

TARTU ÜLIKOOL
LOODUS- JA TÄPPISTEADUSTE VALDKOND
ÖKOLOOGIA JA MAATEADUSTE INSTITUUT
BOTAANIKA OSAKOND

Lauri Laaspere

**Karjatamise ja varjutamise mõju krohmseente ning taimede vahelistele
interaktsioonidele**

Bioloogia ja elustiku kaitse

Bakalaureusetöö (12 EAP)

Juhendajad:

Prof. Marina Semtšenko

PhD Siim-Kaarel Sepp

Tartu 2024

Sisukord

Infoleht	3
Sissejuhatus	5
1. Kirjanduse ülevaade	6
1.1 Arbuskulaarne mükoriisa	6
1.2 Süsinik ja fosfor kui interaktsioonide modulaatorid	7
1.2.1 Sümbioosi varieerumine	7
1.2.2 Bioloogiline turg	8
1.2.3 Karjatamise mõju arbuskulaarsele mükoriisale	9
1.2.4 Varjutamise mõju arbuskulaarsele mükoriisale	12
1.3 Masinõpe ökoloogias ja programm AMFinder	15
2. Metoodika	17
2.1 Andmete allikad ja tehtud katse ülevaade	17
2.2 AMFinderi kasutamine	18
2.3 Andmeanalüüs	20
3. Tulemused	21
3.1 AMFinderi mudelite tulemused	21
3.2 Tulemused klassikalise meetodiga	24
4. Arutelu	26
Kokkuvõte	30
Summary	32
Tänuavaldused	34
Kasutatud kirjandus	35
Lisad	40
Lisa 1. Koostatud R skript	40
Lisa 2. Jooksutatud käsud	50
Lisa 3. Modifitseeritud fail amfinder_convert.py	51
Lihlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks	60

Infoleht

Karjatamise ja varjutamise mõju krohmseente ning taimede vahelistele interaktsioonidele

Karjatamine ja varjutamine võivad mõjutada arbuskulaarset mükoriisat moodustavate seente kolonisatsiooni taimejuurtes nii positiivselt kui ka negatiivselt. Mõlema faktori mõjusid on eelnevalt uuritud, kuid karjatamise mõju kohta puudub ühine arusaam. Oluliseks peetakse karjatamise intensiivsust ja kestvust, kuid mõju oleneb kontekstist. Varjutamise puhul on leitud, et vähese valguse korral võib seente kolonisatsioon taimejuurtes väheneda. Antud töö eesmärk oli katsetada kolonisatsiooni määramist programmiga AMFinder ning võimekust tuvastada karjatamise ja varjutamise võimalikke mõjusid. Selleks kasutati katseandmestikku, kus erinevate töötlustega simuleeriti karjatamist, varjutamist ja karjatamise puudumist.

Programm AMFinder kasutab kolonisatsiooni määramiseks masinõpet ning tehismärgivõrke, et mikroskoobiga tehtud slaidipiltidelt tuvastada arbuskulaarse mükoriisa struktuure. Masinõppe kasutamine võimaldab andmeid analüüsida ning seoseid tuvastada kiiremini.

Võrdluseks määrati kolonisatsioon ka klassikalise, inimhindaja meetodiga. Selle tulemustest selgus, et varjutamise töötlustega taimede arbuskulaarse mükoriisa kolonisatsioon oli keskmiselt kõrgem kui teistel töötlustel. Sama mustrit programm AMFinder tuvastada ei suutnud.

AMFinder katsetamiseks treeniti erinevaid mudeleid, mille puhul kasutati treeningandmestiku aditiivset lähenemist. Tõenäoliselt oleks AMFinder suutnud tuvastada erinevused töötluste vahel, kui oleks kasutatud rohkem treeningandmeid, andmete augmenteerimist ning esitatud kogu andmestik korraga mudelile treenimiseks. Siiski, võrreldes klassikalise meetodiga, on AMFinder kasutamisel potentsiaali, kuna paremini treenitud mudeli puhul oleks võimalik kolonisatsiooni määrata oluliselt kiiremini ning inimhindaja subjektiivsuseta.

Märksõnad: masinõpe, arbuskulaarne mükoriisa, karjatamine, varjutamine, bioloogilised turud

CERCS teadusala kood: B230 Mikrobioloogia, bakterioloogia, viroloogia, mükoloogia; B270
Taimeökoloogia

Effects of grazing and shading on interactions between plants and arbuscular mycorrhizal fungi

Grazing and shading can positively or negatively affect arbuscular mycorrhizal fungal (AMF) colonisation in plant roots. Both factors have been researched, but as of right now, there is no consensus. Grazing has been found to have positive as well as negative effects that depend on the intensity and duration. For shading, there is a common understanding that it negatively affects the colonisation of plant roots due to light deprivation under intense shade. This thesis aimed to test the program AMFinder and its ability to detect AM fungal colonisation in plant roots under simulated grazing (clipping) and shading. For this purpose, an experimental dataset was used where plants were subjected to clipping, shading or no clipping treatments.

AMFinder uses machine learning and neural networks to detect AM fungal structures in microscope images. The use of machine learning enables higher data throughput and faster analysis.

Colonisation was also quantified using a regular human-based method, plants subjected to the shading treatment had higher average AM fungal colonisation than other treatments. Measurements conducted with AMFinder did not distinguish the same pattern.

This was probably due to the structure of training and the datasets used. Different AMFinder models were exposed to the training sets additively, which did not produce sufficiently good models. Better models could be generated with more training data, data augmentation, and different training approaches, such as exposing the model to the maximum available data in one run. Machine learning and computer vision still have the potential to accelerate the quantification of AM fungal colonisation in plant roots and the analysis of large datasets.

Keywords: machine learning, arbuscular mycorrhizal fungi, grazing, shading, biological markets

CERCS research field code: B230 Microbiology, bacteriology, virology, mycology; B270 Plant ecology

Sissejuhatus

Arbuskulaarne mükoriisa (AM) on iseloomulik ligikaudu 72% soontaimedele ning on taimedele oluline taimedele mullast toitainete omastamisel (Brundrett & Tedersoo, 2018). Sümbioos varieerub parasitismist tugeva mutualismini, sõltudes nii biotilistest kui ka abiootilistest faktoritest. Sellist olukorda saab võrrelda bioloogilise turuga, kus taimed ja seened peavad panustama võrdväärselt, et saada ligipääs turul pakutavatele hüvedele (Wyatt *et al.*, 2014).

AM toimimist mõjutab tugevalt taime poolt fikseeritud süsiniku hulk, mis on seotud valguse kättesaadavusega, mida omakorda mõjutavad nii karjatamine kui ka varjutamine naabertaimede poolt. Mõõdukas karjatamine soodustab liigirikkkust, kuid intensiivse karjatamise põhjustatud biomassi kadu piirab taime ressursse ning võimalusi panustada mükoriisa ülalpidamisse. Seniste uuringute tulemusena peetakse oluliseks näiteks karjatamise intensiivsust ja uuringu kestvust (Faghihinia *et al.*, 2020a), kuid ühist arusaama ei ole. Süsiniku limitatsiooni võib põhjustada ka varjutamine – tugeva varju puhul võib arbuskulaarse mükoriisa kolonisatsioon taimejuurtes väheneda (Konvalinková & Jansa, 2016). Siiski täpseid mehhanisme ei teata.

Karjatamise ja varjutamise mõjude uurimiseks kasutati andmestikku katsest, kus taimi kasvatati varjus, lõigati lehti karjatamise simuleerimiseks ja jäeti lehed lõikamata karjatamise puudumise simuleerimiseks. Arbuskulaarmükoriisete seente kolonisatsioon taimejuurtes määrati kahte meetodit kasutades. Klassikalise inimhindajaga meetodi puhul tehti mikroskoobiga sada vaadet värvitud ja slaididele monteeritud taimejuurtest (McGongile *et al.*, 1990; Vierheilig *et al.*, 1998). Samuti kasutati programmi AMFinder hindamiseks samadest slaididest tehtud pilte. AMFinder rakendab kolonisatsiooni määramiseks masinõpet (Evangelisti *et al.*, 2021). Kiirem andmete töötlemine võimaldaks kiirendada analüüsi ning luua parema arusaama sümbioosi interaktsioonide toimimisest erinevate süsinikku limiteerivate faktorite juuresolekul. Töö eesmärk oli tuvastada karjatamise ja varjutamise mõju AM seente kolonisatsioonile taimejuurtes ning võrrelda kahe meetodi, inimhindamine ja AMFinder, võimekust kolonisatsiooni hindamisel.

1. Kirjanduse ülevaade

1.1 Arbuskulaarne mükoriisa

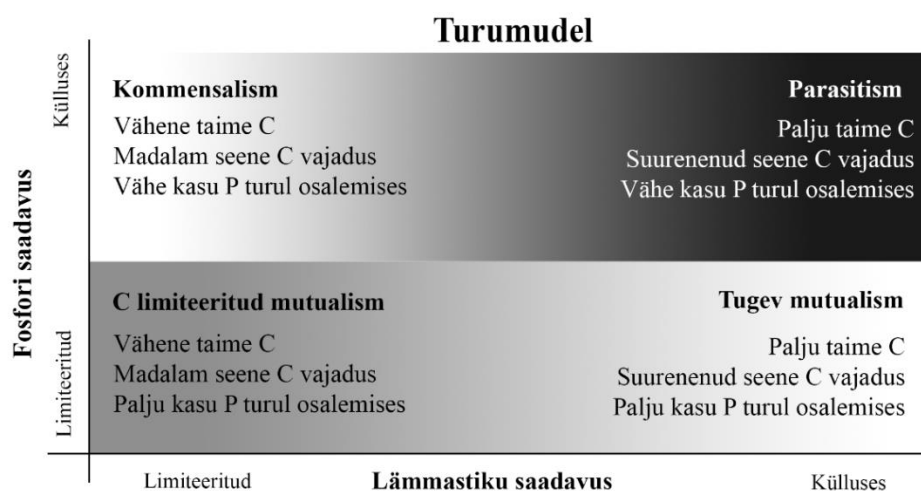
Mükoriisaks ehk seenjuureks nimetatakse sümbioosi taimejuure ja seene vahel, kus seen varustab taime vee ning toitainetega, peamiselt fosfori ja lämmastikuga, saades taimelt vastu elutegevuseks vajalikke süsinikuühendeid (Smith & Read, 2008a). Peamiselt eristatakse nelja mükoriisatüüpi: arbuskulaarne mükoriisa (AM), ektomükoriisa (EcM), erikoidne (ErM) ja orhidoidne mükoriisa (OrM), millest esimene on enimlevinud (ca 72% soontaimedest on arbuskulaarmükoriissed, Brundrett & Tedersoo, 2018). Arbuskulaarne mükoriisa on ka vanim mükoriisatüüp, olles vähemalt 400 miljonit aastat vana (Parniske, 2008).

Arbuskulaarset mükoriisat moodustavad krohmseente (*Glomeromycota*) hõimkonda kuuluvad seened (Brundrett & Tedersoo, 2018). Arbuskulaarse mükoriisa põhilised osad on taimejuur, juuresisesed struktuurid ja seeneniidistik ehk mütseel mullas. (Smith & Read, 2008b) AMile on iseloomulik arbuskulite olemasolu, need on puu- või pöõsasarnased struktuurid juurerakkude perimembraanses ruumis, mille kaudu toimub ainete transport seenest taime ja vastupidi (Parniske, 2008). Eristatakse ka seenehüüfidest moodustunud paljugeerdunud struktuure (*coils*) (Smith & Read, 2008b). Lisaks arbuskulitele on tüüpilisteks tunnusteks vesiikulid, millel arvatakse olevat varuainete säilitusfunktsioon (Parniske, 2008). Vesiikulid võivad paikneda rakkude vahel, nagu hüüfid, või ka juurerakkude sees (Smith & Read, 2008b).

1.2 Süsinik ja fosfor kui interaktsioonide modulaatorid

1.2.1 Sümbioosi varieerumine

Mutualistlikud interaktsioonid, kus erinevad liigid vahetavad vastastikku kasulikke teenuseid, on laialt levinud ja olulised liikide kooseksisteerimise võimaldajad. Arbuskulaarmükoriissed assotsiatsioonid on sümbioosid, mis varieeruvad mutualismist parasitismini (Johnson *et al.*, 1997). Mutualismi säilimise tagavad erinevad stabiliseerivad ja kontrollivad mehhanismid, mis väldivad partneritel teineteise ära kasutamist ja petmist (Chomicki *et al.*, 2020). AM sümbioosi funktsioone juhivad ökoloogilisel skaalal ressursid nagu süsinik, fosfor ja lämmastik, täpsemalt nende kättesaadavus ja omavahelised vahekorrad. Samad faktorid mõjutavad ka liikide evolutsiooni, põhjustades seene ja taimede vastastikust kohastumist ning liikide evolutsiooni tugevama või nõrgema mükoriisse sõltuvuse suunas (Johnson, 2010). Ressursside jaotumise mehhanisme saab käsitleda bioloogilise turu raamistikus, kus lämmastik ei osale otseselt turumehhanismides, kuid selle kättesaadavus reguleerib nii seenpartneri süsiniku kui ka taime fosfori vajadust (Joonis 1). Piisava lämmastiku olemasolu ning fosfori kättesaadavuse piirang soodustab tugevat mutualismi ning turul osalemist, kuid sama süsteem kaldub seenepoolse parasitismi suunas mõlema ressursi külluses, kuna taimel on sel puhul kasulikum ise fosforit omastada, kuid seene kasvu piiravaks faktoriks on sel juhul süsiniku vähesus (Johnson, 2010). Uuringud, kus on käsitletud arbuskulaarmükoriisete seente inokulatsiooni mõju taimedele, on sarnaselt leitud, et mükoriisal on nii positiivseid kui ka negatiivseid mõjusid ning selle funktsioon sõltub biotilistest ja abiotilistest faktoritest, kuid mõju taimele on rohkem positiivne fosfori kui lämmastiku poolt limiteeritud süsteemides (Hoeksema *et al.*, 2010).



Joonis 1. AM seente sümbioosi eeldatavad funktsioonid bioloogilise turu raamistikus. Tähisted C – süsinik, P – fosfor. Joonis kohandatud Johnson (2010) järgi.

1.2.2 Bioloogiline turg

Bioloogilise turu teooria järgi vahetavad turul kaks osapoolt kaupu mõlema kasuks, mida võib nimetada ka mutualismiks (Noë & Hammerstein, 1995). Mutualismi tekkemehhanisme on evolutsiooniteoorias raske kirjeldada, kuna looduslik valik soosib petised, kes ei investeeeri sama palju ressursse koostöök, kuid saavad kergema ligipääsu hüvedele, mistõttu töötab looduslik valik mutualismile vastu. Liigisiselt soosib koostöö püsimist hõimuvallik, kuid liikidevahelise koostöö tekkeks ja püsimiseks on vajalikud lisategurid (Wyatt *et al.*, 2014).

Taimede ja arbuskulaarset mükoriisat moodustavate seensümbiontide interaktsioonid on keerukad, hõlmates tihti mitut partnerit. Paljude osapooltega interaktsioonid suurendavad omakorda riski, et üks osapool kasutab teist ära, kuid arbuskulaarse mükoriisa puhul on kontroll kahesuunaline, sarnanedes turuolukorrale, kus toimivad nõudluse ja pakkumise suhted (Kiers *et al.*, 2011).

Wyatt *et al.* (2014) pakuvad oma töö tulemustes taolise turuolukorra tekkimise põhjusena vastastikku proportsionaalse partnerisse panustamise. Ehk kui taim saab ühelt seensümbiondilt 2/3 vajalikest kaupadest ja 1/3 teiselt, siis saab esimene sümbiont 2/3 taime kaubeldavast süsinikust ja teine omakorda 1/3. Taolise äratundmismehhanismi eksisteerimine võimaldaks evolutsiooniliselt stabiilset strateegiat, kus petmine ei ole võimalik. Juhul, kui turul osalemine on soositud, on soositud ka ühe ressursi iseseisvalt omastamise lõpetamine ja selle ainult turult hankimine, kuna spetsialiseerumine on efektiivsem. Taimed omastavad efektiivsemalt süsinikku ja hangivad turult fosforit. Stabiilses mutualismis ei suurene seene kohasus, kui ta omastab lisaks süsinikku iseseisvalt (Wyatt *et al.*, 2014). Samuti on mutualism eelistatud juhul kui partnerite ligipääs ressursile, mida nad saavad turult, puudub või on piiratud.

Turuolukorda mõjutab ka partnerite rohkus. Mutualismi püsimine on soodustatud juhul kui võimalike konkurentide arv suureneb. Rohkemate seenpartnerite osalemine turul vähendab üksikute seenpartnerite kontrolli turul oleva fosfori üle, mis tingib konkurentsi saadavaloleva süsiniku üle ja fosfori hinna alanemise. Vähendades taimpartnereid või suurendades seenpartnerite arvu, on fosfori kogus turul suurem võrreldes süsinikuga. Taimede puhul tingib see panustamist rohkema süsiniku hankimisse, kas siis enda tarbeks või sellega turul kauplemiseks, kuna selle väärtus on suurenenud. Rohkemate seenpartnerite olemasolu suurendab iga seene koostöövalmidust ning tõstab taime kohasust. Seened peavad turul pakkuma rohkem fosforit, et pääseda ligi limiteeritud süsinikule, suurendades seekaudu taime kohasust. Rohkem taimpartnereid ühe mükoriisaseene kohta vähendab omakorda taime kohasust, kuna taimed peavad konkureerima ühise fosfori varustaja üle. Samas sarnaselt turuolukorrale, suureneb

mükoriisaseente koostöövalmidus, kuna fosfori „turuhind“ tõuseb. Samuti tuuakse välja lisaefektina, et seen võib pakkuda turul vähem fosforit, kuna sama „hinna“ eest saab rohkem süsinikku. (Wyatt *et al.*, 2014)

Kirjeldatud olukorrad tingivad arbuskulaarse mükoriisa puhul ressursside kättesaadavuse turul ning koostöö stabiilsuse, kus vastastikune kasu suureneb konkurentsitingimustes analoogselt toimiva turumajandusega (Kiers *et al.*, 2011).

