Australian Government

MURRAY–DARLING BASIN AUTHORITY

The Murray–Darling Basin Tree Stand Condition Tool Hindcast Report

December 2020

Published by the Murray–Darling Basin Authority
MDBA publication no: 60/20
ISBN (online): 978-1-922396-29-7

GPO Box 1801, Canberra ACT 2601
engagement@mdba.gov.au
1800 230 067
mdba.gov.au

Cataloguing data

Title: The Murray–Darling Basin Tree Stand Condition Tool Hindcast Report, Murray–Darling Basin Authority Canberra, 2020. CC BY 4.0

Accessibility

The Murray–Darling Basin Authority makes its documents and information available in accessible formats. On some occasions the highly technical nature of the document means that we cannot make some sections fully accessible. If you encounter accessibility problems or the document is in a format that you cannot access, please contact us.

Acknowledgement of the Traditional Owners of the Murray–Darling Basin

The Murray–Darling Basin Authority pays respect to the Traditional Owners and their Nations of the Murray−Darling Basin. We acknowledge their deep cultural, social, environmental, spiritual and economic connection to their lands and waters.

The guidance and support received from the Murray Lower Darling Rivers Indigenous Nations, the Northern Basin Aboriginal Nations and our many Traditional Owner friends and colleagues is very much valued and appreciated.

Aboriginal people should be aware that this publication may contain images, names or quotations of deceased persons.

Contents

Glossary

MDBA	Murray–Darling Basin Authority
GA	Geoscience Australia
ARI	Arthur Rylah Institute
VTAG	Vegetation Technical Advisory Group is an expert sub-committee of the Joint Venture Monitoring & Evaluation Working Group with representation form Basin Jurisdictions
Condition	Ecological “condition” refers to the state of ecological systems, which includes their physical, chemical, and biological characteristics and the processes and interactions that connect them. For the purpose of the Stand Condition Tool condition is ascribed a score as defined by Equation *Eq2*.
The Basin	The Murray–Darling Basin
Machine learning	Machine learning refers to a broad area of computer science, and is generally an application of artificial intelligence that provides systems the ability to automatically learn and, in some cases, improve from experience without being explicitly programmed.
Information integrity	Information integrity is the trustworthiness and dependability of information. More specifically, it is the accuracy, consistency and reliability of the information content, processes and systems.
Model training	The process of training a machine learning model involves providing a machine with training data to learn from. The term machine learning model refers to the model artifact that is created by the training process.
Model upgrade	An improvement to any part of the tool including data, algorithms or application, does not necessarily need to involve model training.
Benchmarking	Creating the relationship between condition scores (from 0-1) and condition categories, i.e. good, poor, etc.
Spatial benchmarking	Assigning condition category benchmarks across space.
Temporal benchmarking	Assigning condition category benchmarks across time.

Executive Summary

The Stand Condition Tool (SCT) is a process by which a combination satellite data, field surveys and machine learning algorithms are applied to infer the condition of stands of key tree species across the entire Murray Darling Basin. The MDBA has run the SCT on the 30 years of available Landsat imagery (a Hindcast). The input to the SCT is the Landsat satellite imagery, and the outputs are predictions of tree stand condition. The variables used to train the model and collected in field surveys are, plant area index (PAI), live basal area (LBA) and crown extent (CE). These field variables together provide the metrics by which to calculate the Stand Condition Score (SCS), which is used to infer the condition of tree stands. The field data used to train the SCT has been collected for river red gum, coolibah, and black box dominated tree stands. The purpose of this report is to detail how the Hindcast data was produced. It also continues a discussion around the practicability of, and potential refinements to the tool.

A portion of the field survey data is reserved to validate the outputs of the tool. Due to the modifications to the tools operation this process was vital part of the hindcast. The validation metric (r squared) produced for the hindcast outputs were found to be like those from the 2017 operation of the tool with annual figures in the range 0.5-0.7. Validation with field data is not possible for SCT outputs for years prior to field data being collected, however significantly extending the record of inferred tree stand condition provides context for more recent validated results by providing new information about the dynamics of these ecosystems, where tree stand condition dynamics could previously be described over multiple years they can now be described over multiple decades.

It is evident that the SCT does not replace the need to collect field data, but it is significantly improved by, and enhances the utility of, the field data collected. To this end it is vital to the tools utility that field data is captured in a targeted manner, i.e. of tree stands in areas and in conditions that are not well representing in the existing field survey data set.

The field variables used as indicators of tree health may have different local maximum values due to regional differences in environmental conditions (for example being in different rainfall zones). Thought should be given to determine how the results should be interpreted. One option explored in this report is only to look at relative change within forests to avoid needing to consider the different environmental conditions exerting influence over the field variables. The second option is to undertake a ‘Benchmarking’ exercise to determine what the local maximum condition value and reporting results against this value for each forest, this has the advantage of being able to compare forests while considering regional differences in environmental variables. The MDBA considers addressing this to be paramount if the tool will continue to be used to report on legislative requirements.

The MDBA used the tree species map from Cunningham *et al.* (2013c), however, there are a wide range of species distribution maps being used by various Basin stakeholders. Due to this, we consider that developing an improved standard tree species map across the Basin has the highest utility of all upgrades identified. This task could also be expanded to the development of a process to dynamically identify the extent of key tree species from remotely sensed data. This would allow reporting on extent as well as condition, which is a requirement of the Basin Plan (2012).

Through developing the SCT hindcast results the MDBA has gained a greater understanding of the issues limiting both the development and the application of the tool. Many of these issues could be addressed by moving the SCT data processing and design to an open platform such as Geoscience Australia's Digital Earth Australia SandBox or Google Earth Engine.

