Trainingsdaten werden auf Basis der sampling points extrahiert
* Random Forest Modell
* Noch ohne Kreuzvalidierung
  + Was für eine räumliche Kreuzvalidierungsmethode soll ich implementieren?
  + Nearest Distance Matching
  + Global\_validation/ Sb Blockwise Caret?
  + BlockCV
  + Blockwise? Oder jeden Punkt einmal auslassen?
  + Nochmal erläutern wie blockwise funktioniert
* Literatur für Kreuzvalidierungsmethoden
* Was geben die Fehlerwerte nochmal an? RMSE, R2 und MAE?
* MAE selbst berechnet, stimmt das so?
* Mit oder ohne Variablenselektion?

Difference:

* Wird berechnet
* Leider nicht dieselbe Legende wie outcome und prediction

Weitere Ausgaben:

* AOA, DI und Tabelle der verwendeten Trainingsdaten

**Weitere Fragen:**

* Worauf wollen wir hinarbeiten?
* Welche Zusammenhänge sollen hier dem Nutzer klar werden?
* Wie kann ich diese besser unterstreichen?
* Was soll alles als Ergebnis angezeigt werden?
* Wie soll ich die Ergebnisse anordnen?
  + Verschiedene Seiten? Aber dann kann man schöne Zusammenhänge nicht so gut sehen
* Konkrete Aufgabenstellung?

TO-DO:

CAST::plot\_geodist