1.2.3 Karjatamise mõju arbuskulaarsele mükoriisale

Maakasutuse muutused, nagu karjatamise intensiivistumine või lakkamine rohumaadel, toovad kaasa muutusi taimekooslustes ning taimede ja seenpartnerite süsiniku ning toitainete vahetuses. Üldiselt on märgitud, et karjatamisel on varieeruv mõju liigilisele mitmekesisusele, kus suurema produktiivsusega rohumaadel on karjatamisel üldiselt positiivsem mõju liigirikkusele (Lezama *et al.*, 2014). Pikaajaline liiga intensiivne karjatamine eemaldab süsteemist taimset biomassi ja tingib taimede koosluse muutumise. Karjatamise puudumine on seotud süsiniku madalama allokatsiooniga taime maa-alustesse osadesse, millega võib kaasnedä väiksem juurte biomass ja juureeritiste hulk ning madalam mikroobne biomass (Wilson *et al.*, 2018). Karjatamise lakkamisega kaasneb tihti ka puude sissetung, mis tingib alustaimestiku varjutamise, põhjustades muutusi süsiniku kättesaadavuses ning aja jooksul ka rohumaadele iseloomulike liikide väljasuremist (A.R. & S.J., 2001; Neuenkamp *et al.*, 2016). Mullas leiduv fosfor ja lämmastik on taimekasvaks olulised elemendid ning karjatamisel on täheldatud nii negatiivseid kui ka positiivseid mõjusid nende elementide kättesaadavusele. Saab samuti eeldada, et fosfor ja lämmastik mõjutavad AM seente rikkust ning ohtrust taimejuurtes kuid ka sel puhul on tehtud uuringutes ja katsetes ebakõla (Faghihinia *et al.*, 2020a).

Medina-Roldán *et al.* (2012) tehtud katses, kus uuriti taimeliikide *Eriophorum vaginatum* ja *Nardus stricta* näitel karjatamise mõjusid mullale ja taimede vahelisele konkurentsile, leiti, et karjatamine mõjutab kaudselt kahe liigi interaktsioone mullas saadavaloleva lämmastiku ja mullamikroobide kaudu. Autorid järeldavad ka, et karjatamine on üks peamisi radu, mille kaudu toimuvad muutused mulla omadustes ja toitainete kättesaadavuses. Mõjusid arbuskulaarsele mükoriisale selles katses ei uuritud.

Karjatamise mõjud AM sümbioosile sõltuvad karjatamise intensiivsusest – mõõdukas karjatamine tõstab taimeliikide mitmekesisust, kuna see võib suurendada liikide kooseksisteerimist läbi eri liikide konkurentsivõime võrdsustamise ja suurendada investeeringut seenpartnerisse suurenenud ressursivajaduse tõttu (Faghihinia *et al.*, (2020a). Loomade väljaheited, nagu sõnnik ja uriin,

sisaldavad samuti lämmastikku ja fosforit, mis karjatamise tulemusel satuvad tagasi mulda, kuigi kogus sõltub mitmest tegurist, nagu näiteks majandusviis ja väetamine (Haynes & Williams, 1993; Vertès *et al.*, 2019).

Karjatamise indutseeritud muutused keskkonnas mõjutavad nii taime kui ka mulla tunnuseid, kuid nende olulisus ning ulatus varieerub ning mõjud AM seentele on kontekstipõhised (Faghihinia *et al.*, 2020a). Vähenenud maapealne biomass vähendab taime süsinikusidumisvõimet, mis omakorda vähendab taime maa-alustesse osadesse suunatavat süsinikku ja mõjutab seeläbi AM moodustavaid seeni. Ba *et al.* (2012) tehtud katses stepiniidul, kus mõõdeti AM seente liigilist koosseisu ja rohkust vastavalt karjatamise intensiivsusele, selgus, et AM seente mitmekesisus oli suurem kerge ja mõõduka karjatamisintensiivsuse juures ning vähenes intensiivse karjatamise puhul. Autorid arvavad ka, et intensiivselt karjatatud ja suure maapealse biomassi kaoga alal võis väheneda AM seente levik, kuna vähenes AM moodustavate seente liigiline mitmekesisus.

Tehtud uuringute tulemused pikaajalise karjatamise mõjudest on vastuolulised. Paremaks mõistmiseks oleks vaja rohkem tähelepanu pöörata karjatamise mõjudele erinevate intensiivsustasemetes juures ning teha katseid ka suuremal skaalal, et tuvastada herbivooria efekti erineva kliimaga piirkondades (Faghihinia *et al.*, 2020a).

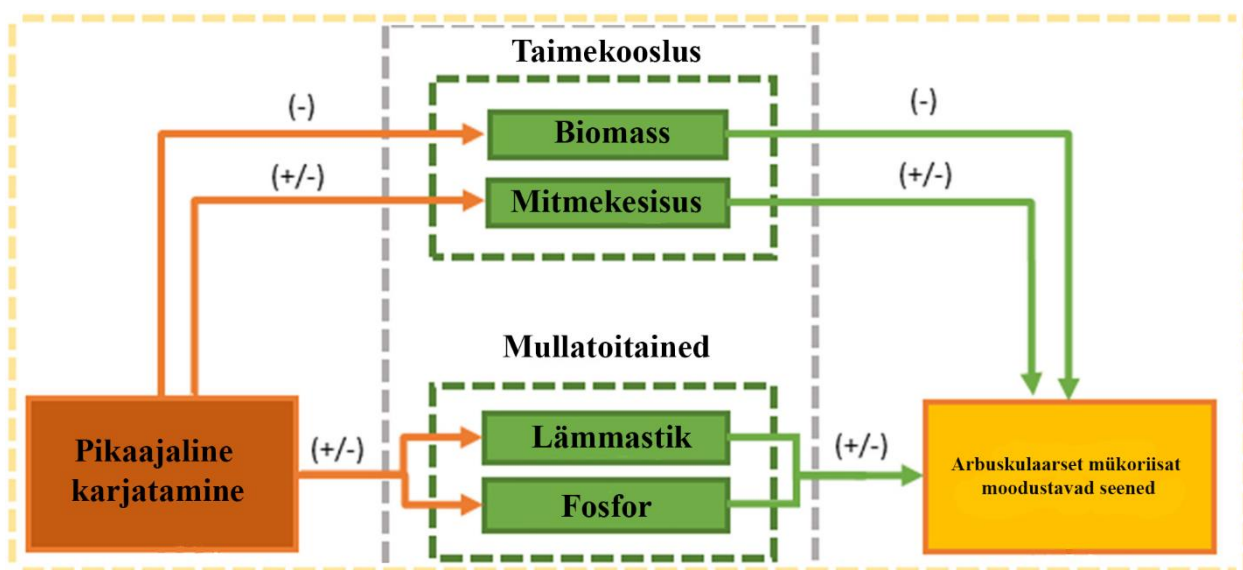
Karjatamise mõjud AM sümbioosile võivad varieeruda ka seetõttu, et mõnes olukorras tõstavad AM seened taimede karjatamise taluvust ning uuringute ebakõlad võivad tuleneda mõõdetud AM seente tunnusest, taime ja mulla enda mõjudest seentele ning uuringute ajalise kestuse variatsioonist (Heyde *et al.*, 2017). Hüpooteeside testimiseks tehtud katses leiti samuti, et karjatamise mõju AM seentele ei ole ühene – karjatatud aladel oli küll mullasiseste hüüfide pikkus väiksem, kuid eoste tihedus suurem ning mõju taimejuurte kolonisatsioonile ei ilmnunud. Olulise faktorina hinnati, mis ajast alates karjatamine lõpetati ning parema arusaamise karjatamise mõjude loomiseks soovitatakse hinnata AM seente erinevaid näitajaid ja teha katseid pikema aja jooksul.

Klironomos *et al.* (2004) tehtud simuleeritud karjatamise katses leiti, et lõikamisel olid negatiivsed mõjud juuresisestele hüüfidele ja arbuskulitele, samas vesiikulite ja spooride produktsioon oli stimuleeritud, kuigi mõjud erinesid sõltuvalt AM seene liigist. Samas Faghihinia *et al.* (2020b) tehtud katses, kus uuriti pikaajalise karjatamise mõju erinevate intensiivsustasemetes juures, leiti, et intensiivistatud karjatamine mõjutas negatiivselt mullasiseste hüüfide tihedust, kuid mitte juurte kolonisatsiooni AM seente poolt. Kõige suurem erinevus juuresiseste ja -väliste struktuuride suhtes ilmnis mõõduka karjatamise juures. Karjatamise vähest mõju juurte kolonisatsioonile täheldasid ka Barto & Rillig (2010) oma metanalüüsis. Faghihinia *et al.* (2020a) leiavad samuti

oma analüüsis, et AM struktuurid reageerivad erinevalt herbivooriale: mullasiseste hüüfide pikkus üldiselt väheneb, kuid uuringute tulemused varieeruvad taimejuurte kolonisatsioonis ning spooride tiheduse poolest.

Karjatamise intensiivsus mõjutab selgelt mükoriisete seente kooslust ning mükoriisa toimimist, mis omakorda võib mõjutada ökosüsteemi funktsioone ning protsesse (Faghihinia *et al.*, 2020b). Mõõduka intensiivsuse juures on taimetele tõenäoliselt kuluefektiivsem investeerida mükoriisapartnerisse, kes otsiks mullast vajalikke toitained, kui hankida ressursse iseseisvalt, et tagada häiringujärgne võsude juurdekasv.

Kokkuvõtvalt võib öelda, et uuringud on mitmekesised ja omavahel vasturääkivad (Joonis 2). Hindamaks karjatamise mõju taimede ja AM seente vahelisele mutualismile tuleb samal ajal hinnata palju teisi nii abiootilisi kui ka biootilisi tegureid ning arvestada süsteemi omapärasusega. Olulisena saab välja tuua karjatamise intensiivsuse, mõjude ajalise dünaamika ja AM seente erinevate struktuuride mõõtmise. Üldiselt võib eeldada, et mõõdukas karjatamine soodustab taimepoolset investeringut mükoriisasse, kuna karjatamise tõttu süsteemist kadunud toitained suurendavad taime vajadust ressursside järele, mis positiivse tagasisidena suunab taime investeerima mükoriisasse.



Joonis 2. Arbuskulaarset mükoriisat moodustavaid seeni mõjutavad võimalikud tegurid, mis võivad olla kas positiivsed või negatiivsed, vastavalt + või -. Joonis on kohandatud Faghihinia *et al.* (2020a) järgi.

1.2.4 Varjutamise mõju arbuskulaarsele mükoriisale

Fotosünteesiliselt aktiivse kiirguse muundamine keemiliseks energiaks ja süsiniku sidumine fotosünteesis on elutähtis, et taimed saaksid konkureerida teiste individidega ning paljuneda (Givnish, 1988). Varjutamise all mõistetaksegi üldiselt valguse kättesaadavuse piiramist taimele, mis võib mõjuda stressorina ja põhjustada fikseeritud süsiniku limitatsiooni. Taimedel on erinevad mehhanismid vähese valgusega hakkama saamiseks, mis hõlmavad nii morfoloogilisi, füsioloogilisi kui ka sümbiootilisi muutusi.

Konvalinková & Jansa (2016) toovad oma ülevaateartiklis välja, et AM seentel on tugev reaktsioon peremeestaimetele saadaoleva valguse järsule vähenemisele, mille tagajärjel lülitatakse kiiresti välja fosfori transport taimetele. Looduslikus keskkonnas toimub valgustingimuste muutumine kiiresti, näiteks pilvkatte tekkimisel. Kui kiiresti reageerivad olude muutumisele AM seened, pole põhjalikult uuritud. Teadustöid, mis keskenduksid AM mutualismi funktsioneerimisele muutuvate valgustingimuste korral ning mis kestnuks kauem kui kümme päeva, on vähe. Konvalinková *et al.* (2015) ja Fellbaum *et al.* (2014) leidsid, et fosfori transport väheneb kiiresti. Samas olulisi muutusi lämmastiku transpordis pole täheldatud (Hodge & Fitter, 2010).

Hoeksema *et al.* (2010) on oma metanalüüsis toonud välja, et kuigi katse asukoha faktoril (kasvuhoone vs. välitingimused) pole olulist rolli, võivad erinevad valgustingimused mõjutada süsiniku jaotamist ning taime ja seene vahelise mutualismi toimimist. Seetõttu on erinevate ja muutlike valgustingimuste tõttu katsete korratavus raskendatud (Konvalinková & Jansa, 2016).

Konvalinková & Jansa, (2016) arutlevad, et vähese valguse ja seekaudu ka piiratud süsiniku kättesaadavuse tingimustes kasvab taime jaoks AM seene ülalpidamine oluliselt ning taimetele ei ole kasulik seenpartneri süsinikuga varustamine. Juurte mükoriisne kolonisatsioon võib antud olukorras väheneda, olles potentsiaalselt reguleeritud kas taimpartneri poolse süsiniku transpordi piiramise kaudu või tuleneda passiivsest reaktsioonist, mis on tingitud vähese seenele vajamineva süsiniku olemasolust. Siiski rõhutavad autorid, et üldjuhul ei kao seente kolonisatsioon juurtes täielikult ka sel juhul.

Konvalinková *et al.* (2015) läbi viidud katses mudelliigiga *Medicago truncatula*, mis inokulatsiooni korral sõltub peaaegu täielikult mükoriisa poolt transporditud fosforist, leiti nii lühiajalise kui ka pikaajalise varjutamise mõjusid fosfori omastamisele erinevatel intensiivsustasetel. Lühiajalisel varjutamisel suurenes fosfori kogunemine juurtes, tõenäolistelt seente poolt koloniseeritud struktuurides, kuid edasine transport võrsetesse langes kiiresti. Olulisi muutusi üldises juurte kolonisatsioonis ei olnud. Samas pikaajalisel varjutamisel olid drastilisemad efektid:

vähemalt 50% varjutuse korral vähenes hüüfide ja vesiikulite arv juures oluliselt, biomassi tootmine vähenes ja toimusid üldised struktuursed muutused taime ehituses, osutades tugevale süsiniku defitsiidile. Kõige tugevama varjutamisastme korral (90%) kadusid mükoriisse sümbioosi positiivsed mõjud, samuti langes oluliselt kõikide mükoriisastruktuuride hulk. Võrreldes mitterükoriisse kontrolliga oli fosfori omastamine ühe ühiku biomassi kohta siiski suurem kogu katse vältel, märkides taimepoolse süsinikuinvesteeringu kasu ning sümbioosi mõningast toimimist ka ekstreemsemates oludes. Autorid toovad ka välja, et varjutamise tingimustes toimusid taimel kompensatsioonimehhanismid nagu pikenemine ja lehe pindala suurenemine, mis ilmselt vähendasid varju negatiivset mõju taime kasvule. Samuti ilmselt kindel lävi, millest alates muutus vari taimekasvule limiteerivaks – kui valgusintensiivsus langes alla 35%, muutus sümbioos taimel kulukamaks kui saadav kasu.

Lisaks valguse tingitud süsiniku limitatsioonile leidsid Johnson *et al.* (2015) oma läbiviidud katses, et kui fosfori limiteeritud keskkonnades on mükoriisa funktsioon kaldumas mutualismile, siis valguse ja lämmastiku limitatsiooni tingimustes kaldub sümbioos kommensalismile või parasitismile. Autorid leidsid ka, mükoriisse sümbioosi ressursituru toimimisel on oluline aspekt ka mullaviljakus. Oluliseks hinnatakse ka Liebigi miinimumireeglit, mis aitab hinnata sümbioosi funktsiooni mõjutavaid faktoreid. Liebigi miinimumireegli puhul piirab taimede kasvu eelkõige see element, mille kontsentratsioon keskkonnas, võrreldes vajadusega, on väikseim. Samas Liu *et al.* (2015) tehtud katses mäestikuniidul leiti, et varjutamine ja väetamine eraldiseisvana ei mõjutanud juurte kolonisatsiooni, kuid nende kombineeritud mõju tõttu vähenes see neljakordselt.

Üheks võimalikuks mehhanismiks, mis võimaldaks juurte kolonisatsiooni säilimist AM seente poolt ka süsiniku limitatsiooni tingimustes, on kolonisatsiooni hoidmine tulevikuinvesteeringuna (Konvalinková & Jansa, 2016). Seen võib hoida kolonisatsiooni juures juhaks kui taime süsinikuvaru suureneb ja tulevikus on võimalik taas tõhustada ressursside vahetust. See mehhanism võib toimida looduslikes süsteemides eksisteeriva ühise mükoriisavõrgustiku (*common mycorrhizal networks*) näol. Nimelt võivad varjutatud taimed saada kasu ühistelt seenpartneritelt varjutamata taimede pakutud süsiniku tõttu. Ühised mükoriisavõrgustikud ühendavad omavahel sarnaseid või erinevaid liike seeneniidistiku kaudu ning taime süsinikuinvesteeringuga kaasneb ligipääs toitaineturule ja -vahetusele ning võimalus kasutada võrgustikku signaliseerimiseks, näiteks herbivooria korral (Wipf *et al.*, 2019).

Üldine konsensus varjutamise mõjude suhtes on, et vähenenud valguse kättesaadavus tingib taime poolt seenpartneritega vahetatava süsiniku limitatsiooni, vähendades seega sümbioosi kasutegurit taimel. Siiski on taimedel endal ka võime ja mehhanismid süsiniku limitatsiooniga hakkama

saamiseks ning seenpartneri säilitamiseks. Teatud lüvendit ületades võib kolonisatsioon juurtes siiski langeda, mis võib olla põhjustatud ainult valguse piirangust või on lisategureid. Konvalinková & Jansa (2016) hinnangul ei ole molekulaarsete ülekandemehhanismide täpse kirjeldamiseni võimalik tuvastada süsiniku ja fosfori vahetuse reguleerimismehhanisme.