Moving the implementation of the tool to either of these platforms will enable:

- Easy integration into low cost high-performance computing environments
- Flexibility to modify the parameters and process flow experimentally
- Scalability to use higher resolution imagery and support more complex models.
- The removal of barriers to collaboration between jurisdictions and with the academic community
- Better informed discussions on work prioritisation and resourcing

The MDBA recommends the creation of an officer level working group with membership from relevant state and federal government agencies. The purpose of this group would be two-fold, firstly to collaboratively operationalize the tool in an open source cloud computing environment and secondly to inform strategic discussions around improvements and resource allocation.

1. Background

Notable declines in the condition of floodplain and riverine forests of the Murray–Darling Basin (the Basin) have been evident for decades (Cunningham *et al.* 2007 & Overton *et al.* 2005) and in some cases exacerbated by the millennium drought (Department for Environment and Water 2019). These forests provide and support important social, environmental, cultural and material resources and values.

Governments have invested heavily in protecting and restoring the water dependent ecosystems of the Basin. Scientifically robust, reliable and repeatable monitoring and evaluation of the condition of these ecological communities is essential to effective adaptive management of these ecosystems and the Basin's water resources on which they rely.

To meet this need the MDBA and Basin States have invested in extensive field surveys of tree stand condition and engaged the Arthur Rylah Institute (ARI) to apply machine learning technologies to develop relationships between remotely sensed and field data so that stand condition can be inferred spatially and continuously across very large areas. This resulted in the production of the Murray Stand Condition Modeller (V1.08) in 2013 (Cunningham & Griffioen 2013). In 2017 the underlying algorithms which determine stand condition based on Landsat imagery were updated to incorporate what was then new field data (Newell *et al.* 2017) and it is now widely referred to as the Stand Condition Tool (SCT) or Stand Condition Assessment tool (SCA Tool).

The SCT was applied as part of the 2017 Basin Plan evaluation and reported on condition stands of river red gum (*Eucalyptus Canadensis*), black box (*Eucalyptus largiflorens*) and coolibah (*Eucalyptus coolabah*) occurring within the managed floodplain (MDBA 2017a) for each of the Basin-wide environmental Watering Strategy (BWS) regions (*Basin-wide environmental watering strategy* 2014) for 6 years, 2009, 2010, 2012, 2014, 2015 and 2016. A collective assessment of these findings by Basin States lead to a joint venture monitoring & evaluation project titled 'Improving evaluation and reporting on woody vegetation condition outcomes, at regional and Basin scales' (more information in Appendix C). The overall aims of the Joint Venture Steering Committee (JVSC) project were to:

- Become familiarised with the SCT and conduct a case study demonstration
- Improve knowledge of woody vegetation population demographics
- Report on stand condition model refinement

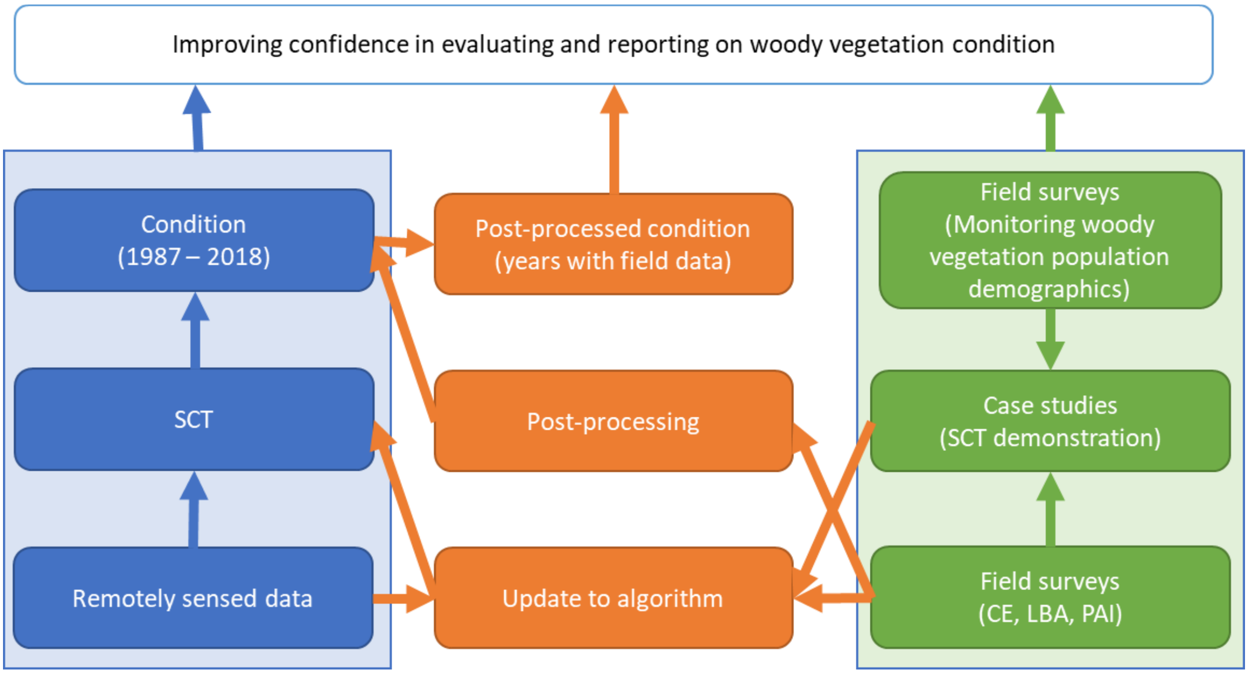


Figure 1. A conceptual diagram of the relationship between the MDBA's (blue) and VTAG's (green) and possible (orange) activities in relation to the SCT.

Unfortunately, not all Basin States were able to fully utilise the SCT due to its operational requirements and the difficulties associated with producing the input satellite imagery. This meant that some Basin States could not properly become familiarised with the SCT nor assess it to determine what improvements would be beneficial. Therefore, the MDBA committed to:

1. Develop the capability to run the SCT
2. Test this new capability through validation
3. Apply the tool to the Landsat imagery archive

This report details the methods and results of this process and then reassesses the future SCT refinement activities in light of the MDBA enhanced capability and understanding of the SCT.

1.1. Stand Condition Tool

The various interactions and applications of the stand condition tool are detailed in Table 1. The 2017 SCT software package, called the Murray–Darling Vegetation Monitor (MDVM), is a piece of software which maps conditions of native tree stands across the MDB. It requires Landsat summary images as input, and its outputs predictions of field variables, these are stand condition, plant area index (PAI), live basal area (LBA) and crown extent (CE). These predictions can be compared with relevant field data to determine the model fit using the in-built validation option. Additionally, field data can be used to post-process the model to better fit the observed conditions and thus produce more representative maps (*Murray–Darling Vegetation Monitor User's Guide* 2017). Due to the field data collected it should only be used where river red gum, black box or coolibah are dominant.

Table 1. Timeline of notable developments in the SCT.

TIMELINE	
2012	- Work for mapping condition of river red gums and black box in TLM sites done (Cunningham *et al.* 2013b) - This used PAI standardised to local maxima
2013	- Cunningham *et al.* (2013b) report published - Report on mapping floodplain vegetation types published (Cunningham *et al.* 2013c) - User's Guide for SCT (Murray Stand Condition Modeller – V1.08) published (Cunningham & Griffioen 2013)
2014	- Report on using the SCT (2013) - SCT with RapidEye option published (Cunningham *et al.* 2014)
2014	- SCT applied Basin wide in the development of the first Basin Wide Environmental Watering Strategy (MDBA 2014)
2015	- Report on stand condition assessments at Gunbower for 2013 published (MDBA 2015) - This was fed into the RapidEye model of the SCT
2016	- Cunningham confirmed in email that the epoch was '1 September of the previous year and the 31 December' of the desired year (Shaun Cunningham, personal communication, 24th June 2016)
2017	- Stand Condition Tool updated: Murray–Darling Vegetation Monitor (MDVM), - Newell *et al.* (2017) report - TLM stand condition assessments (Bennetts & Jolly 2017) - No mention of scaling or standardising PAI
2018	- Field protocol updated (Cunningham *et al.* 2018)
2019	- The Hindcast (i.e. this report)

1.1.1. Development

The SCT algorithms were trained using:

- Field data (crown extent, live basal area, plant area index and stand condition)
- Three constant images of tree presence
- River red gum, black box and coolibah distribution maps (Type)
- Median Autumn Landsat image constant (derived from 2008, 2009, 2013, 2014 and 2015).
- Image summaries (median, upper and lower quartile) to match the year in question.

The model itself is a bagged ensemble of twenty multi-objective regression trees. Each tree estimates the four field data variables and these estimates are averaged across the ensemble for each pixel of the output map. SCT used multi-threaded computing to efficiently sample each image, pass all the values through the bagged-regression trees. These determine the four condition estimates. These values are then re-assembled into a coherent multi-layer map (*Murray–Darling Vegetation Monitor User's Guide* 2017).

1.1.2. Options

The SCT has three options (Table 2), each with its own underlying algorithm. For this report we consider only the 'Veg Monitor Map All Years' as this is the more complex model and produces slightly better results regardless of the year of the image (for 2008, 2009, 2013, 2014 and 2015) (*Murray–Darling Vegetation Monitor User's Guide* 2017).

Table 2. The three models contained in the SCT.

Model	Input
Veg Monitor Map All Years	Median Landsat image (PCT_50), the lower quartile (PCT_25) and the upper quartile (PCT_75)
Quick Veg Map All Years	Require only a single Landsat image, be it the median image (PCT_50) or some other 6-band Landsat image
Veg 2016 and 2017 Only	Require only a single Landsat image, be it the median image (PCT_50) or some other 6-band Landsat image

1.1.3. Post-processing capability

From the *Murray–Darling Vegetation Monitor User's Guide* (2017): 'post-processing of condition maps may be performed if validation sites are provided. The post-processing step may re-align the model output with observed conditions that will further improve model performance. This is achieved by providing a simple inverse-linear regression model overlay onto the current model output. The validation data must supply site positions and observations of the four condition values. These are supplied to the MDVM as a CSV file which compares these values to the current modelled values. A least-squares inverse-linear regression is then calculated for each condition index and applied to the current model map resulting in a corrected model map'.

1.2. Google Earth Engine

Google Earth Engine (EE) is a cloud-based platform available for scientific analysis and visualization of geospatial datasets, for academic, non-profit, business and government users (Google Earth Engine 2019a). It contains petabytes of information, including imagery from Landsat, Sentinel, MODIS and PALSAR missions. Since this imagery does not need to be downloaded when it is processed in the EE code editor it saves considerable amounts of personal storage. The processing capability generally makes using EE for analysis far quicker than similar tasks run on local machines, in programs such as ArcMap. The code editor is a Javascript API that is commonly used for processing and most of the documentation is in Javascript. Alternatively, a python API is available for users familiar with that language. Access to EE is restricted and a sign in is required. With the exception of commercial users, access is generally free of charge.

2. Aims

The purpose of this analysis is to test the validity of the SCA tool and assess its usefulness as a method of measuring changes in tree stand condition.

The main questions being investigated are:

1. How well does the Hindcast data (EE generated RS data) compare to the 2017 Evaluation data (GA annual summary)?
2. How well do the Hindcast condition results match the 2017 Evaluation condition results?
3. How well do other measures of condition (such as NDVI) correlate with the Hindcast condition results?