1.3 Masinõpe ökoloogias ja programm AMFinder

Klassikalised mõõtmismeetodid on ökoloogias olulised elulooduse kirjeldamisel, interaktsioonide uurimisel ja ennustuste tegemisel. Samas on klassikaliste, inimhinnangul põhinevate meetodite kasutamine töömahukas ning digitaalsed meetodid võimaldavad andmeid kiiremini koguda ja analüüsida (Lürig *et al.*, 2021).

Masinõppel on potentsiaal kiirendada andmete annoteerimist, näha seoseid eelarvamusteta, eemaldada erinevatest inimhindajatest tingitud variatsioon ning osaline võimekus ka puuduliku andmestiku põhjal seoseid luua (Evangelisti *et al.*, 2021; Thessen, 2016).

Masinõpe on tehisintellekti alamvaldkond, mis võimaldab arvutitel automaatselt ülesandeid täita ja õpetada neid otseselt programmeerimata (Tampuu, 2020). Sügavõpe (*deep learning*) on masinõppe algoritm, mis kasutab inimese ajule sarnanevaid tehisnärvivõrke, et õppida suurest hulgast andmetest (Sügis, 2020). Programm AMFinder on loodud tehisintellekti ja sügavõpet kasutades ning on võimeline piltidelt tuvastama juuresiseseid arbuskulaarmükoriisete seente struktuure ja kvantifitseerida taimejuure kolonisatsiooni (Evangelisti *et al.*, 2021). AMFinder kasutab juhendatud masinõpet, et tuvastada vastavaid struktuure ka veel seni nägemata andmetel. Programmi tuumikuks on konvolutsioonilised närvivõrgud (*convolutional neural networks, CNN*), mis rakendavad sisendile korduvalt filtreid ning loovad väljundi, mis sisaldab erinevaid tunnuseid, näiteks pildil olevate objektide ääred (Tampuu, 2020). Rakendatavad filtrid ning nende väärtused on muudetavad, võimaldades tehisnärvivõrkudel ise õppida, milliseid tunnuseid pildilt otsida. Oluline on ka, et rakendatavates filtrites säilib informatsioon objektide paigutuse kohta ja suhted muude tunnustega pildil. Tehisnärvivõrkude viimaste kihtide hulka kuulub ka klassifikaator, mis otsustab tuvastatud objekti tüübi ehk klassi.

Masinõppe mudelite hindamiseks kasutatakse mõõdikuid täpsus (*accuracy*) ja valideerimistäpsus (*validation accuracy*). Täpsus on treenimise käigus kasutatud treeningandmete määramisel põhinev väärtus, kus mudeli õigesti hinnatud annoteringud jagatakse koguannoteringute arvuga. Täpsus on analoogne statistilise mudeli eristusvõimsusega. Valideerimistäpsus arvutatakse aga mudelile seni tundmata andmete annoteerimisel ning põhineb valideerimishulgal, millel on korrektsed annoteringud ning mis eraldatakse tavaliselt treeninghulgast. Valideerimistäpsus on analoogne statistilise mudeli ennustusvõimsusega, andes meile parema hinnangu mudeli võimekusele selle rakendamisel ehk kui hästi võiks mudel annoteringu klassi ennustada tundmatutel andmetel (Tampuu, 2020). Suur mudeli täpsuse ning valideerimistäpsuse suhe

tähendab üldjuhul, et mudel on ülesobitunud (*overfitted*) ehk on ära õppinud konkreetsed näited treeninghulgast ning ei suuda üldistusi luua (Tampuu, 2020).

Evangelisti *et al.*, (2021) on mükoriisse kolonisatsiooni tarbeks loonud kaks peamist närvivõrku, CNN1 ja CNN2. Programm saab sisendiks mikroskoobislaidipildi, mis segmenteeritakse ruudustikuks. CNN1 arvutab iga ruudu kohta tõenäosuse, kas see kuulub klassi „koloniseeritud juurelõik (M+)“ (*colonised root section*), „mittekoloniseeritud juurelõik (M-)“ (*noncolonised root section*) või „taust/pole juur/muu (X)“ (*background/not a root/other*). Peale CNN1 hinnangut on võimalik kasutada „M+“ ruutudes CNN2 võrku, mis ennustab arbuskulite (A), vesiikulite (V), hüüfide (IH) ja *hyphopodium*-ide (H) esinemist. Mõlema närvivõrgu struktuur ja ülesehitus on sarnane, erinedes viimastes kihtides. Mõlemast närvivõrgust on algteenitud ka versioon (CNN1v2 ja CNN2v2), mille puhul kasutati andmete täiustamist, pöörates pilti horisontaal- ja vertikaalteljel ning muutes suvaliselt pildi heledust, värvide pööret (*inversion*), küllastust (*saturation*) ja pildi tooni.

Evangelisti *et al.* (2021) kasutatud CNN1 algteenimise andmestik koosnes 32-st tubakataime *Nicotiana benthamiana* juurteslaidide pildist, millest 5 olid saadud skanneriga ning 27 tehtud mikroskoobiga. CNN2 andmestik koosnes 55-st pildist, millest kõik olid tehtud mikroskoobiga. Antud andmestiku hinnangud (annotatsioonid) vaadati käsitsi üle ning parandati alla 0,5 tõenäosusega hinnangud korrektseks. Mõlemat närvivõrku treeniti 100 epohhi. Testimise järel saadi CNN1v1 mudeli klassi mükoriisaga juur (M+) täpsuseks 98%, klassi mükoriisata juur (M-) 96% ning tegemist pole juurega (X) 98%. AMFinderi üks eeliseid on võimalus parandada närvivõrkude hinnangu täpsust spetsiifiliste slaidide peal, kasutades oma andmestikku.

2. Metoodika

2.1 Andmete allikad ja tehtud katse ülevaade

Varjutamise ja lõikamise mõju mudelliigiks katses oli keskmine värihein (*Briza media*), mis pärines 11-st rohumaa paarist Lääne-Eestis, kus olid paaris traditsioonilise karjatamisega ning lähedalasuvad hüljatud karjamaad. Igast asukohast võeti katsesse 10 genotüüpi ning need istutati 2-liitristesse pottidesse. Taimed istutati steriliseeritud liiva ja kõikidest asukohtadest korjatud mulla segusse, millele lisati taime asukohast pärit mullamikroobide inokulaat (kodumuld) või inokulaat kõrvalasuvalt rohumalt, mis oli kontrastse majandamisviisiga (võõras muld). Iga *B. media* populatsiooni ja mullainokulaadi kombinatsiooni kasvatati kolme töötluse all. Karjatamist simuleeriti taimede maapealsete osade lõikamisega ja varjutamist paigutades potid varjutelkidesse, mille peale oli paigutatud fooliumribadega kangas, mis peegeldas tagasi 75% valgusest. Varjutustelkidesse paigutatud taimede lõikamist ei toimunud. Kolmandaks töötluseks oli karjatamise lakkamise simulatsioon, kus taimede lehti kasvuperioodi jooksul ei lõigatud. Kokku koosnes katse 132 potist, 44 karjatamise simuleerimise töötlusega, 44 varjutamise töötlusega ning 44 karjatamise puudumise töötlusega.

Suve jooksul lõigati karjatamise simuleerimise töötluses taimi kaks korda, esimene kord 4 nädalat katse algusest, teine kord 4 nädalat esimesest lõikamisest. Taimed lõigati kuni 3 sentimeetri kõrguseks mõlemal korral. Katse võeti maha 14 nädala möödumisel algusajast. Igast potist võeti mullaproov juurte eraldamiseks.

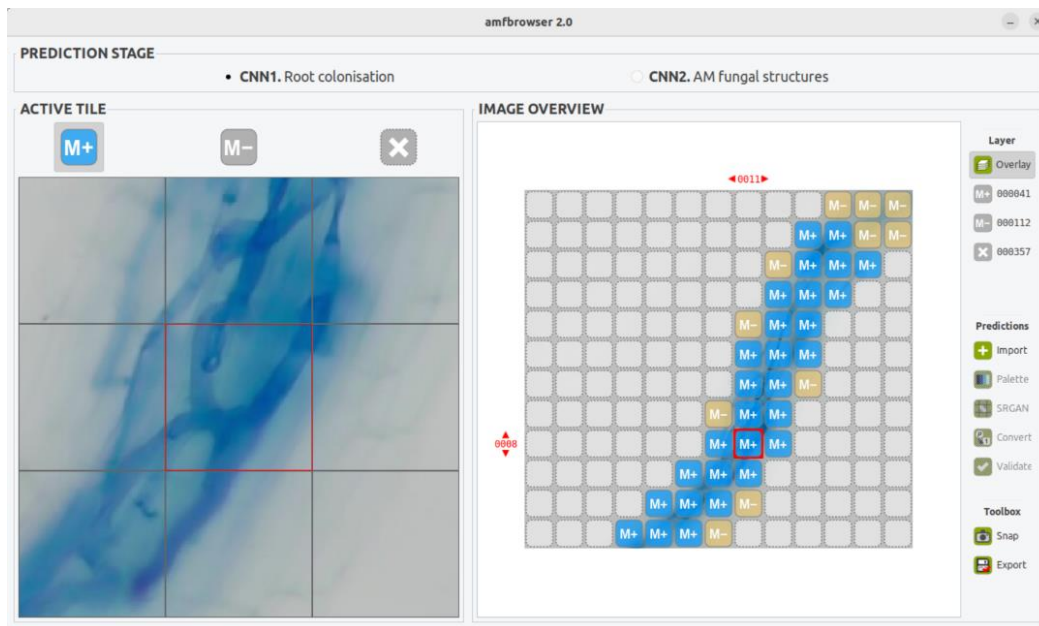
Mullaproovist pärinevad juured pesti veega puhtaks ning AM seente kolonisatsiooni määramiseks värviti sinise tindi ja äädika lahuses vastava meetodiga (Vierheilig *et al.*, 1998). Seejärel hoiti juuri ClearSee lahuses üks minut, et parandada mükoriisastruktuuride nähtavust (Kurihara *et al.*, 2015). Pärast töötlemist monteeriti juured glütserooliga klaasslaididele.

Kolonisatsiooni määramine teostati kahel viisil: a) klassikalise meetodiga, mis hõlmab juurelõikudes arbuskulite, hüüfide ja vesiikulite visuaalset hindamist valgusmikroskoobiga (McGongile *et al.*, 1990) ja b) programmi AMFinderiga, mis kasutab konvolutsioonilisi närvivõrke ja sügavõpet slaidipiltidelt AM kolonisatsiooni hindamiseks (Evangelisti *et al.*, 2021). Programmi katsetamiseks tehti igast slaidist kümnest suvalisest kohast pilti, kokku saadi 1320 pilti. Pildistamiseks kasutati mikroskoobile monteeritud Zeiss Axiocam 208 color kaamerat.

2.2 AMFinderi kasutamine

AMFinderi installimiseks laeti AMFinderi lähtekood ja andmed alla GitHubist (<https://github.com/SchornacklabSLCU/amfinder>). Programmi jooksumiseks kasutati WSLi (*Windows Subsystem for Linux*) ning käitati kaustas *amf* installimiskripti *install.sh*. Programmi kasutamiseks tuleb eelnevalt WSL käsuviibaga navigeerida kausta *amf* ning sisestada soovitud käsud. Tuli ka modifitseerida algset faili *amfinder_convert.py*, kuna täpselt tuvastamata vea tõttu ei suutnud skript importida vajalikku Pythoni teeki *imagesize*. Muudatusena lisati faili funktsioon *imagesizeget*, mis tagastab failiteena etteantud pildi kõrguse ja laiuse ning kommenteeriti välja teegi *imagesize* impordikäsk. AMFinder võimaldab ka eelnevalt valmiskirjutatud skripti käitamist käsurealt, vajades seeläbi vähem kasutaja sekkumist.

Algselt tuleb iga pildi peal jooksumata *predict* käsku, mis segmenteerib pildi ruutudeks ning annab tõenäosuse iga ruudu kohta, kas see kuulub klassi M+ (mükoriisiga juur), M- (mükoriisata juur) või X (tegemist pole juurega). Ruudustiku suurust saab kasutaja vajadusel ise muuta, vaikeväärtusena on ruudu küljepikkus 126 pikslit. Käesoleva uuringu andmestiku piltidelt loodi ruudustik laiusega 30 ja kõrgusega 17 ruutu ehk iga pildi kohta oli 510 ruutu. Programm loob iga pildi kohta ZIP formaadis kausta, kuhu salvestab hinnangute tulemused TSV (*Tab-separated values*) formaadis faili, milles on tulbad rida (*row*), veerg (*col*), Y, N ja X. Väli Y tähistab klassi M+ tõenäosust, N klassi M- ja X klassi *non-root tile* tõenäosust. Väljad rida ja veerg tähistavad segmendi asukohta ruudustikus. Seejärel tuleb käitada käsku *convert*, mis muundab hinnangud annoteringuteks ning salvestab tulemused iga pildi kohta samasse ZIP kausta *col.tsv* faili samade väljadega. Nii *predict* kui ka *convert* käskude tulemusi saab vaadata graafilise liidesega *amfbrowser* (Evangelisti *et al.*, 2021; Joonis 3).



Joonis 3. Kuvatõmmis amfbrowser graafilisest liidesest (Evangelisti *et al.*, 2021).

Kolonisatsiooni protsentuaalseks esitamiseks kasutati tarkvara R v4.4.0 (R Core Team, 2024) ja RStudio (Posit team, 2024). Selleks koostati R skript, mis loeb kaustas olevad ZIP failide nimed listi ning pakib igast kaustas lahti *col.tsv* faili ning loeb selle väärtused koos nimedega teise listi. Seejärel leitakse pildil ruutude summa, mis annoteeriti juureks (väljad Y ja N), ja ruudud (väli Y), mis hinnati olevat koloniseeritud. Väärtused omavahel jagades saab leida kolonisatsiooni protsendi ning see salvestatakse CSV (*Comma separated values*) formaadis faili.

AMFinderi treenimise algandmestiku loomiseks kasutati 6 juhuslikult valitud mikroskoobislaidi juurepreparaati, millest tehti mikroskoobiga 53 pilti. Õigete annoteeringute loomiseks rakendati algset CNN1v2 mudelit piltide peal, seejärel, kasutades amfbrowserit, parandati käsitsi valesti hinnatud ruudud. Saadud andmestik jagati viieks hulgaks ($v1, v2, v3, v4, v5$), mida tutvustati aditiivselt närvivõrgule, suurendades igal etapil mudeli treenimiseks kasutatavat andmekogust. Aditiivne lähenemise kasuks otsustati arvuti mälu mahu piirangu tõttu, kuna korraga ei saanud kogu andmestikku treenimiseks mällu laadida. Samuti sülearvuti kasutamisega mudeli annotatsioonide loomiseks sai katsetada mudeli käitamist realistlikemates oludes. Algmudeliks võeti CNN1v2, mida treeniti hulgaga $v1$, saades mudeli CNN1v2_v1. Valik jooksutatud käskudest on leitavad lisades (Lisa 2).

Treeniti kümme mudelit, kus ühes kasutati andmete augmenteerimist (CNN1v2_0105) ning teises mitte (CNN1v2_1905). WSL käsuviiba väljund mudelite treenimise ajal salvestati samuti, vahepealsed mudelid salvestati nimeformaadiga CNN1v2_(mudeli versioon)_(treenimise kuupäev) (Lisa 2). Mudeli CNN1v2_1904 täpsus peale igat treenimisjärku tõusis üle 80%, kuid

peale viimast järku jäi madalaks (Tabel 1). Mitteaugmenteeritud andmetega mudelite nimetamiseks kasutati formaati Finder_(mudeli versioon) ning augmenteeritud andmetega formaati FinderAug_(mudeli versioon).

Tabel 1. Mudeli täpsus (*accuracy*) ja valideerimistäpsus (*validation accuracy*) pärast iga järjestikuse treeninghulgaga treenimist

Järk	<i>Finder_v1</i>	<i>Finder_v2</i>	<i>Finder_v3</i>	<i>Finder_v4</i>	<i>Finder_v5</i>
Mudeli täpsus	0,8238	0,8769	0,8031	0,8234	0,1639
Valideerimistäpsus	0,8238	0,8662	0,7904	0,8678	0,0402

2.3 Andmeanalüüs

Treenitud mudelite hinnangud saadi jooksutades käsku *predict* ja *convert* ning seejärel arvutati kolonisatsiooniprotsent. Saadud tulemusi võrreldi käsitsi annoteeritud piltidega, saades andmestiku, kus iga pildi kohta oli olemas korrektselt annoteeritud kolonisatsioon ning kümne mudeli kolonisatsiooni hinnang. Korrelatsiooni testimiseks kasutati Spearmani korrelatsiooni (Joonis 4, Tabel 2). Samad testid tehti ka augmenteeritud andmetega mudeli peal (Joonis 5, Tabel 3).