Answering these questions generates a nuanced understanding of the performance of the SCT. This is information is a critical consideration in the application of the tool. The next application by the MDBA of the information generated by the tool will be in the assessment of environmental outcomes in the 2020 Basin Plan Evaluation.

3. Methods

2.1 Data

The data requirements of the tool are split into three distinct categories: 1) the field observations, 2) the constant data and 3) the remotely sensed data for each year.

2.1.1 Field Data

There are three field data variables collected:

1. Plant Area Index (PAI) – the area of canopy per square meter, calculated from hemispherical photos.
2. Crown Extent (CE) – the percentage of canopy that is alive.
3. Live Basal Area (LBA) – calculated from diameter of tree trunks at breast height.

The three field data variables are used to calculate the Stand Condition Score (SCS) (Equation *Eq2*), referred to throughout this document simply as 'condition'.

$$\text{Stand Condition Score (SCS)} = \frac{(2\text{CE} + 5\text{PAI} + 0.1\text{LBA})}{3}$$

Field data collection is not undertaken by the MDBA but rather by the relevant Basin State authorities, and the method of collection is described in the field protocol document (Cunningham *et al.* 2018).

In the delivery of the SCT to the MDBA, the MDBA was provided with field data for 2009, 2010, 2014, 2015 and 2016. Some field data had been used to train the SCT while some had not and so we refer to each of these as the 'train' and 'test' datasets, respectively.

2.1.2 Constant data

2.1.2.1 In-built vegetation masks

There are two static maps, native vegetation and woody vegetation that can be used to mask the input data in the SCT i.e. the condition scores will not be calculated for excluded areas.

In the SCT the mask can be set to be either more inclusive (risk of including non-native vegetation) or more exclusive (risk of excluding areas of interest) and it is up to the user to decide a suitable masking threshold.

Table 3. The different bands in the two vegetation masks.

Bands	Native mask (MDBAVegNative)	Woody mask (SppModels)
1	BlackBox	NativeProb
2	Coolibah	NonNativeProb
3	Dryland	VegClass
4	Lignum	WaterProb
5	RiverCooba	
6	RiverOak	
7	RiverRedGum	
8	Wetland	
9	Woody	

It was decided that no mask would be applied within the SCT so that condition across the entire basin would be calculated. Therefore, after calculating condition, masking by different and perhaps improved datasets would be possible.

Due to the MDBA's interest in monitoring three key tree species we masked using the vegetation layer described in Cunningham *et al.* 2013 (Table 4). This also meant no decision about using a threshold had to be made.

2.1.2.2 BWS regions

The results for changing condition are reported on by different areas, these are the Basin-wide environmental Watering Strategy (BWS) regions (Murray–Darling Basin Authority 2018) shown in Table 4.

Table 4. The 20 BWS Regions and whether they predominantly lie in the North or South of the MDB.

North	South
Barwon-Darling	Campaspe
Border Rivers	Eastern Mt Lofty Ranges
Condamine-Balonne	Goulburn-Broken
Gwydir	Lachlan
Macquarie-Castlereagh	Loddon
Moonie	Lower Darling
Namoi	Murray
Nebine	Murrumbidgee
Paroo	Ovens
Warrego	Wimmera-Avoca

2.1.2.3 Combined floodplain and vegetation layers

The vegetation layer described in Cunningham *et al.* 2013 (Table 3) provides an understanding of where the three target species occur. However, there are areas where it misclassifies pixels, particularly street trees in suburbs. Therefore, it was decided to use other mapping layers to refine the layer. There are some areas that have been mapped as forests predominantly containing the target species, and at the basin-scale this has been done by the Australian National Aquatic Ecosystem (ANAE). However, species might also occur outside these areas, particularly along channels. As used in the 2017 application of the SCT, the managed floodplain layers produced as part of the 2014 BWS was used to mask the outputs.

To define the full extent of the floodplain, three layers were used (MDBA 2017b):

- Floodplain undeveloped valleys
- Managed floodplain constraints relaxed
- Managed floodplain current

The 2017 ANAE (Department of Environment and Energy 2017) contains several layers, two of which were determined to be useful:

- Interim Western NSW Floodplain ANAE
- Wetlands ANAE 20171025

From each of these layers, only some of the polygons cover areas of relevant to the Hindcast. In the Wetlands ANAE layer, there are 67 attributes under 'ANAE_DESC' and only five of these were found to contain the three target species (Table 5). In the Interim NSW layer, under 'ANAE_DESC', there were 16 attributes and nine of these were relevant (Table 5).

Table 5 Table showing the attributes used to determine which polygons would be used from each of the ANAE layers to create the floodplain vegetation mask.

Interim Western NSW Floodplain ANAE	Wetlands ANAE 20171025
• River red gum forest floodplain • Black box woodland floodplain • Black box forest floodplain • Upland river red gum forest floodplain • Upland black box forest floodplain • River red gum woodland floodplain • Coolibah woodland and forest floodplain • Upland coolibah woodland and forest floodplain • Upland black box woodland floodplain	• Temporary river red gum swamp • River red gum riparian zone or floodplain • Black box riparian zone or floodplain • Temporary black box swamp • Coolibah riparian zone or floodplain

Each of the layers (three floodplain and two wetlands) were converted into rasters using the 'polygon to raster' tool in ArcMap. The settings were such that the resulting raster was:

- snapped to the extent of the species map
- the same extent as the species map
- had a cell size of 0.00025 (same as the species map)
- cells were assigned using the maximum combined area

The five resulting rasters were then uploaded into EE. The five layers (select attributes from the wetland layers) were combined. This essentially means that pixels from the species map were selected if they sufficiently overlapped with any of the five layers. The final mask that was used to calculate condition area was the intersection of the species map and the combined 5 layers.

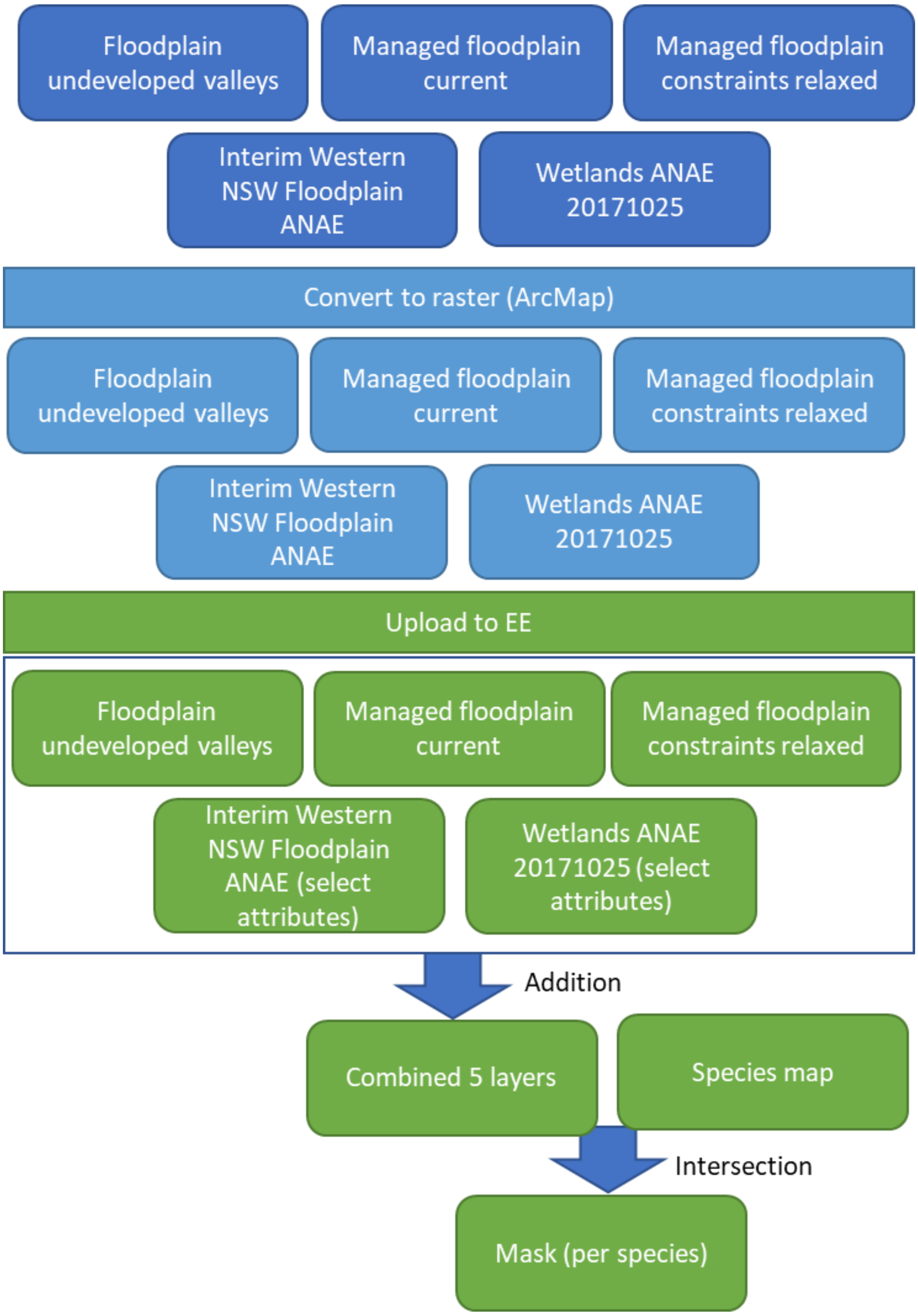


Figure 2 Schematic showing the production of the mask.

2.1.3 Remotely sensed data

Landsat satellites are owned and operated by the United States Geological Survey. The images taken by the satellites are freely and publicly available online.

In EE Landsat raw scenes, top-of-atmosphere, surface reflectance and a variety of other products are available. These products are sourced or generated by Google using Docker images supplied by the USGS. The products used in this report were the USGS Landsat 5, 7 and 8 Surface Reflectance Tier 1 collections (Table 6).

Table 6. The Landsat collections and their date avaliability.

Collection	Start	End
USGS Landsat 5 Surface Reflectance Tier 1	1984-01-01	2012-05-05
USGS Landsat 7 Surface Reflectance Tier 1	1999-01-01	Ongoing (2003-05-30*)
USGS Landsat 8 Surface Reflectance Tier 1	2013-04-11	Ongoing

*Landsat 7 had a scan–line correction malfunction on the 30th May 2003, and therefore images taken after this date have been excluded from analysis.

To produce surface reflectance, the images must first be processed to Level- 1TP (Precision and Terrain Corrected) i.e. radiometrically and topographically corrected (USGS 2018c).

Landsat 5 and 7 surface reflectance has been atmospherically corrected using LEDAPS (USGS 2018a) and have cloud, shadow, water and snow masks as produced by CFMASK (USGS 2018b), as well as a per-pixel saturation mask.

Landsat 8 surface reflectance has been atmospherically corrected using LaSRC (USGS 2018a) and also has a cloud, shadow, water and snow mask as produced by CFMASK (USGS 2018b), as well as a per-pixel saturation mask.