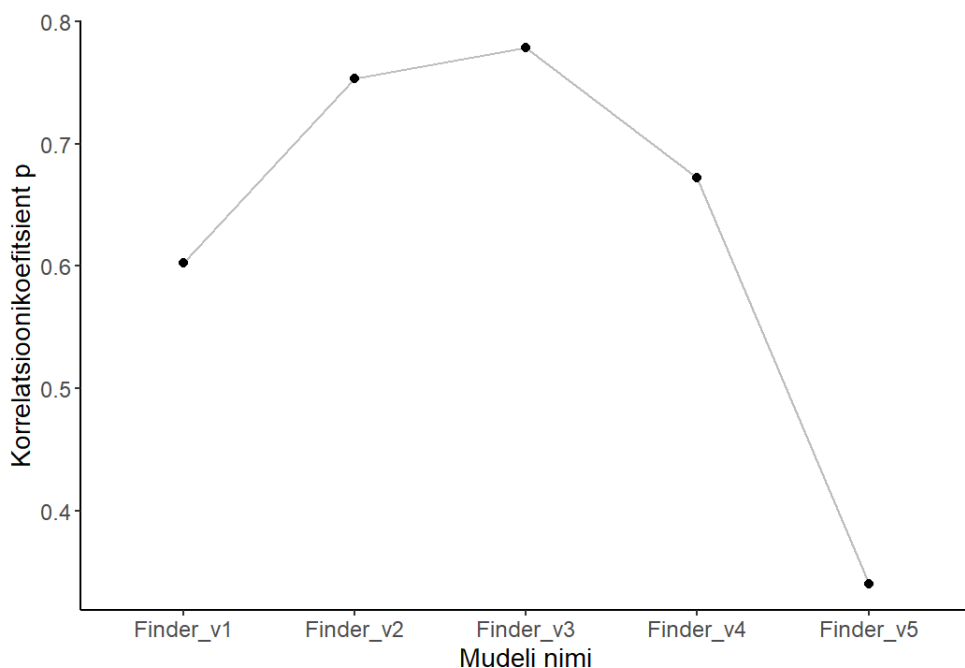
Slaidi koguhinnangu saamiseks keskmistati slaidi iga pildi AMFinderi mudeli hinnangud ning võrreldi klassikalise meetodi slaidipõhiste hinnangutega (Joonis 6).

Karjatamise ja varjutamise mõjude hindamiseks AM kolonisatsioonile kasutati lineaarset segamudelit (*lmer*) R paketist lme4 (Bates *et al.*, 2015). Sõltuv faktor oli AM kolonisatsiooni hinnang (logit-transformeeritud), fikseeritud faktorid olid mulla inokulaat ja töötlus. Juhuslike faktoritena lisati mudelisse taimede ja mulla inokulaadi päritolu asukohad

3. Tulemused

3.1 AMFinderi mudelite tulemused

Korrelatsioon paranes esimese mudeli puhul kolmanda versioonini, kuid pärast treeningandmestiku hulga $v4$ lisamist korrelatsioon korrektsete annotatsioonidega vähenes (Joonis 4; Tabel 2).



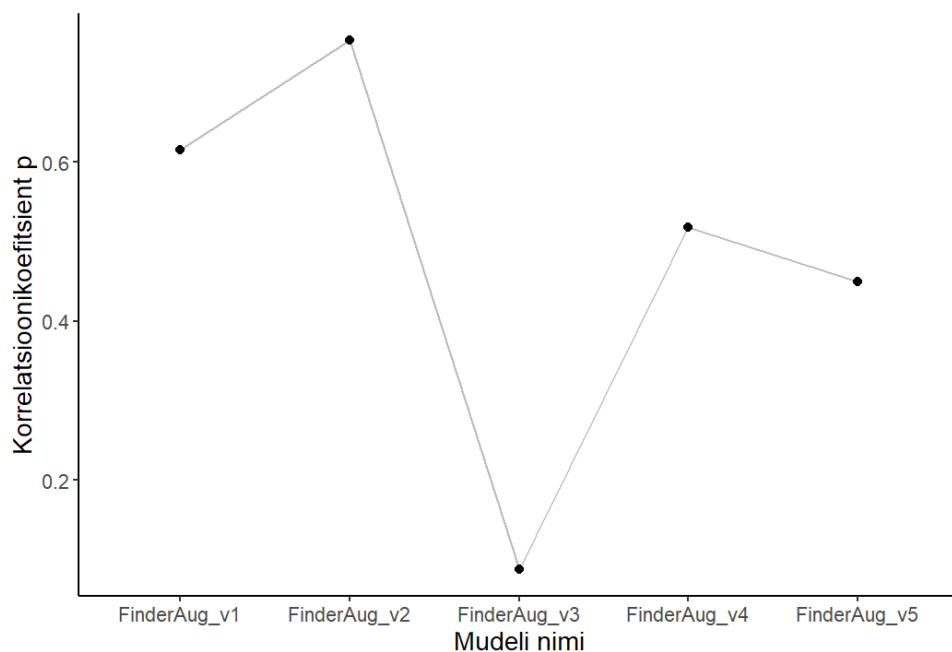
Joonis 4. Korrelatsioon CNN1v2_1904 erinevate versioonide ning korrektsete annotatsioonide (käsitsi annoteeritud) AM seente kolonisatsiooni vahel.

Tabel 2. Korrelatsioonikordajad AMFinder CNN1v2_1904 mudeli versioonide ja korrektsete annotatsioonide vahel. S statistik on vigade ruutude summa Spearmani korrelatsioonitestis.

Mudel	P väärtus	S statistik	Korrelatsioonikoefitsient ρ
Finder_v1	$1,78 \cdot 10^{-6}$	9851,90	0,60
Finder_v2	$7,75 \cdot 10^{-11}$	6124,06	0,75
Finder_v3	$6,98 \cdot 10^{-12}$	5500,29	0,78
Finder_v4	$3,62 \cdot 10^{-8}$	8137,93	0,67
Finder_v5	$1,34 \cdot 10^{-2}$	15436,41	0,34

Teise mudeli korral, kus kasutati andmete augmenteerimist, paranes korrelatsioon kuni andmehulga $v3$ lisamiseni. Pärast andmehulka $v3$ korrelatsioon korrektsete annotatsioonidega

vähenes oluliselt, kuid peale andmehulka v4 korrelatsioon taas paranes. Pärast andmehulka v5 vähenes korrelatsioon uuesti (Joonis 5; Tabel 4).

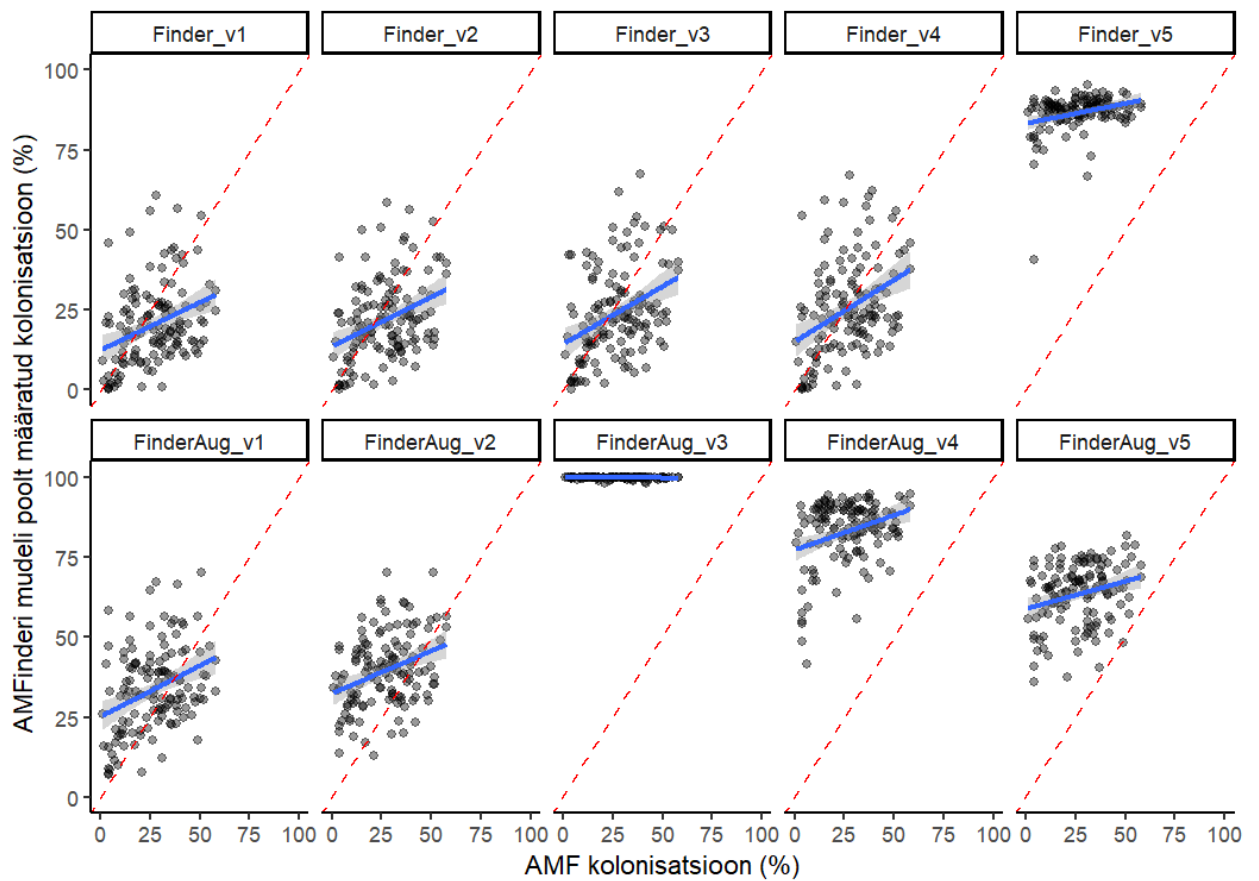


Joonis 5. Korrelatsioon CNN1v2_0105 erinevate versioonide ning korrektsete annotatsioonide kolonisatsiooni vahel.

Tabel 3. Korrelatsioonikordajad AMFinder CNN1v2_0105 mudeli versioonide ja korrektsete annotatsioonide vahel.

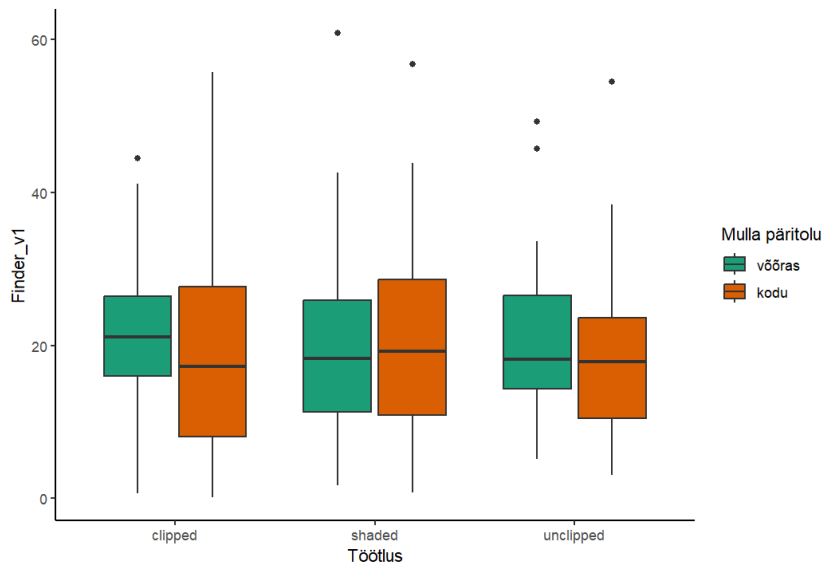
Mudel	P väärtus	S statistik	Korrelatsioonikoefitsient ρ
FinderAug_v1	$9,68 \cdot 10^{-7}$	9553,69	0,61
FinderAug_v2	$7,75 \cdot 10^{-11}$	6124,06	0,75
FinderAug_v3	0,5358	22647	0,09
FinderAug_v4	$7,12 \cdot 10^{-5}$	11956,9	0,52
FinderAug_v5	0,00074	13657,7	0,45

Võrreldes klassikalise meetodi kolonisatsiooni hinnangut ning AMFinder'i mudeleid on näha, et AMFinder süstemaatiliselt alahindab kõrget kolonisatsiooni ning ülehindab madalat kolonisatsiooni (Joonis 6).



Joonis 6. AMFinderi mudelite kolonisatsiooni hinnangud võrreldes klassikalise meetodi AMF hinnanguga. Sinise joonega märgitud lineaarne seos, hall pind tähistab trendide standardviga. Punase punktiirjoonega on näidatud 1:1 seose joon.

Dispersioonanalüüs AMFinderi mudelite hinnangutele ei olnud statistiliselt olulised ($P > 0,05$). Mudelid ei suutnud tuvastada erinevusi töötluste vahel ning töötluste kolonisatsiooni hinnangute keskmised jäid sarnaseks (Joonis 7).



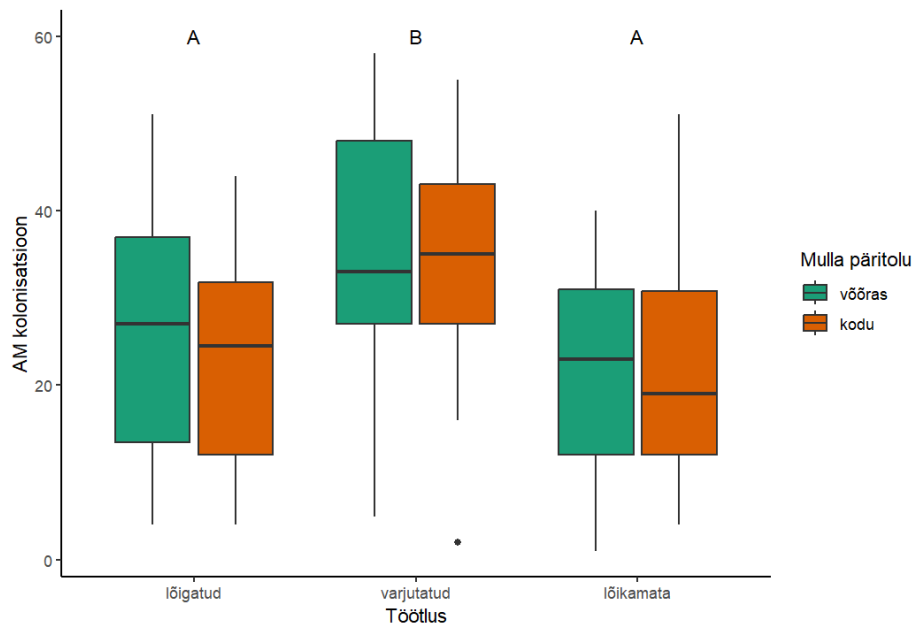
Joonis 7 AMFinderi mudeli Finder_v1 arbuskulaarmükoriisne kolonisatsioon taimede juurtes lõigatud, varjutatud, lõikamata töötluste ja võõra (roheline) ning kodumulla (oranž) päritolu võrdluses. Näidatud on grupi keskmine väärtus (keskmise joon), 1. ja 3. kvartiil (kastid) ning 1,5-kordne kvartiilivahemik (vurrud) ja erandid (punktid). Dispersioonanalüüs ei olnud statistiliselt oluline.

3.2 Tulemused klassikalise meetodiga

Katse töötlus mõjutas AM kolonisatsiooni statistiliselt oluliselt (Tabel 4). Varjutustöötluste korral oli taimejuurte kolonisatsioon AM seente poolt statistiliselt oluliselt kõrgem kui lõigatud ja lõikamata töötlustes (Joonis 8).

Tabel 4. Dispersioonanalüüsi tulemused klassikalise kolonisatsioonimeetodiga määramisel

	Vabadusastmed	F väärtus	P väärtus
Töötlus	2	8,28	0,0004
Inokulaat	1	0,002	0,968
Töötlus*inokulaat	2	0,47	0,624
Jääkhajuvus	123		



Joonis 8. Arbuskulaarmükoriisne kolonisatsioon taimede juurtes lõigatud, varjutatud, lõikamata töötluste ja võõra (roheline) ning kodumulla (oranž) päritolu võrdluses. Näidatud on grupi keskmine väärtus (keskmine joon), 1. ja 3. kvartiil (kastid) ning 1,5-kordne kvartiilvahemik (vurrud) ja erandid (punktid). Erinevad tähed töötluste kohal tähendavad statistiliselt olulist erinevust.