In comparison Geoscience Australia directly downloads Landsat imagery from the satellites and then conducts their own atmospheric and topographic corrections. Their surface reflectance product is produced using Nadir-corrected BRDF Adjusted Reflectance, where BRDF stands for Bidirectional reflectance distribution function (Geoscience Australia 2018). One of the differences between USGS and GA-processed Landsat imagery is that the resolution of the pixels is 30m^2 for the USGS, and 25m^2 from GA.

A summary of the process is provided below (Table 7)

Table 7. The difference in processing satellite imagery between Geoscience Australia and Earth Engine.

	Geoscience Australia	Earth Engine
Landsat collection	Download raw data from Landsat satellites	Surface Reflectance Tier 1 supplied by the United States Geological Survey (USGS)
Surface reflectance processing	nbar	LEDAPS for Landsat 5 and 7 LaSRC for Landsat 8 (see USGS 2019)
Cloud masking	fmask	Bitmask
Pixel size	25m	30m
Projection	Unprojected using EPSG 4326 GCS_WGS_1984	Landsat tiles as supplied are projected into UTM zones

		using the UTM North projection parameters Google Earth Engine unprojects the data into EPSG 4326, input to the SCT is the same
Tiling schema	1 degree	In WRS path and row in EE. The mosaic image exported into Google Cloud is tiled variably as a result of parallel processing in EE.

2.1.3.1 Epoch

Previously Geoscience Australia used 16-month image stacks, starting in September and finishing in on the first of January two years later (see Table 8). Field data is generally collected sometime in April or May; therefore the 16-month epochs includes images 7 months either side of the field data (Shaun Cunningham, personal communication, 24th June 2016). Therefore, for the hind-cast we have used the same epoch structure.

Table 8. The epochs originally processed and used in the 2017 Evaluation. Center year corresponds to when the field data was collected.

Epoch	Start	Center	End
20080901-20100101	2008	2009	2010
20090901-20110101	2009	2010	2011
20130901-20140101	2013	2014	2015
20140901-20160101	2014	2015	2016
20150901-20170101	2015	2016	2017

Due to the Landsat 7 SLC-error, when the affected images are excluded there is a gap between the 5th of May 2012 (end of Landsat 5) and 11th of April 2013 (start of Landsat 8) two epochs are incomplete (Figure 3) and therefore excluded from our analysis.

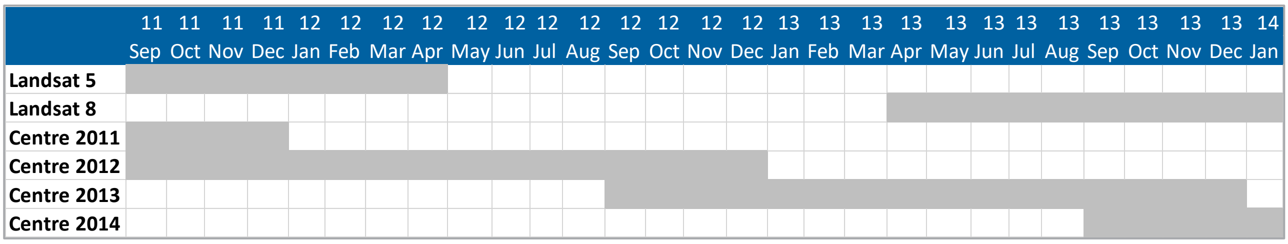


Figure 3. Schematic showing the period where there is a gap in Landsat avaliability and the affected epochs (Centre 2012 and 2013).

2.1.3.2 Boundaries (spatial)

GA used 1-degree tiles to cover the North and South Basin. There are several areas where this does not cover the basin (Figure 4), however in order to hind-cast the work previously done we kept the same schema.

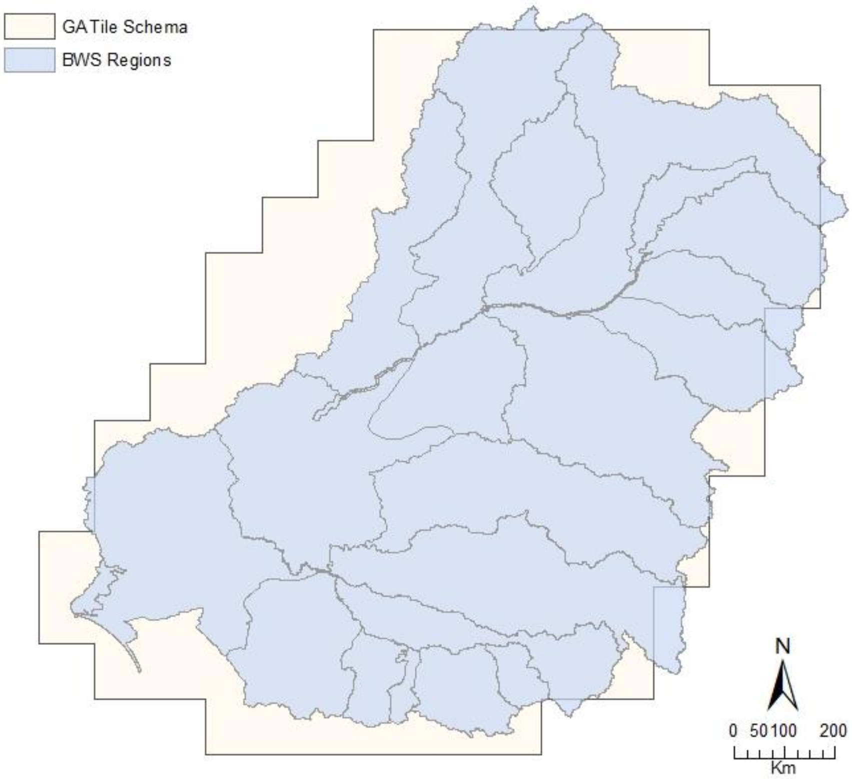


Figure 4. The coverage used in the 2017 Evaluation (beige) and the MDB (blue).

2.2 Hindcast

Since all Landsat imagery is now widely available and does not require special or expensive software to process it was decided that it would be possible and beneficial to run the SCT for the entire Landsat collection. As the SCT is utilising all historical data, it is dubbed a hindcast, run in 2019. The five epochs that were part of the 2017 Evaluation were re-run so that it could be checked that the new processing methods were performing comparably with the methods used at that time.

2.2.1 – Calculate percentiles

Calculating percentile images was completed in EE. Landsat 5 and 7 surface reflectance has been atmospherically corrected using LEDAPS (United States Geological Survey 2019a) and has cloud, shadow, water and snow masks as produced by CFMASK (United States Geological Survey 2019b). Optical satellite imagery is often affected by clouds, which block out the signal from the land's surface. Therefore removing, usually called 'masking', cloudy pixels is common practice and leaves behind a gap in the image rather than the cloud signal. For analysis using time series there are usually other images taken when cloud was not present which fill in the gaps. Due to the different sensors on Landsat 8 compared to Landsat 5 and 7 there are different methods to removing cloud. For Landsat 5 and 7 the code selects pixels where the cfmask has determined that there is a high chance that cloud, or cloud shadow are present and masks them. To remove edge effects, this code also determines whether all pixels bands have values present, if not those pixels are masked. For Landsat 8 the code selects pixels where the cfmask has determined that there is no cloud or cloud shadow present and masks them.

The SCT model (Veg Map Monitor All Years) uses summary images as input. Summary images are produced by stacking all images in each epoch and calculating for each pixel the 25th, 50th and 75th percentile (Figure 5). To calculate condition of a given year, percentile images were created using an epoch that starts in the September of the previous year and ends at the end of the year of interest. For example, to determine condition in 2018, images in the date range 01-09-2017 to the 01-01-2019 would be used (images on the last date would not be included). After the clouds have been masked percentile images for each of the epochs are created. This finds the percentile value of each pixel for each band within the epoch (Figure 5).

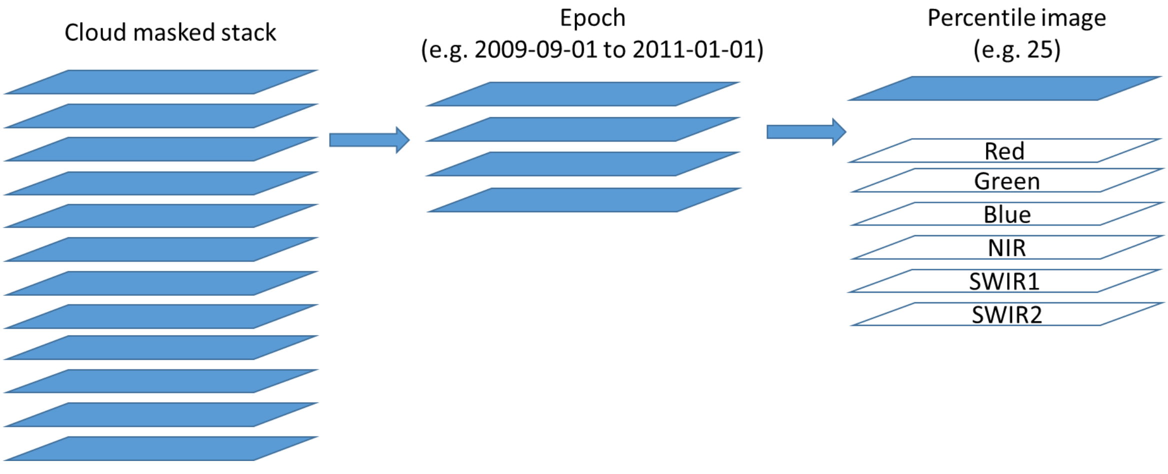


Figure 5. Schematic showing the process of calculating the percentile images.

The percentile images are then exported out of EE and onto the Google Cloud. After exporting is completed, these can then be downloaded from a web browser directly, or batch downloaded using the Google SDK (Appendix B).

2.2.2 – TIFs to ERS

The percentile images produced in EE are TIF files. The SCT however require ERS files, therefore they must be converted.

2.2.3 – Run the stand condition tool

Each epoch is run in the SCT separately. A batch option is available, but some issues were encountered running more than 3 sequential processes. The map type used was 'Veg Monitor Map All Years' and no masking options were selected. The output of the tool is a four-band raster (condition, LBA, PAI, CE) in ERS format. This takes approximately 3 hours to complete for each epoch. Information on how to run the SCT is available in the user's guide (*Murray–Darling Vegetation Monitor User's Guide* 2017).

2.2.4 – Calculate condition in EE

Calculating area of condition was completed in EE. The SCT output condition maps are in ERS format so they must be converted back into TIF files for ingestion into EE. The script used to calculate the areas in EE is available in Appendix E.

To calculate condition per species per BWS region it was determined that moving the images into EE for calculation would be relatively quick, which if necessary, would make re-calculating areas much easier than in ArcMap.

As EE only accepts GeoTIFF of TFRecord images, the band condition output from the SCT was converted from ERS to TIF format. This was completed manually in ArcMap using export data. Once the conversion was complete the raster was then uploaded into EE using the asset manager (Google Earth Engine 2019b).

The script used to then count area of different condition classes is presented in Appendix E. The condition image has values ranging from 0 to 10, therefore a new image is produced on the fly to have pixel values correlating to its condition class (as presented in Table 9) from *Stand Condition Assessment of Forests and Woodlands of Gunbower Forest – 2013* (2013).

Table 9. The relationship between the condition score and the descriptive 'class'.