4. Arutelu

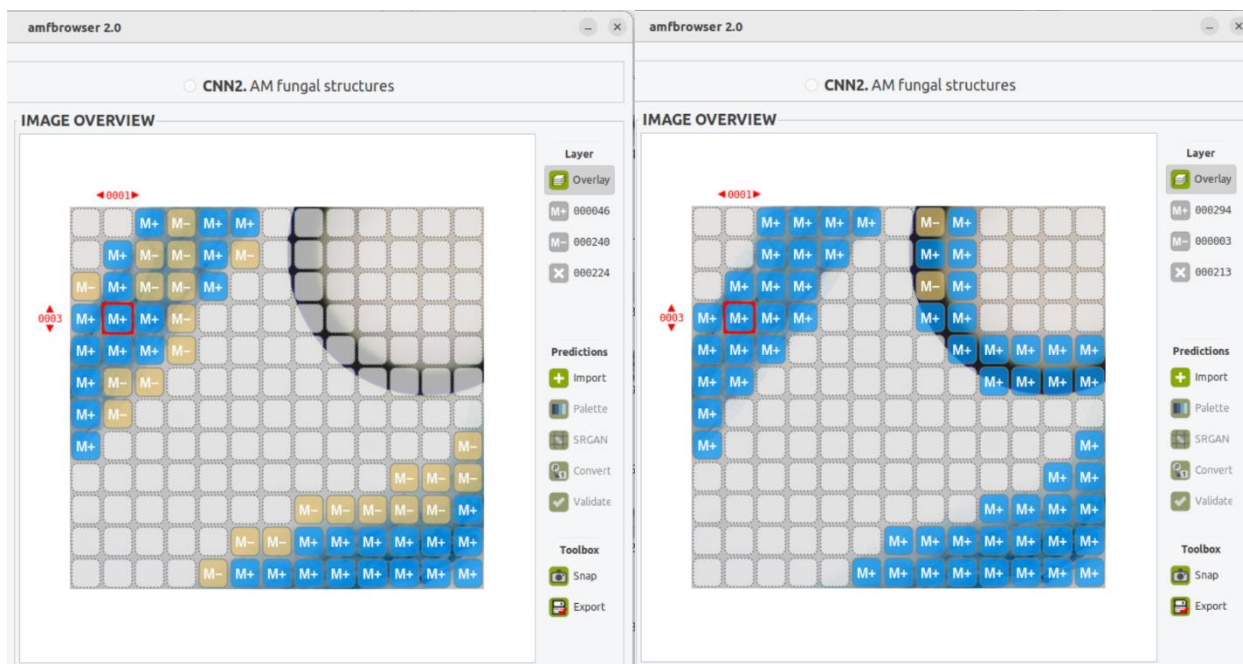
Andmeanalüüsisist selgus, et varjutamisel oli mõju AM kolonisatsioonile taimejuurtes, samas kui karjatamise simuleerimise mõju ei olnud statistiliselt oluline. Oluline on siinkohal kontekst, nagu on välja toonud oma analüüsis Faghihinia *et al.* (2020a). Tehtud katses ei olnud mudelliik *B. media* oma tavapärasest koosluses, kus tuleb konkureerida ka teiste liikidega, mistõttu oleks mõju mudelligile võinud selgemalt avalduda. Sarnaselt Heyde *et al.* (2017) ja Faghihinia *et al.* (2020b) katsetele ei muutunud karjatamise simuleerimise tõttu mükoriisne kolonisatsioon juurtes antud katses oluliselt. Karjatamise mõju arbuskulaarsele mükoriisale pole arvatavasti lineaarne ning sõltub karjatamise sagedusest ja intensiivsusest. Antud katses tehtud lehtede lõikamine kahel korral suve jooksul simuleerimaks karjatamist polnud ilmselt piisavalt intensiivne, et mõju avalduks. Samuti on *B. media* aeglase kasvuga ning võimaliku mõju avaldumiseks oleks vaja uurida seda pikemal ajaskaalal. Teisi AM seente tunnuseid täpsemalt ei uuritud. Sarnaselt siiaamaani tehtud uuringutele ja analüüsidele on karjatamise mõjud ebaselged ning täpsema arusaama loomiseks on vaja uurida korraga rohkem tunnuseid ning pikaajalisemalt (Faghihinia *et al.*, 2020a; Heyde *et al.*, 2017).

Varjutamise mõju oli statistiliselt oluline ning mõlema mullainokulaadi puhul oli varjutatud taimede AM seente kolonisatsiooni keskmine kõrgem kui lõigatud ja lõikamata töötluste keskmised Siiaamaani tehtud katsed näitavad AM seente kolonisatsiooni vähenemist tugeva varjutuse olemasolul (Konvalinková *et al.*, 2015). Samuti toob valguse limitatsioon kaasa süsiniku limitatsiooni, mistõttu võiks eeldada kolonisatsiooni vähenemist (Konvalinková & Jansa, 2016), mida siiski ei ilmnenud. Varjutamine ei olnud antud katses arvatavasti veel piisav, et mõjutada negatiivselt taimekasvu ja piirata süsiniku allokatsiooni AM seentesse. Võimalik, et varjutamata taimedele oli keskkond stressirohke, mis põhjustas suuremat süsiniku limitatsiooni ja madalamat AM seente kolonisatsiooni võrreldes varjus kasvanud taimedega. Seda toetab uurimus, kus enamik vaadeldud rohumaade taimeliike kasvasid suuremaks poolvarjus, seda eriti aeglase kasvustrateegiaga liikide puhul (Semchenko *et al.*, 2012). Antud katses kasutati 75% varju ning mudelliik *B. media* on aeglase kasvustrateegiaga. Varjutatud taimedel olid soodsamad kasvutingimused, nagu parem kaitse UV kiirguse ja tugeva tuule eest, võis see võimaldada taimedel panustada rohkem ressursse mükoriispartneri ülalpidamisse (Semchenko *et al.*, 2012).

Tulemustest selgus, et AMFinderi tuvastatud kolonisatsiooni põhjal ei avaldanud töötlusted statistiliselt olulist mõju, samas kui klassikalisel meetodil see nii oli. AMFinder hindas süstemaatiliselt kõrget kolonisatsiooni madalamaks ja madalat kolonisatsiooni kõrgemaks, mis

tõenäoliselt tasandas varjutamise töötluse erinevuse. Samuti on tehtud pildid ainult juhuslik väljavõtte slaididest ning ei pruugi olla esinduslikud või piisavad robustse kolonisatsiooni hinnangu saamiseks. Samas põhineb ka klassikaline meetod vaid saja juhusliku vaatepunkti registreerimisest, mis ei pruugi sarnaselt tagada robustset hinnangut kogu slaidi ega juurestiku kohta. Seega on võimalik, et suurema andmekoguse korral oleks dispersioonanalüüsi tulemused erinevad. Närvivõrkude üks olulisi komponente on treeningandmete hulk ja väheste saadavalolevate treeningandmete puhul kasutatakse tihti õppimise ülekandmist (*transfer learning*) teistest mudelitest (Agrawal *et al.*, 2019). Üks levinud mudel objektide tuvastamiseks on Ultralytics Yolov8, mis on oluliselt keerulisema struktuuriga ning algtreenitud COCO andmestiku peal (Jocher *et al.*, 2023). Evangelisti *et al.* (2021) loodud treeningandmestik ning sinna lisatud uurimuses kasutatud andmestik ei olnud piisavad, et leida inimhindajaga võrreldavaid statistiliselt olulisi erinevusi töötluste vahel. On ka võimalik, et AMFinder'i närvivõrk pole piisavalt keerulise ülesehitusega ning vajaks täiustamist, et tuvastada AM seente kolonisatsiooni paremini, kasutades õppimise ülekandmist. Samas keerulisema mudeliga tekib oht ülesobitamiseks (*overfitting*) ning mudeli treenimisele kuluv ajahulk suureneb.

AMFinder'i mudeli korrelatsiooni vähenemine korrektsete annotatsioonidega ning täpsuse suur muutumine pärast treenimishulka $v3$ tulenes arvatavasti treenimisandmestiku ülesehitusest ning treenimise aditiivsest lähenemisest. Treenimise aditiivne lähenemine tulenes mälu piirangust – korraka ei olnud võimalik anda ette kogu treeninghulka, mistõttu jagati see alamhulkadeks. Hulgas $v3$ ilmusid piltidele õhumullid, mis ajasid mudeli juba väljatreenitud neuronid segadusse, põhjustades nende liigitamist valesse klassi (Joonis 9). Varasemalt õpitud pildiomaduste järgi võivad õhumullid mudeli jaoks sarnaneda mükoriisa- ning juurestruktuuridele, mida rakendades liigitatigi palju ruute valesse klassi. Tõenäoliselt toimus suurem närvivõrgu neuronite kaalude väärtuste muutmine peale antud andmehulka, kuid jäädes eelnevale piisavalt sarnaseks, mistõttu said järgnevad mudelid tulemust parandada. Võis toimuda ka alasobitamine (*underfitting*), mille tingib, et mudel ei pruugi olla piisavalt võimas, et õppida ära keerulisi tunnuseid ning nende põhjal üldistusi tegema. Uuteks liiga keerulisteks tunnusteks võisid olla õhumullid.



Joonis 9. Väljavõte Amfbrowserist (Evangelisti *et al.*, 2021). Vasakul pildi EXP2_37_5.jpg käsitsi anoteeritud versioon ning paremal FinderAug_v3 hinnang. M+ tähistab mükoriisaga, M- mükoriisata juuresegmenti.

Mudeli täpsuse parandamiseks ning paremate tulemuste saavutamiseks tuleks kaaluda järgnevat:

1. Hankida rohkem treeningandmeid. Kui see pole võimalik, siis kasutada andmete augmenteerimist ning katsetada erinevaid viise ning meetodeid. Pildi kõrguse ja laiuse muutmine võib parandada mudeli üldistusvõimet ja vähendada ülesobitamist (*overfitting*) (Thanapol *et al.*, 2020).
2. Anda ette maksimaalne võimalik treeninghulk, et laiendada närvivõrgule korraga nähtavat andmestikku ning seejärel aditiivselt suurendada treeninghulka. Võimaluse korral suurendada treenimiseks kasutatava arvuti mäluhulka, arvutusvõimsust ja kasutada masinõppele paremini optimeeritud riistvara (üldiselt graafikakaardid). Tasub kaaluda ka treeningandmete tihendamist. Lisaväärtust annaks slaidist mikroskoobiga tehtud tervikpilt, mitte ainult 10 vaadet. Antud lähenemine võimaldaks mudelil näha slaidi kui tervikut ning sellelt otsida AM tunnuseid ning need ära õppida.
3. Treenida mudelit kauem ehk rohkem epoche. Kui eelnevad lähenemised rahuldavaid tulemusi ei too, tuleks kaaluda mudeli keerukuse tõstmist või kasutada baasmudelina keerulisemat mudelit, näiteks Ultralytics Yolov8, ning kohandada see arbuskulaarse mükoriisa tuvastamise jaoks ja rakendada õppimise ülekandmist (*transfer learning*).

Töös kasutatud programm AMFinder on võimekas ning selle täiendatud versiooni loomine võimaldaks kiirendada andmete analüüsimist ning seoste loomist ökoloogias, väheneks ka kuluv inimtöötundide arv. Slaididest piltide tegemine võttis aega alla kahe tööpäeva ehk maksimaalselt 16 tundi. Mudeli versiooni treenimine kõigi viie treeninghulgaga optimeeritud riistvaraga (graafikakaart) võttis aega ligikaudu 1 tund. Sama palju aega läks ka ühel mudelil 1320 pildi kolonisatsiooni hindamiseks ehk ligikaudu 2 tunniga oli olemas mudel ning uued hinnangud. Kokku oli ajakulu slaidi pildistamisest hinnanguteni kuni 18 tundi. Klassikalise meetodi puhul kulus mikroskopeerimisele 25 tundi. Eri mudelite treenimine, omavaheline võrdlus ja katsetamine võtavad samuti aega, hea mudeli olemasolul on kolonisatsiooni hindamise protsess aga kiire. AMFinder programmi kasutamine ei ole kõigile operatsioonisüsteemidele optimeeritud, nõudes seetõttu vigade leidmiseks lisa-aega, raskendades antud lahenduse laiemat kasutamist. Samas kulub algajal kolonisatsiooni määramisel mikroskopeerimisele samuti tunduvalt rohkem aega. Programmi AMFinder kasutades kaoks ära erinevate kolonisatsioonihindajate subjektiivsusest tingitud variatsioon andmestikus. Tulevikuvaatena on masinõppel kindel potentsiaal leida rakendust AM struktuuride tuvastamises ning piisavate treeningandmete olemasolul ka võibolla luua uusi seoseid puudulike andmehulkade korral.

Kokkuvõte

Töös uuriti karjatamise ja varjutamise mõju arbuskulaarset mükoriisat moodustavate seente kolonisatsioonile taimejuurtes. Arbuskulaarne mükoriisa on vanim mükoriisatüüp ning antud sümbioos varieerub mutualismist parasitismini. Antud sümbioos sobitub ka bioloogilise turu teooria konteksti, kus taimel ning seenpartneril on turu üle kahesuunaline kontroll ja vahetatavateks ressursideks on taime poolt kogutud süsinik ning seene poolt fosfor. Turuolukorda mõjutavad nii karjatamine kui ka varjutamine, kuna nad põhjustavad taimetele fotosünteesiks vajaliku lehepinna või kättesaadava valguse vähenemist ning seeläbi süsiniku limitatsiooni. Nende mõjud arbuskulaarsele mükoriisale varieeruvad uuringute vahel, karjatamise puhul pole uurimuste vahel ühist konsensust. Olulised faktorid, mis ilmselt karjatamise mõju AM sümbioosile moduleerivad, on karjatamise intensiivsus ning ajaline skaala. Varjutamise puhul on arusaam, et kindlat lävendit ületades võib AM seente kolonisatsioon juurtes langeda.

Mõjude suuna hindamiseks tehti katse, kus objektiks oli keskmise väriheina (*Briza media*) genotüübid 22-st eri populatsioonist, mida kasvatati 10-st genotüüpi koosnevate gruppidega kas koos samalt alalt pärit mulla (kodumuld) või lähedalasuva erineva majandamisviisiga rohumaa mulla inokulaadiga. Lisaks kasvatati iga taimegrupp kas simuleeritud karjatamise töötuse all, varjutamise töötuse all või karjatamise puudumise töötuse all. AM seente kolonisatsiooni uurimiseks võeti igast potist juureproov, mis värviti, kasutades äädika ning tindilahust. Juured monteeriti slaididele ning kolonisatsiooni määramiseks kasutati kahte meetodikat: klassikalist inimhindajat ning programmi AMFinder.

AMFinder on masinõppeprogramm, mis põhineb konvolutsioonilistel närvivõrkudel ning on võimeline piltidelt tuvastama värvitud mükoriisastruktuure. Närvivõrgu treenimiseks tehti kuuest slaidist pildid ning loodi viis treeninghulka, millega treeniti kümme mudelit, kasutades aditiivset lähenemist andmehulga suurendamiseks mudelil. Mudeli hindamiseks kasutati Spearmani korrelatsioonikoeffitsienti, võrreldes pildipõhiseid mudeli annotatsioone treeningandmestiku annotatsioonidega. Seejärel lasti igal mudelil hinnata igast slaidist tehtud 10 pildil AM seente kolonisatsiooni ning keskmistati nende põhjal slaidi kolonisatsiooni hinnang. Töötuste mõju hindamiseks kasutati dispersioonanalüüsi, mille tulemusel selgus, et klassikalise meetodiga tehtud hinnangute põhjal oli varjutamise töötusel statistiliselt oluline mõju AM seente kolonisatsioonile. Samu tulemusi AMFinder'i mudelite hinnangud ei leidnud.

Varjutatud taimedele olid keskkonnatingimused soodsamad võrreldes karjatamise simuleerimise taimedega, tingides keskmisest kõrgema mükoriisa kolonisatsiooni. Varjus taimed said panustada

rohkem ressursse seenpartneri ülalpidamise. Karjatamise simuleerimine ei olnud piisavalt intensiivne, et avalduks statistiliselt oluline mõju mükoriisa kolonisatsioonile.

Programmi AMFinder'i tulemuste parandamiseks tuleks suurendada treeningandmete hulka, võimaluse korral teha slaidist kogupilt ning suurendada treenimiseks kasutatava arvuti mälu kogust ja arvutusvõimsust. Masinõppe kasutamine võimaldaks tulevikus säästa aega.

Summary

This thesis studied the effects of grazing and shading on the colonisation of arbuscular mycorrhizal fungi in plant roots. Arbuscular mycorrhiza is the oldest type of mycorrhiza, which varies from mutualism to parasitism. In the context of the biological market theory, where plant and fungal partners have two-way control over the market and exchanged commodities are plant carbon and fungal phosphorus, arbuscular mycorrhizal symbiosis fits well. Both grazing and shading affect the market as they can cause a reduction in leaf area or available light and reduce plant photosynthetic capability, causing carbon limitation. Said effects on arbuscular mycorrhiza vary between studies. In the case of grazing, there is no consensus among studies. The intensity and temporal scale of grazing are important factors that likely modulate its effect on AM symbiosis. In the case of shading, it is understood that exceeding a certain threshold may cause a decrease in AM fungal colonisation.

An experiment was conducted where the subjects were genotypes of *Briza media* from 22 different populations, which were grown in groups consisting of 10 genotypes either with soil from the same site (home soil) or inoculum from a nearby grassland with different management. Grazing was simulated with clipping. Each group was grown either under clipping treatment, shading treatment or the absence of clipping treatment. To quantify AM fungal colonisation, root samples were taken from each pot, which were stained using vinegar and ink solution. The roots were mounted on slides, and two methods were used to determine colonisation: a classical human-based method and the AMFinder program.

AMFinder is a machine learning program built on convolutional neural networks, which can detect mycorrhizal structures from stained root images. For training the neural network, images were taken from six slides and five training sets were created, with which ten models were trained, using an additive approach to increase the dataset. The model was evaluated using the Spearman correlation coefficient and comparing predicted and training dataset annotations. Then, each model was used to predict AM fungal colonisation on 10 images taken from each slide, whole slide colonisation estimate was averaged. Analysis of variance was used to assess the effects of treatments; it was found that the shading treatment had a statistically significant effect on AM fungal colonisation based on estimates made by the classical method. Similar results were not achieved by the AMFinder models.

Shaded plants had more favourable conditions than plants with clipping treatment, resulting in higher average mycorrhizal colonisation. Plants in the shade were able to invest more resources

into maintaining their fungal partner. Clipping intensity was not enough to have a statistically significant effect on colonisation.

To improve the performance of AMFinder, the amount of training data should be increased. If possible, the whole slide should be imaged. Memory size and computational power should also be increased. The use of machine learning has the potential to save time in the quantification of AM fungal colonisation in plant roots.

Tänuavaldused

Soovin tänada oma juhendajaid, Marina Semtšenkot ja Siim-Kaarel Seppa, kes on mind palju aidanud lõputöö kirjutamisega, olles alati abivalmid ning toetavad. Samuti soovin tänada Evolutsioonilise taimeökoloogia töörühma mind kaasamast teadusesse ja töörühma tegemistesse, tunnen end hoituna. Lisaks tahan tänada oma kursusekaaslast ja sõpru, kes mind on saatnud bakalaureuseõpingutel.

Kasutatud kirjandus

- Agrawal, T., Gupta, R., & Narayanan, S. (2019). On Evaluating CNN Representations for Low Resource Medical Image Classification. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1363–1367. <https://doi.org/10.1109/ICASSP.2019.8682397>
- A.R., W., & S.J., O. (2001). Grasslands, grazing and biodiversity: Editors' introduction. *Journal of Applied Ecology*, 38(2), 233–237. <https://doi.org/10.1046/j.1365-2664.2001.00621.x>
- Ba, L., Ning, J., Wang, D., Facelli, E., Facelli, J. M., Yang, Y., & Zhang, L. (2012). The relationship between the diversity of arbuscular mycorrhizal fungi and grazing in a meadow steppe. *Plant and Soil*, 352(1–2), 143–156. <https://doi.org/10.1007/s11104-011-0985-6>
- Barto, E. K., & Rillig, M. C. (2010). Does herbivory really suppress mycorrhiza? A meta-analysis. *Journal of Ecology*, 98(4), 745–753. <https://doi.org/10.1111/j.1365-2745.2010.01658.x>
- Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1), 1–48. <https://doi.org/10.18637/jss.v067.i01>
- Brundrett, M. C., & Tedersoo, L. (2018). Evolutionary history of mycorrhizal symbioses and global host plant diversity. *New Phytologist*, 220(4), 1108–1115. <https://doi.org/10.1111/nph.14976>
- Chomicki, G., Kiers, E. T., & Renner, S. S. (2020). The Evolution of Mutualistic Dependence. *Annual Review of Ecology, Evolution, and Systematics*, 51(1), 409–432. <https://doi.org/10.1146/annurev-ecolsys-110218-024629>
- Evangelisti, E., Turner, C., McDowell, A., Shenhav, L., Yunusov, T., Gavrín, A., Servante, E. K., Quan, C., & Schornack, S. (2021). Deep learning-based quantification of arbuscular mycorrhizal fungi in plant roots. *New Phytologist*, 232(5), 2207–2219. <https://doi.org/10.1111/nph.17697>
- Faghihinia, M., Zou, Y., Chen, Z., Bai, Y., Li, W., Marrs, R., & Staddon, P. L. (2020a). Environmental drivers of grazing effects on arbuscular mycorrhizal fungi in grasslands. *Applied Soil Ecology*, 153, 103591. <https://doi.org/10.1016/j.apsoil.2020.103591>
- Faghihinia, M., Zou, Y., Chen, Z., Bai, Y., Li, W., Marrs, R., & Staddon, P. L. (2020b). The response of grassland mycorrhizal fungal abundance to a range of long-term grazing intensities. *Rhizosphere*, 13, 100178. <https://doi.org/10.1016/j.rhisph.2019.100178>

- Fellbaum, C. R., Mensah, J. A., Cloos, A. J., Strahan, G. E., Pfeffer, P. E., Kiers, E. T., & Bücking, H. (2014). Fungal nutrient allocation in common mycorrhizal networks is regulated by the carbon source strength of individual host plants. *New Phytologist*, *203*(2), 646–656. <https://doi.org/10.1111/nph.12827>
- Givnish, T. (1988). Adaptation to Sun and Shade: A Whole-Plant Perspective. *Functional Plant Biology*, *15*(2), 63. <https://doi.org/10.1071/PP9880063>
- Haynes, R. J., & Williams, P. H. (1993). Nutrient Cycling and Soil Fertility in the Grazed Pasture Ecosystem. In *Advances in Agronomy* (Vol. 49, pp. 119–199). Elsevier. [https://doi.org/10.1016/S0065-2113\(08\)60794-4](https://doi.org/10.1016/S0065-2113(08)60794-4)
- Heyde, M. V. D., Bennett, J. A., Pither, J., & Hart, M. (2017). Longterm effects of grazing on arbuscular mycorrhizal fungi. *Agriculture, Ecosystems & Environment*, *243*, 27–33. <https://doi.org/10.1016/j.agee.2017.04.003>
- Hodge, A., & Fitter, A. H. (2010). Substantial nitrogen acquisition by arbuscular mycorrhizal fungi from organic material has implications for N cycling. *Proceedings of the National Academy of Sciences*, *107*(31), 13754–13759. <https://doi.org/10.1073/pnas.1005874107>
- Hoeksema, J. D., Chaudhary, V. B., Gehring, C. A., Johnson, N. C., Karst, J., Koide, R. T., Pringle, A., Zabinski, C., Bever, J. D., Moore, J. C., Wilson, G. W. T., Klironomos, J. N., & Umbanhowar, J. (2010). A meta-analysis of context-dependency in plant response to inoculation with mycorrhizal fungi. *Ecology Letters*, *13*(3), 394–407. <https://doi.org/10.1111/j.1461-0248.2009.01430.x>
- Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics YOLO* (8.0.0) [Tarkvara]. <https://github.com/ultralytics/ultralytics>
- Johnson, N. C. (2010). Resource stoichiometry elucidates the structure and function of arbuscular mycorrhizas across scales. *New Phytologist*, *185*(3), 631–647. <https://doi.org/10.1111/j.1469-8137.2009.03110.x>
- Johnson, N. C., Graham, J., & Smith, F. A. (1997). Functioning of mycorrhizal associations along the mutualism–parasitism continuum*. *New Phytologist*, *135*(4), 575–585. <https://doi.org/10.1046/j.1469-8137.1997.00729.x>
- Johnson, N. C., Wilson, G. W. T., Wilson, J. A., Miller, R. M., & Bowker, M. A. (2015). Mycorrhizal phenotypes and the Law of the Minimum. *New Phytologist*, *205*(4), 1473–1484. <https://doi.org/10.1111/nph.13172>

- Kiers, E. T., Duhamel, M., Beesetty, Y., Mensah, J. A., Franken, O., Verbruggen, E., Fellbaum, C. R., Kowalchuk, G. A., Hart, M. M., Bago, A., Palmer, T. M., West, S. A., Vandenkoornhuyse, P., Jansa, J., & Bücking, H. (2011). Reciprocal Rewards Stabilize Cooperation in the Mycorrhizal Symbiosis. *Science*, 333(6044), 880–882. <https://doi.org/10.1126/science.1208473>
- Klironomos, J. N., McCune, J., & Moutoglis, P. (2004). Species of arbuscular mycorrhizal fungi affect mycorrhizal responses to simulated herbivory. *Applied Soil Ecology*, 26(2), 133–141. <https://doi.org/10.1016/j.apsoil.2003.11.001>
- Konvalinková, T., & Jansa, J. (2016). Lights Off for Arbuscular Mycorrhiza: On Its Symbiotic Functioning under Light Deprivation. *Frontiers in Plant Science*, 7. <https://doi.org/10.3389/fpls.2016.00782>
- Konvalinková, T., Püschel, D., Janoušková, M., Gryndler, M., & Jansa, J. (2015). Duration and intensity of shade differentially affects mycorrhizal growth- and phosphorus uptake responses of *Medicago truncatula*. *Frontiers in Plant Science*, 6. <https://doi.org/10.3389/fpls.2015.00065>
- Kurihara, D., Mizuta, Y., Sato, Y., & Higashiyama, T. (2015). ClearSee: A rapid optical clearing reagent for whole-plant fluorescence imaging. *Development*, dev.127613. <https://doi.org/10.1242/dev.127613>
- Lezama, F., Baeza, S., Altesor, A., Cesa, A., Chaneton, E. J., & Paruelo, J. M. (2014). Variation of grazing-induced vegetation changes across a large-scale productivity gradient. *Journal of Vegetation Science*, 25(1), 8–21. <https://doi.org/10.1111/jvs.12053>
- Liu, Y., Mao, L., Li, J., Shi, G., Jiang, S., Ma, X., An, L., Du, G., & Feng, H. (2015). Resource availability differentially drives community assemblages of plants and their root-associated arbuscular mycorrhizal fungi. *Plant and Soil*, 386(1–2), 341–355. <https://doi.org/10.1007/s11104-014-2261-z>
- Lürig, M. D., Donoughe, S., Svensson, E. I., Porto, A., & Tsuboi, M. (2021). Computer Vision, Machine Learning, and the Promise of Phenomics in Ecology and Evolutionary Biology. *Frontiers in Ecology and Evolution*, 9, 642774. <https://doi.org/10.3389/fevo.2021.642774>
- McGongile, T. P., Miller, M. H., Evans, D. G., Fairchild, G. L., & Swan, J. A. (1990). A new method which gives an objective measure of colonization of roots by vesicular—Arbuscular mycorrhizal fungi. *New Phytologist*, 115(3), 495–501. <https://doi.org/10.1111/j.1469-8137.1990.tb00476.x>

- Medina-Roldán, E., Paz-Ferreiro, J., & Bardgett, R. D. (2012). Grazing-induced effects on soil properties modify plant competitive interactions in semi-natural mountain grasslands. *Oecologia*, *170*(1), 159–169. <https://doi.org/10.1007/s00442-012-2287-y>
- Neuenkamp, L., Lewis, R. J., Koorem, K., Zobel, K., & Zobel, M. (2016). Changes in dispersal and light capturing traits explain post-abandonment community change in semi-natural grasslands. *Journal of Vegetation Science*, *27*(6), 1222–1232. <https://doi.org/10.1111/jvs.12449>
- Noë, R., & Hammerstein, P. (1995). Biological markets. *Trends in Ecology & Evolution*, *10*(8), 336–339. [https://doi.org/10.1016/S0169-5347\(00\)89123-5](https://doi.org/10.1016/S0169-5347(00)89123-5)
- Parniske, M. (2008). Arbuscular mycorrhiza: The mother of plant root endosymbioses. *Nature Reviews Microbiology*, *6*(10), 763–775. <https://doi.org/10.1038/nrmicro1987>
- Posit team. (2024). *RStudio: Integrated Development Environment for R*. Posit Software, PBC. <http://www.posit.co/>
- R Core Team. (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Semchenko, M., Lepik, M., Götzenberger, L., & Zobel, K. (2012). Positive effect of shade on plant growth: Amelioration of stress or active regulation of growth rate? *Journal of Ecology*, *100*(2), 459–466. <https://doi.org/10.1111/j.1365-2745.2011.01936.x>
- Smith, S. E., & Read, D. (2008a). INTRODUCTION. In *Mycorrhizal Symbiosis* (pp. 1–9). Elsevier. <https://doi.org/10.1016/B978-012370526-6.50002-7>
- Smith, S. E., & Read, D. (2008b). The symbionts forming arbuscular mycorrhizas. In *Mycorrhizal Symbiosis* (pp. 13–41). Elsevier. <https://doi.org/10.1016/B978-012370526-6.50003-9>
- Sügis, E. (2020). *Tehisintellekti algkursus*. Kasutatud 01.05.2024, https://courses.cs.ut.ee/2020/Tehisintellekti_algkursus/spring/Main/PARTITehisint
- Tampuu, A. (2020). *Tehisintellekti algkursus*. Kasutatud 01.05.2024, https://courses.cs.ut.ee/2020/Tehisintellekti_algkursus/spring/Main/PARTIIml
- Thanapol, P., Lavangnananda, K., Bouvry, P., Pinel, F., & Leprevost, F. (2020). Reducing Overfitting and Improving Generalization in Training Convolutional Neural Network (CNN) under Limited Sample Sizes in Image Recognition. *2020 - 5th International Conference on Information Technology (InCIT)*, 300–305. <https://doi.org/10.1109/InCIT50588.2020.9310787>

- Thessen, A. (2016). Adoption of Machine Learning Techniques in Ecology and Earth Science. *One Ecosystem*, 1, e8621. <https://doi.org/10.3897/oneeco.1.e8621>
- Vertès, F., Delaby, L., Klumpp, K., & Bloor, J. (2019). C–N–P Uncoupling in Grazed Grasslands and Environmental Implications of Management Intensification. In *Agroecosystem Diversity* (pp. 15–34). Elsevier. <https://doi.org/10.1016/B978-0-12-811050-8.00002-9>
- Vierheilig, H., Coughlan, A. P., Wyss, U., & Piché, Y. (1998). Ink and Vinegar, a Simple Staining Technique for Arbuscular-Mycorrhizal Fungi. *Applied and Environmental Microbiology*, 64(12), 5004–5007. <https://doi.org/10.1128/AEM.64.12.5004-5007.1998>
- Wilson, C. H., Strickland, M. S., Hutchings, J. A., Bianchi, T. S., & Flory, S. L. (2018). Grazing enhances belowground carbon allocation, microbial biomass, and soil carbon in a subtropical grassland. *Global Change Biology*, 24(7), 2997–3009. <https://doi.org/10.1111/gcb.14070>
- Wipf, D., Krajinski, F., Van Tuinen, D., Recorbet, G., & Courty, P. (2019). Trading on the arbuscular mycorrhiza market: From arbuscules to common mycorrhizal networks. *New Phytologist*, 223(3), 1127–1142. <https://doi.org/10.1111/nph.15775>
- Wyatt, G. A. K., Kiers, E. T., Gardner, A., & West, S. A. (2014). A biological market analysis of the plant-mycorrhizal symbiosis. *Evolution*, 68(9), 2603–2618. <https://doi.org/10.1111/evo.12466>

Lisad

Lisa 1. Koostatud R skript

```
suppressPackageStartupMessages(library(readr))
suppressPackageStartupMessages(library(stringr))
suppressPackageStartupMessages(library(dplyr))
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(reshape2))
suppressPackageStartupMessages(library(lme4))
suppressPackageStartupMessages(library(lmerTest))
suppressPackageStartupMessages(library(car))
suppressPackageStartupMessages(library(readxl))
suppressPackageStartupMessages(library(emmeans))

#set the correct path

setwd("~/Baka töö/baka_R/Data_raw/correct_annotations")
zip_files <- list.files(pattern = "\\\\.zip$")
data_col <- list()
#this might take a few minutes
for (zip_file in zip_files) {
  df1 <- read_tsv(unzip(zip_file, "col.tsv"), show_col_types = F)
  cleaned_string <- str_sub(zip_file, end = -5)
  data_col[[cleaned_string]] <- df1
  file.remove("col.tsv")
}

# Adding together tiles where roots were found, y - is colonised, n - not col
onised, but still a root tile, x -non-root tile

sum_data <- data.frame(filename = character(), yes = integer(), total = inte
ger(), stringsAsFactors = T)
for (name in names(data_col)){
  filename2 <- name
  sum_N <- sum(data_col[[name]][["N"]])
  sum_Y <- sum(data_col[[name]][["Y"]])
  sum_all <- sum_N+sum_Y
  sum_data <- sum_data %>% add_row(filename = filename2 , yes = sum_Y, total
= sum_all)
}

#For saving the data in csv format
write.csv(sum_data, "Data/sum_data_col.csv", row.names = T)

name_index <-1
collap_data <- data.frame(sample_name = character(), percentage_y = double()
, stringsAsFactors = T) #for general col
#132 samples, change if there are more than 132
while (name_index < 133) {
  search_string <- str_c("^EXP2_", name_index)
  sum_Y <- sum(sum_data$yes[grepl(search_string, sum_data$filename)])
}
```



```

sum_all <- sum(sum_data$total[grep(search_string, sum_data$filename)])
percentage_y <- (sum_Y/sum_all)*100
cleaned_string <- substring(search_string, first = 2)
collap_data <- collap_data %>% add_row(sample_name = cleaned_string , perc
centage_y = percentage_y)
name_index <- name_index+1
}

#saving collap_data in csv
write.csv(collap_data, "Data/general_col.csv", row.names = T)

img_col_data <- data.frame(sample_name = character() , percentage_y = double(
), stringsAsFactors = T)
name_index <- 1
while (name_index <= nrow(sum_data)) {
  root_y <- sum_data[name_index, 2]
  root_total <- sum_data[name_index, 3]
  percentage_y <- (root_y/root_total)*100
  img_col_data <- img_col_data %>% add_row(sample_name = sum_data[name_index
, 1] , percentage_y = percentage_y)
  name_index <- name_index+1
}

#saving img_col_data in csv
write.csv(img_col_data, "Data/img_col_data.csv", row.names = T)

#Correlation tests

#Correct annotations vs AMFinder_1904

dat1904 <- read_csv("Data/img_col_data_total_1904.csv", show_col_types = F)
test_result_v1 <- cor.test(dat1904$percentage_correct, dat1904$Finder_v1_1904
, method = "spearman")
test_result_v2<- cor.test(dat1904$percentage_correct, dat1904$Finder_v2_1904,
method = "spearman")
test_result_v3<- cor.test(dat1904$percentage_correct, dat1904$Finder_v3_1904,
method = "spearman")
test_result_v4<- cor.test(dat1904$percentage_correct, dat1904$Finder_v4_1904,
method = "spearman")
test_result_v5<- cor.test(dat1904$percentage_correct, dat1904$Finder_v5_1904,
method = "spearman")

cor_results = data.frame(model = character() , p_value = double(), s_statisti
c = double(), rho_estimate = double(), stringsAsFactors = T)

cor_results <- cor_results %>% add_row(model = 'Finder_v1', p_value = test_r
esult_v1$p.value, s_statistic = test_result_v1[["statistic"]][["S"]], rho_est
imate = test_result_v1[["estimate"]][["rho"]])

cor_results <- cor_results %>% add_row(model = 'Finder_v2', p_value = test_r
esult_v2$p.value, s_statistic = test_result_v2[["statistic"]][["S"]], rho_est
imate = test_result_v2[["estimate"]][["rho"]])

cor_results <- cor_results %>% add_row(model = 'Finder_v3', p_value = test_r

```

```

result_v3$p.value, s_statistic = test_result_v3[["statistic"]][["S"]], rho_est
imate = test_result_v3[["estimate"]][["rho"]])

cor_results <- cor_results %>% add_row(model = 'Finder_v4', p_value = test_r
esult_v4$p.value, s_statistic = test_result_v4[["statistic"]][["S"]], rho_est
imate = test_result_v4[["estimate"]][["rho"]])

cor_results <- cor_results %>% add_row(model = 'Finder_v5', p_value = test_r
esult_v5$p.value, s_statistic = test_result_v5[["statistic"]][["S"]], rho_est
imate = test_result_v5[["estimate"]][["rho"]])

#print results table
knitr::kable(cor_results)

#write.csv(cor_results, "Data/cor_results_1904.csv", row.names = T)

#plot the results
ggplot(data = cor_results, aes(x=model, y=rho_estimate, group = 1)) + geom_li
ne(color = "grey") + geom_point(size = 2) + labs(x = "Mudeli nimi", y = "Korr
elatsioonikoefitsient p") + theme_classic() + theme(axis.title = element_tex
t(size = 14), axis.text = element_text(size = 12))

#Correct annotations vs AMFinder_0105

dat0105 <- read_csv("Data/img_col_data_total_0105.csv", show_col_types = F)

test_result_v1 <- cor.test(dat0105$correct, dat0105$Finder_v1_0105, method =
"spearman")
ntest_result_v2<- cor.test(dat0105$correct, dat0105$Finder_v2_0105, method =
"spearman")
test_result_v3<- cor.test(dat0105$correct, dat0105$Finder_v3_0105, method = "
spearman")
test_result_v4<- cor.test(dat0105$correct, dat0105$Finder_v4_0105, method = "
spearman")
test_result_v5<- cor.test(dat0105$correct, dat0105$Finder_v5_0105, method = "
spearman")

cor_results_0105 = data.frame(model = character() , p_value = double(), s_sta
tistic = double(), rho_estimate = double(), stringsAsFactors = T)

cor_results_0105 <- cor_results_0105 %>% add_row(model = 'FinderAug_v1', p_v
alue = test_result_v1$p.value, s_statistic = test_result_v1[["statistic"]][["
S"]], rho_estimate = test_result_v1[["estimate"]][["rho"]])

cor_results_0105 <- cor_results_0105 %>% add_row(model = 'FinderAug_v2', p_v
alue = test_result_v2$p.value, s_statistic = test_result_v2[["statistic"]][["
S"]], rho_estimate = test_result_v2[["estimate"]][["rho"]])

cor_results_0105 <- cor_results_0105 %>% add_row(model = 'FinderAug_v3', p_v
alue = test_result_v3$p.value, s_statistic = test_result_v3[["statistic"]][["
S"]], rho_estimate = test_result_v3[["estimate"]][["rho"]])

cor_results_0105 <- cor_results_0105 %>% add_row(model = 'FinderAug_v4', p_v

```

```

alue = test_result_v4$p.value, s_statistic = test_result_v4[["statistic"]][["S"]], rho_estimate = test_result_v4[["estimate"]][["rho"]])

cor_results_0105 <- cor_results_0105 %>% add_row(model = 'FinderAug_v5', p_value = test_result_v5$p.value, s_statistic = test_result_v5[["statistic"]][["S"]], rho_estimate = test_result_v5[["estimate"]][["rho"]])

knitr::kable(cor_results_0105)

#write.csv(cor_results_0105, "Data/cor_results_0105.csv", row.names = T)

#plot the results
ggplot(data = cor_results_0105, aes(x=model, y=rho_estimate, group = 1)) + geom_line(color = "grey") + geom_point(size = 2) + labs(x = "Mudeli nimi", y = "Korrelatsioonikoefitsient p") + theme_classic() + theme(axis.title = element_text(size = 14), axis.text = element_text(size = 11))

#AMF total colonization vs AMFinder

#Classical method plotted against AMFinder models

#Data loading
dat_general <- read_xlsx("Data/AM_col.xlsx")
dat_general <- dat_general %>% filter(!is.na(AMF_colonization_total))

#Dat_general to long format

dat_long <- melt(dat_general, id.vars = 1:5, variable.name = "Finder_ver", value.name = "Finder_col") %>%
  mutate(Finder_ver = factor(Finder_ver, levels = c("Finder_v1", "Finder_v2", "Finder_v3", "Finder_v4", "Finder_v5", "Finder2_v1", "FinderAug_v1", "FinderAug_v2", "FinderAug_v3", "FinderAug_v4", "FinderAug_v5", "Train_set_sample")))

ggplot(dat_long %>% filter(Finder_ver != "Train_set_sample") , aes(x = AMF_colonization_total , y = Finder_col)) +
  geom_point(alpha = 0.4) +
  stat_smooth(method = "lm") +
  facet_wrap(~Finder_ver, ncol = 5) +
  geom_abline (slope=1, intercept = 0, linetype = "dashed", color="Red") +
  xlim(0, 100) +
  ylim(0, 100) +
  labs(y = "AMFinderi mudeli poolt määratud kolonisatsioon (%)", x = "AMF kolonisatsioon (%)") +
  theme_classic(base_size = 11)

#Ecology

#ANOVA tests for all colonization estimations

dat_general <- dat_general %>% mutate(home = as.factor(ifelse(SoilID == PlantID, "home", "away")),
  soil_aband = as.factor(ifelse(str_detect(SoilID, "^.+A$"), "soil_aband", "soil_managed")),
  plant_aband = as.factor(ifelse(str_detect(PlantID, "^.+A$"), "plant_aband", "plant_managed")))

```

```

m0 <- lmer(car::logit(AMF_colonization_total/100) ~ Treat * home + (1|PlantID)
) + (1|SoilID), data = dat_general)
m1 <- lm(car::logit(AMF_colonization_total/100) ~ Treat * home, data = dat_ge
neral)
m2 <- lm(car::logit(AMF_colonization_total/100) ~ Treat * home + plant_aband
+ soil_aband, data = dat_general)
m3 <- lmer(car::logit(AMF_colonization_total/100) ~ Treat * home + (1|PlantID
) + (1|SoilID), data = dat_general)
m4 <- lmer(car::logit(AMF_colonization_total/100) ~ Treat * plant_aband * soi
l_aband + (1|PlantID) + (1|SoilID), data = dat_general)
m5 <-lm(car::logit(AMF_colonization_total/100) ~ Treat * plant_aband * soil_a
band, data = dat_general)

anova(m0,m1,m2,m3,m4,m5)

anova_cor <- car::Anova(m1)
knitr::kable(anova_cor)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

dat_general2 <- dat_general %>%
  mutate(Treat = ifelse(Treat == "clipped", "lõigatud", Treat))
dat_general2 <- dat_general2 %>%
  mutate(Treat = ifelse(Treat == "shaded", "varjutatud", Treat))
dat_general2 <- dat_general2 %>%
  mutate(Treat = ifelse(Treat == "unclipped", "lõikamata", Treat))
dat_general2$Treat <- as.factor(dat_general2$Treat)
dat_general2$Treat <- factor(dat_general2$Treat, levels = c("lõigatud", "varj
utatud", "lõikamata"))

amf_col_plot <- ggplot(dat_general2, aes(y = AMF_colonization_total, x = Trea
t, fill = home)) +
  geom_boxplot() + labs(x = "Töötlus", y = "AM kolonisatsioon") +
  annotate("text", x = 1, y = 60, label = "A") +
  annotate("text", x = 2, y = 60, label = "B") +
  annotate("text", x = 3, y = 60, label = "A") +
  scale_fill_brewer(labels = c("võõras", "kodu"), palette = "Dark2")+
  theme_classic() +
  labs(fill = "Mulla päritolu")

m0 <- lmer(car::logit(Finder_v1/100) ~ Treat * home + (1|PlantID) + (1|SoilID
), data = dat_general)
m1 <- lm(car::logit(Finder_v1/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(Finder_v1/100) ~ Treat * home + plant_aband + soil_aband,
data = dat_general)
m3 <- lmer(car::logit(Finder_v1/100) ~ Treat * home + (1|PlantID) + (1|SoilID
), data = dat_general)
m4 <- lmer(car::logit(Finder_v1/100) ~ Treat * plant_aband * soil_aband + (1|
PlantID) + (1|SoilID), data = dat_general)

```

```

m5 <- lm(car::logit(Finder_v1/100) ~ Treat * plant_aband * soil_aband, data =
dat_general)

anova(m0,m1,m2,m3,m4,m5)

anova_v1 <- car::Anova(m1)
knitr::kable(anova_v1)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

ggplot(dat_general, aes(y = Finder_v1, x = Treat, fill = home)) +
  geom_boxplot() + labs(x = "Töötlus", y = "Finder_v1") +
  scale_fill_brewer(labels = c("võõras", "kodu"), palette = "Dark2")+
  theme_classic() +
  labs(fill = "Mulla päritolu")

m0 <- lmer(car::logit(Finder_v2/100) ~ Treat * home + (1|PlantID) + (1|SoilID
), data = dat_general)
m1 <- lm(car::logit(Finder_v2/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(Finder_v2/100) ~ Treat * home + plant_aband + soil_aband,
data = dat_general)
m3 <- lmer(car::logit(Finder_v2/100) ~ Treat * home + (1|PlantID) + (1|SoilID
), data = dat_general)
m4 <- lmer(car::logit(Finder_v2/100) ~ Treat * plant_aband * soil_aband + (1|
PlantID) + (1|SoilID), data = dat_general)
m5 <- lm(car::logit(Finder_v1/100) ~ Treat * plant_aband * soil_aband, data =
dat_general)

anova(m0,m1,m2,m3,m4,m5)

anova_v2 <- car::Anova(m1)
knitr::kable(anova_v2)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

multcomp::cld(emm_mod)

ggplot(dat_general, aes(y = Finder_v2, x = Treat, fill = home)) +
  geom_boxplot()

m0 <- lmer(car::logit(Finder_v3/100) ~ Treat * home + (1|PlantID) + (1|SoilID
), data = dat_general)
m1 <- lm(car::logit(Finder_v3/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(Finder_v3/100) ~ Treat * home + plant_aband + soil_aband,
data = dat_general)
m3 <- lmer(car::logit(Finder_v3/100) ~ Treat * home + (1|PlantID) + (1|SoilID
), data = dat_general)
m4 <- lmer(car::logit(Finder_v3/100) ~ Treat * plant_aband * soil_aband + (1|
PlantID) + (1|SoilID), data = dat_general)

```

```

m5 <- lm(car::logit(Finder_v3/100) ~ Treat * plant_aband * soil_aband, data =
dat_general)

anova(m0,m1,m2,m3,m4,m5)

anova_v3 <- car::Anova(m1)
knitr::kable(anova_v3)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

multcomp::cld(emm_mod)

ggplot(dat_general, aes(y = Finder_v3, x = Treat, fill = home)) +
  geom_boxplot()

m0 <- lmer(car::logit(Finder_v4/100) ~ Treat * home + (1|PlantID) + (1|SoilID
), data = dat_general)
m1 <- lm(car::logit(Finder_v4/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(Finder_v4/100) ~ Treat * home + plant_aband + soil_aband,
data = dat_general)
m3 <- lmer(car::logit(Finder_v4/100) ~ Treat * home + (1|PlantID) + (1|SoilID
), data = dat_general)
m4 <- lmer(car::logit(Finder_v4/100) ~ Treat * plant_aband * soil_aband + (1|
PlantID) + (1|SoilID), data = dat_general)
m5 <- lm(car::logit(Finder_v4/100) ~ Treat * plant_aband * soil_aband, data =
dat_general)

anova(m0,m1,m2,m3,m4,m5)

anova_v4 <- car::Anova(m1)
knitr::kable(anova_v4)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

multcomp::cld(emm_mod)

ggplot(dat_general, aes(y = Finder_v4, x = Treat, fill = home)) +
  geom_boxplot()

m0 <- lmer(car::logit(Finder_v5/100) ~ Treat * home + (1|PlantID) + (1|SoilID
), data = dat_general)
m1 <- lm(car::logit(Finder_v5/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(Finder_v5/100) ~ Treat * home + plant_aband + soil_aband,
data = dat_general)
m3 <- lmer(car::logit(Finder_v5/100) ~ Treat * home + (1|PlantID) + (1|SoilID
), data = dat_general)
m4 <- lmer(car::logit(Finder_v5/100) ~ Treat * plant_aband * soil_aband + (1|

```

```

PlantID) + (1|SoilID), data = dat_general)
m5 <- lm(car::logit(Finder_v5/100) ~ Treat * plant_aband * soil_aband, data =
dat_general)

anova(m0,m1,m2,m3,m4,m5)

anova_v5 <- car::Anova(m1)
knitr::kable(anova_v5)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

multcomp::cld(emm_mod)

ggplot(dat_general, aes(y = Finder_v5, x = Treat, fill = home)) +
  geom_boxplot()

m0 <- lmer(car::logit(FinderAug_v1/100) ~ Treat * home + (1|PlantID) + (1|Soi
lID), data = dat_general)
m1 <- lm(car::logit(FinderAug_v1/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(FinderAug_v1/100) ~ Treat * home + plant_aband + soil_aba
nd, data = dat_general)
m3 <- lmer(car::logit(FinderAug_v1/100) ~ Treat * home + (1|PlantID) + (1|Soi
lID), data = dat_general)
m4 <- lmer(car::logit(FinderAug_v1/100) ~ Treat * plant_aband * soil_aband +
(1|PlantID) + (1|SoilID), data = dat_general)
m5 <- lm(car::logit(FinderAug_v1/100) ~ Treat * plant_aband * soil_aband, data
= dat_general)

anova(m0,m1,m2,m3,m4,m5)

anovaAug_v1 <- car::Anova(m1)
knitr::kable(anovaAug_v1)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

multcomp::cld(emm_mod)

ggplot(dat_general, aes(y = FinderAug_v1, x = Treat, fill = home)) +
  geom_boxplot()

m0 <- lmer(car::logit(FinderAug_v2/100) ~ Treat * home + (1|PlantID) + (1|Soi
lID), data = dat_general)
m1 <- lm(car::logit(FinderAug_v2/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(FinderAug_v2/100) ~ Treat * home + plant_aband + soil_aba
nd, data = dat_general)
m3 <- lmer(car::logit(FinderAug_v2/100) ~ Treat * home + (1|PlantID) + (1|Soi
lID), data = dat_general)
m4 <- lmer(car::logit(FinderAug_v2/100) ~ Treat * plant_aband * soil_aband +
(1|PlantID) + (1|SoilID), data = dat_general)
m5 <- lm(car::logit(FinderAug_v2/100) ~ Treat * plant_aband * soil_aband, data
= dat_general)

anova(m0,m1,m2,m3,m4,m5)

```

```

anovaAug_v2 <- car::Anova(m1)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

multcomp::cld(emm_mod)

ggplot(dat_general, aes(y = FinderAug_v2, x = Treat, fill = home)) +
  geom_boxplot()

m0 <- lmer(car::logit(FinderAug_v3/100) ~ Treat * home + (1|PlantID) + (1|SoilID), data = dat_general)
m1 <- lm(car::logit(FinderAug_v3/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(FinderAug_v3/100) ~ Treat * home + plant_aband + soil_aband, data = dat_general)
m3 <- lmer(car::logit(FinderAug_v3/100) ~ Treat * home + (1|PlantID) + (1|SoilID), data = dat_general)
m4 <- lmer(car::logit(FinderAug_v3/100) ~ Treat * plant_aband * soil_aband + (1|PlantID) + (1|SoilID), data = dat_general)
m5 <- lm(car::logit(FinderAug_v3/100) ~ Treat * plant_aband * soil_aband, data = dat_general)

anova(m0,m1,m2,m3,m4,m5)

anovaAug_v3 <- car::Anova(m1)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

multcomp::cld(emm_mod)

ggplot(dat_general, aes(y = FinderAug_v3, x = Treat, fill = home)) +
  geom_boxplot()

m0 <- lmer(car::logit(FinderAug_v4/100) ~ Treat * home + (1|PlantID) + (1|SoilID), data = dat_general)
m1 <- lm(car::logit(FinderAug_v4/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(FinderAug_v4/100) ~ Treat * home + plant_aband + soil_aband, data = dat_general)
m3 <- lmer(car::logit(FinderAug_v4/100) ~ Treat * home + (1|PlantID) + (1|SoilID), data = dat_general)
m4 <- lmer(car::logit(FinderAug_v4/100) ~ Treat * plant_aband * soil_aband + (1|PlantID) + (1|SoilID), data = dat_general)
m5 <- lm(car::logit(FinderAug_v2/100) ~ Treat * plant_aband * soil_aband, data = dat_general)

anova(m0,m1,m2,m3,m4,m5)

anovaAug_v4 <- car::Anova(m1)
knitr::kable(anovaAug_v4)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

multcomp::cld(emm_mod)

```



```

ggplot(dat_general, aes(y = FinderAug_v4, x = Treat, fill = home)) +
  geom_boxplot()

m0 <- lmer(car::logit(FinderAug_v5/100) ~ Treat * home + (1|PlantID) + (1|SoilID), data = dat_general)
m1 <- lm(car::logit(FinderAug_v5/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(FinderAug_v5/100) ~ Treat * home + plant_aband + soil_aband, data = dat_general)
m3 <- lmer(car::logit(FinderAug_v5/100) ~ Treat * home + (1|PlantID) + (1|SoilID), data = dat_general)
m4 <- lmer(car::logit(FinderAug_v5/100) ~ Treat * plant_aband * soil_aband + (1|PlantID) + (1|SoilID), data = dat_general)
m5 <- lm(car::logit(FinderAug_v5/100) ~ Treat * plant_aband * soil_aband, data = dat_general)

anova(m0,m1,m2,m3,m4,m5)

anovaAug_v5 <- car::Anova(m1)
knitr::kable(anovaAug_v5)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

multcomp::cld(emm_mod)

ggplot(dat_general, aes(y = FinderAug_v5, x = Treat, fill = home)) +
  geom_boxplot()

#Correct annotations (hand annotations)

m0 <- lmer(car::logit(Train_set_sample/100) ~ Treat * home + (1|PlantID) + (1|SoilID), data = dat_general)
m1 <- lm(car::logit(Train_set_sample/100) ~ Treat * home, data = dat_general)
m2 <- lm(car::logit(Train_set_sample/100) ~ Treat * home + plant_aband + soil_aband, data = dat_general)
m3 <- lmer(car::logit(Train_set_sample/100) ~ Treat * home + (1|PlantID) + (1|SoilID), data = dat_general)
m4 <- lmer(car::logit(Train_set_sample/100) ~ Treat * plant_aband * soil_aband + (1|PlantID) + (1|SoilID), data = dat_general)
m5 <- lm(car::logit(Train_set_sample/100) ~ Treat * plant_aband * soil_aband, data = dat_general)

anova(m0,m1,m2,m3,m4,m5)

car::Anova(m1)

emm_mod <- emmeans(m1, "Treat")
pairs(emm_mod)

multcomp::cld(emm_mod)

ggplot(dat_general, aes(y = Train_set_sample, x = Treat, fill = home)) + geom_boxplot()

```

Lisa 2. Jooksutatud käsud

```
cd amfinder-master/amf

source amfenv/bin/activate

./amf train -net CNN1v2.net -o ./new_networks ./set1/*.jpg | tee -a
terminal_training_CNN1v2_1904.txt

cd new_networks/

unzip 2024-04-19_18\ :20\ :16.075730_training.zip
#renamed col.h5 to CNN1v2_v1_1903.h5 and moved to trained_networks
cd ..

./amf train -net CNN1v2_v1.net -o ./new_networks ./set2/*.jpg | tee -a
terminal_training_CNN1v2_1904.txt

cd new_networks/
unzip 2024-04-19_18\ :32\ :45.305877_training.zip
#renamed col.h5 to CNN1v2_v2_1903.h5 and moved to trained_networks
cd ..

./amf train -net CNN1v2_v2.net -o ./new_networks ./set3/*.jpg | tee -a
terminal_training_CNN1v2_1904.txt

cd new_networks/
unzip 2024-04-19_18\ :46\ :26.494599_training.zip
#renamed col.h5 to CNN1v2_v3_1903.h5 and moved to trained_networks
cd ..

./amf train -net CNN1v2_v3.net -o ./new_networks ./set4/*.jpg | tee -a
terminal_training_CNN1v2_1904.txt

cd new_networks/
unzip 2024-04-19_19\ :00\ :16.132400_training.zip
#renamed col.h5 to CNN1v2_v4_1903.h5 and moved to trained_networks
cd ..

./amf train -net CNN1v2_v4.net -o ./new_networks ./set5/*.jpg | tee -a
terminal_training_CNN1v2_1904.txt

cd new_networks/
unzip 2024-04-19_19\ :04\ :41.501444_training.zip
#renamed col.h5 to CNN1v2_v5_1903.h5 and moved to trained_networks
cd ..

deactivate
echo "done"
```

Lisa 3. Modifitseeritud fail amfinder_convert.py

```
# AMFinder - amfinder_convert.py
#
# MIT License
# Copyright (c) 2021 Edouard Evangelisti, Carl Turner
#
# Permission is hereby granted, free of charge, to any person obtaining a
# copy
# of this software and associated documentation files (the "Software"), to
# deal in the Software without restriction, including without limitation the
# rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
# sell copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
# DEALINGS
# IN THE SOFTWARE.

#import imagesize
import io
import os
import re
import struct
from xml.etree import ElementTree
import json
import numpy as np
import pandas as pd
#import zipfile as zf
# Reference: https://stackoverflow.com/a/69115481
# Local version of zipfile to get access to <remove>.
import amfinder_zipfile as zf
import amfinder_log as AmfLog
import amfinder_save as AmfSave
import amfinder_train as AmfTrain
import amfinder_config as AmfConfig
import amfinder_diagnose as AmfDiagnose

NROWS = None
NCOLS = None

_UNIT_KM = -3
_UNIT_100M = -2
_UNIT_10M = -1
_UNIT_1M = 0
_UNIT_10CM = 1
_UNIT_CM = 2
_UNIT_MM = 3
_UNIT_0_1MM = 4
_UNIT_0_01MM = 5
_UNIT_UM = 6
_UNIT_INCH = 6
```

```

_TIFF_TYPE_SIZES = {
1: 1,
2: 1,
3: 2,
4: 4,
5: 8,
6: 1,
7: 1,
8: 2,
9: 4,
10: 8,
11: 4,
12: 8,
}

def initialize_size(path, zfile):
    """
    Retrieve the number of rows and columns based on image and tile sizes.
    """
    tile_size = None
    width, height = imagesizeget(path)

    with zf.ZipFile(zfile, 'r') as z:
        data = z.read(AmfSave.IMG_SETTINGS).decode('utf-8')
        tile_size = json.loads(data)['tile_edge']

    global NROWS, NCOLS
    assert tile_size is not None
    NCOLS = width // tile_size
    NROWS = height // tile_size

def preds_to_python_annot_1(path, preds):
    """
    Convert level 1 predictions to Python annotations.
    """

    # Only one file, nothing special to choose.
    preds = io.StringIO(preds)
    preds = pd.read_csv(preds, sep='\t')

    # Get the predictions for automatic conversion.
    coord = preds[['row', 'col']]
    preds = AmfDiagnose.remove_coordinates(preds)
    preds = preds.to_numpy()

    # Perform the conversion, ignoring ties.
    conv = np.zeros_like(preds)
    conv[np.arange(len(preds)), preds.argmax(1)] = 1
    conv = conv.astype(np.uint8)

    print(os.path.basename(path) + '\t' +
          '\t'.join([str(x) for x in np.sum(conv, axis=0)]))

    conv = pd.DataFrame(data=conv, columns=AmfConfig.get('header'))

    # Generate the final table.
    return pd.concat([coord, conv], axis=1)

```

```

def preds_to_python_annot_2(path, preds):
    """
    Convert level 2 predictions to Python annotations.
    """

    # Only one file, nothing special to choose.
    preds = io.StringIO(preds)
    preds = pd.read_csv(preds, sep='\t')

    # Get the predictions for automatic conversion.
    coord = preds[['row', 'col']]
    preds = AmfDiagnose.remove_coordinates(preds)
    preds = preds.to_numpy()

    # Perform the conversion, ignoring ties.
    conv = np.zeros_like(preds)
    conv[preds >= AmfConfig.get('threshold')] = 1

    conv = conv.astype(np.uint8)

    print(os.path.basename(path) + '\t' +
          '\t'.join([str(x) for x in np.sum(conv, axis=0)]))

    header = AmfConfig.get('header')
    conv = pd.DataFrame(data=conv, columns=header)

    # Generate the final table.
    final = pd.concat([coord, conv], axis=1)

    # Remove the columns without annotations
    final = final.loc[(final[header].sum(axis=1) != 0), ]

    return final

def python_annot_to_ocaml_1(out, zfile):
    """
    Convert level 1 predictions to OCaml annotations.
    """

    mat1 = np.zeros((NROWS, NCOLS)).astype('<U1')
    mat2 = np.zeros((NROWS, NCOLS)).astype('<U1')

    header = AmfConfig.get('header')
    out = out.reset_index()

    for _, row in out.iterrows():
        r = row['row']
        c = row['col']
        mat1[r, c] = header[np.argmax(row[header])]
        mat2[r, c] = '' # this one will remain empty.

    with zf.ZipFile(zfile, 'a') as z:

        d1 = pd.DataFrame(mat1).to_csv(sep='\t', encoding='utf-8',
                                       index=False, header=False,
                                       mode='a', line_terminator='')
        zi = AmfSave.get_zip_info('annotations/col.caml', 1)

```

```

# There is still a trailing character at the end of the csv text.
z.writestr(zi, d1[:-1])
d2 = pd.DataFrame(mat2).to_csv(sep='\t', encoding='utf-8',
                              index=False, header=False,
                              mode='a', line_terminator='')
zi = AmfSave.get_zip_info('annotations/myc.caml', 0)
# There is still a trailing character at the end of the csv text.
z.writestr(zi, d2[:-1])

```

```

def python_annot_to_ocaml_2(out, zfile):
    """
    Convert level 2 predictions to OCaml annotations.
    """

    mat2 = np.zeros((NROWS, NCOLS), str).astype('<U4')

    header = AmfConfig.get('header')
    out = out.reset_index()

    for _, row in out.iterrows():
        r = row['row']
        c = row['col']
        indices = [i for i, x in enumerate(row[header]) if x == 1]
        mat2[r, c] = ''.join(sorted([header[i] for i in indices]))

    with zf.ZipFile(zfile, 'a') as z:
        z.remove('annotations/myc.caml')
        d2 = pd.DataFrame(mat2).to_csv(sep='\t', encoding='utf-8',
                                       index=False, header=False,
                                       mode='a', line_terminator='')
        zi = AmfSave.get_zip_info('annotations/myc.caml', 0)
        # There is still a trailing character at the end of the csv text.
        z.writestr(zi, d2[:-1])

```

```

def preds_to_python_annot(path, preds):
    """
    Convert predictions to Python annotations.
    """
    if AmfConfig.get('level') == 1:
        return preds_to_python_annot_1(path, preds)
    else:
        return preds_to_python_annot_2(path, preds)

```

```

def python_annot_to_ocaml(out, zfile):
    """
    Convert and save predictions to OCaml annotations.
    """
    if AmfConfig.get('level') == 1:
        python_annot_to_ocaml_1(out, zfile)
    else:

```

```

        python_annot_to_ocaml_2(out, zfile)

update_archive(out, zfile)

def update_archive(out, zfile):
    """
    Save annotations in Python format.
    """
    with zf.ZipFile(zfile, 'a') as z:
        data = out.to_csv(sep='\t', encoding='utf-8', index=False,
                          mode='a', line_terminator='')
        zi = AmfSave.get_zip_info(AmfConfig.tsv_name(),
AmfConfig.string_of_level())
        z.writestr(zi, data[:-1])

def create_annotations(path, zfile):
    preds = []

    with zf.ZipFile(zfile, 'r') as z:
        if AmfConfig.tsv_name() in z.namelist():
            AmfLog.info(f'Skipping {path} as annotations already exist')
            return

        for x in z.namelist():
            if os.path.dirname(x) == 'predictions':
                if z.getinfo(x).comment.decode('utf-8') ==
AmfConfig.string_of_level():
                    preds.append(x)

        if preds == []:
            AmfLog.info(f'Skipping {path} as no predictions could be found')

        elif len(preds) == 1:
            data = z.read(preds[0]).decode('utf-8')
            out = preds_to_python_annot(path, data)
            python_annot_to_ocaml(out, zfile)

        else:
            AmfLog.info(f'Skipping {path} as <amf predict> does not \
support multiple prediction files.')

def run(input_images):
    print('Image\t' + '\t'.join(AmfConfig.human_redable_header()))

```

```

for path in input_images:

    # Make sure the image comes with a valid zip file.
    zfile = AmfTrain.get_zipfile(path)

    if not zf.is_zipfile(zfile):

        AmfLog.warning(f'File {path} has no associated zip file.')

    initialize_size(path, zfile)
    create_annotations(path, zfile)

def imagesizeget(filepath):
    """
    Return (width, height) for a given img file content
    no requirements
    :type filepath: Union[bytes, str, pathlib.Path]
    :rtype Tuple[int, int]
    """
    height = -1
    width = -1

    if isinstance(filepath, io.BytesIO): # file-like object
        fhandle = filepath
    else:
        fhandle = open(filepath, 'rb')

    try:
        head = fhandle.read(31)
        size = len(head)
        # handle GIFs
        if size >= 10 and head[:6] in (b'GIF87a', b'GIF89a'):
            # Check to see if content_type is correct
            try:
                width, height = struct.unpack("<hh", head[6:10])
            except struct.error:
                raise ValueError("Invalid GIF file")
            # see png edition spec bytes are below chunk length then and finally
            the
        elif size >= 24 and head.startswith(b'\211PNG\r\n\032\n') and
        head[12:16] == b'IHDR':
            try:
                width, height = struct.unpack(">LL", head[16:24])
            except struct.error:
                raise ValueError("Invalid PNG file")
            # Maybe this is for an older PNG version.
        elif size >= 16 and head.startswith(b'\211PNG\r\n\032\n'):
            # Check to see if we have the right content type
            try:
                width, height = struct.unpack(">LL", head[8:16])
            except struct.error:
                raise ValueError("Invalid PNG file")
        # handle JPEGs
        elif size >= 2 and head.startswith(b'\377\330'):
            try:
                fhandle.seek(0) # Read 0xff next
                size = 2
                ftype = 0

```



```

while not 0xc0 <= ftype <= 0xcf or ftype in [0xc4, 0xc8,
0xcc]:
    fhandle.seek(size, 1)
    byte = fhandle.read(1)
    while ord(byte) == 0xff:
        byte = fhandle.read(1)
    ftype = ord(byte)
    size = struct.unpack('>H', fhandle.read(2))[0] - 2
    # We are at a SOFn block
    fhandle.seek(1, 1) # Skip `precision' byte.
    height, width = struct.unpack('>HH', fhandle.read(4))
except (struct.error, TypeError):
    raise ValueError("Invalid JPEG file")
# handle JPEG2000s
elif size >= 12 and head.startswith(b'\x00\x00\x00\x0cjp
\r\n\x87\n'):
    fhandle.seek(48)
    try:
        height, width = struct.unpack('>LL', fhandle.read(8))
    except struct.error:
        raise ValueError("Invalid JPEG2000 file")
# handle big endian TIFF
elif size >= 8 and head.startswith(b"\x4d\x4d\x00\x2a"):
    offset = struct.unpack('>L', head[4:8])[0]
    fhandle.seek(offset)
    ifdsizesize = struct.unpack(">H", fhandle.read(2))[0]
    for i in range(ifdsizesize):
        tag, datatype, count, data = struct.unpack(">HLLL",
fhandle.read(12))
        if tag == 256:
            if datatype == 3:
                width = int(data / 65536)
            elif datatype == 4:
                width = data
            else:
                raise ValueError("Invalid TIFF file: width column
data type should be SHORT/LONG.")
        elif tag == 257:
            if datatype == 3:
                height = int(data / 65536)
            elif datatype == 4:
                height = data
            else:
                raise ValueError("Invalid TIFF file: height column
data type should be SHORT/LONG.")
            if width != -1 and height != -1:
                break
            if width == -1 or height == -1:
                raise ValueError("Invalid TIFF file: width and/or height IDS
entries are missing.")
        elif size >= 8 and head.startswith(b"\x49\x49\x2a\x00"):
            offset = struct.unpack('<L', head[4:8])[0]
            fhandle.seek(offset)
            ifdsizesize = struct.unpack("<H", fhandle.read(2))[0]
            for i in range(ifdsizesize):
                tag, datatype, count, data = struct.unpack("<HLLL",
fhandle.read(12))
                if tag == 256:
                    width = data
                elif tag == 257:
                    height = data

```

```

        if width != -1 and height != -1:
            break
    if width == -1 or height == -1:
        raise ValueError("Invalid TIFF file: width and/or height IDS
entries are missing.")
    # handle little endian BigTiff
    elif size >= 8 and head.startswith(b"\x49\x49\x2b\x00"):
        bytesize_offset = struct.unpack('<L', head[4:8])[0]
        if bytesize_offset != 8:
            raise ValueError('Invalid BigTIFF file: Expected offset to be
8, found {} instead.'.format(offset))
        offset = struct.unpack('<Q', head[8:16])[0]
        fhandle.seek(offset)
        ifdsizesize = struct.unpack("<Q", fhandle.read(8))[0]
        for i in range(ifdsizesize):
            tag, datatype, count, data = struct.unpack("<HHQQ",
fhandle.read(20))
            if tag == 256:
                width = data
            elif tag == 257:
                height = data
            if width != -1 and height != -1:
                break
    if width == -1 or height == -1:
        raise ValueError("Invalid BigTIFF file: width and/or height
IDS entries are missing.")

    # handle SVGs
    elif size >= 5 and (head.startswith(b'<?xml') or
head.startswith(b'<svg')):
        fhandle.seek(0)
        data = fhandle.read(1024)
        try:
            data = data.decode('utf-8')
            width = re.search(r'^-]width="(.*?)"', data).group(1)
            height = re.search(r'^-]height="(.*?)"', data).group(1)
        except Exception:
            raise ValueError("Invalid SVG file")
        width = _convertToPx(width)
        height = _convertToPx(height)

    # handle Netpbm
    elif head[:1] == b"P" and head[1:2] in b"123456":
        fhandle.seek(2)
        sizes = []

        while True:
            next_chr = fhandle.read(1)

            if next_chr.isspace():
                continue

            if next_chr == b"":
                raise ValueError("Invalid Netpbm file")

            if next_chr == b"#":
                fhandle.readline()
                continue

            if not next_chr.isdigit():

```

```

        raise ValueError("Invalid character found on Netpbm
file")

    size = next_chr
    next_chr = fhandle.read(1)

    while next_chr.isdigit():
        size += next_chr
        next_chr = fhandle.read(1)

    sizes.append(int(size))

    if len(sizes) == 2:
        break

    fhandle.seek(-1, os.SEEK_CUR)
    width, height = sizes
    elif head.startswith(b"RIFF") and head[8:12] == b"WEBP":
        if head[12:16] == b"VP8 ":
            width, height = struct.unpack("<HH", head[26:30])
        elif head[12:16] == b"VP8X":
            width = struct.unpack("<I", head[24:27] + b"\0")[0]
            height = struct.unpack("<I", head[27:30] + b"\0")[0]
        elif head[12:16] == b"VP8L":
            b = head[21:25]
            width = (((b[1] & 63) << 8) | b[0]) + 1
            height = (((b[3] & 15) << 10) | (b[2] << 2) | ((b[1] & 192)
>> 6)) + 1
        else:
            raise ValueError("Unsupported WebP file")

    finally:
        fhandle.close()

    return width, height

```

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Lauri Laaspere,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

„Karjatamise ja varjutamise mõju krohmseente ning taimede vahelistele interaktsioonidele“,

mille juhendajad on Marina Semtšenko ja Siim-Kaarel Sepp,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Lauri Laaspere

22.05.2024