Class	Bin
Good	> 8 – 10
Moderate	> 6 – 8
Poor	> 4 – 6
Degraded	> 2 – 4
Severely degraded	0 – 2

The species raster is then split into each of its components, one image each for river red gum, black box and coolibah. Each of these is then multiplied by the inbuilt ee.Image.pixelArea() function provided in EE which uses Lambert's Equal Area to assign each pixel a value equal to its area in metres squared. For each condition class in the binned condition image, all the underlying pixels in the species area raster are summed, therefore producing the total area for each species for each BWS region. The output is re-formatted for readability and then exported to Google Cloud as a CSV.

Figure 6. Schematic of how binned condition area is calculated.

2.2.5 Post-processing

We ran the post-processing on EE results for 2009, 2010, 2014, 2015 and 2016 using field data. These were the 'test' datasets as opposed to the 'train' datasets which were used to train the SCT, provided by ARI. Post processing can be done for a given year when field data is available. After a model has been run in the SCT, there is a 'Validate' option which allows the user to select a CSV of field data in order to apply a correction to the current modelled map. This is calculated using a least-squares inverse-linear regression for each field variable and condition (*Murray–Darling Vegetation Monitor User's Guide* 2017).

2.3 NDVI check

Normalised difference vegetation index (NDVI) (Rouse *et al.* 1974) is a remote sensing tool commonly used to assess vegetation condition. NDVI is calculated using equation (Eq 1), and uses the fact that plants absorb photosynthetically active radiation, here measured as red light, and re-emits near infrared light.

NDVI was not used as an input into the stand condition tool, although the raw red and near-infrared bands are included. Therefore, it was decided that NDVI could provide a reasonable 'sanity check' on the results from the SCT. NDVI values were calculated for each year to check that any trends present in the SCT results can be corroborated with another indicator. Calculating NDVI was completed in EE. The script used to complete this task is available in Appendix E.

$$NDVI = \frac{NIR - Red}{NIR + Red} \quad \text{Eq 1}$$

The collections were cloud filtered as described in *2.2.1 – Calculate percentiles (*except for 1999 which experienced an internal error). The only difference in processing was that instead of using the 16-month epoch, a full calendar year was used, so that is aligns with the centre year of the epoch.

4. Results

There are several results from this report, they answer the questions of:

1. How well was the Hindcast data (EE generated RS data) modelled in comparison to the 2017 Evaluation data (GA annual summary)?
2. How well do the hindcast condition results match the 2017 Evaluation condition results?
3. How well do other measures of condition (such as NDVI) correlate with the Hindcast condition results?

Answering these questions generates a nuanced understanding of the performance of the SCT. This is information is a critical consideration in the application of the tool. The next application by the MDBA of the information generated by the tool will be in the assessment of environmental outcomes in the 2020 Basin Plan Evaluation.

3.1 Post-processing

Post-processing by using the 'Validate' option in the SCT (*Murray–Darling Vegetation Monitor User's Guide* 2017) aims to make the model fit better to extreme values. This is the second 'learning' component of machine learning, whereby the tool uses training data from field observations to perform and compare predictions based on the model, both with and without the field observations. It essentially informs the tool operator whether the additional field observations have improved model predictions.

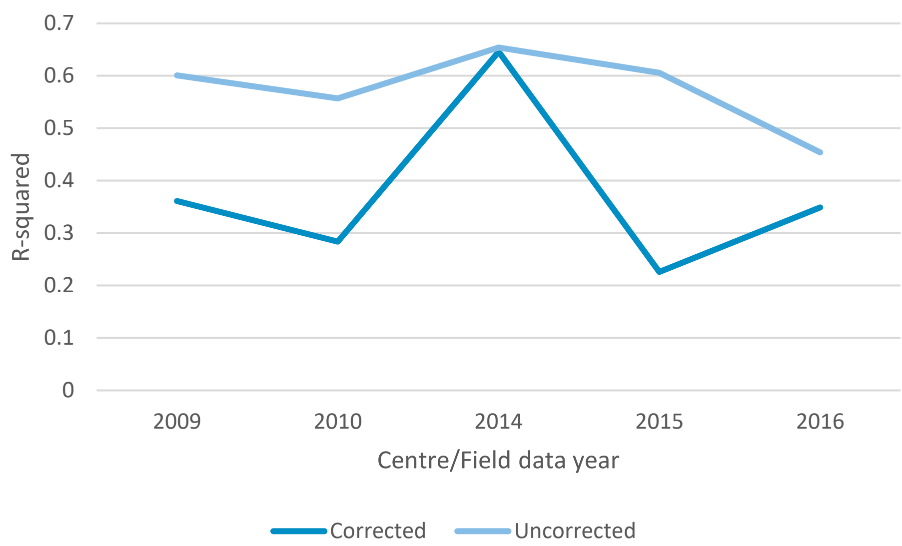


Figure 7: Graph showing the correlation between field data and remotley-sensed data for corrected and uncorrected condition maps.

As illustrated in Figure 7, validation results showed that the first four years of training data resulted in a decrease in model fit. For 2015 a small increase in model fit after post-processing is observed. This is possibly due to the small number of test points for each year (Table 10), and in theory using a small number of field data points could skew the result. Therefore, we did not use the post-processed results. The only time it might be applicable to use post-processing, is using lots of field data for correcting later (or indeed earlier) years where the SCT had not already been trained with data from that year.

Table 10. The number of field data sites for each year that were used in either testing or training the model.

Centre year	Test	Train
2009	35	140
2010	35	140
2013	33	139
2014	93	383
2015	30	120
2016	120	488

3.2 Comparisons

The R-squared values for the Hindcast and 2017 Evaluation data were calculated using the Validate option in the SCT for both the ‘test’ and ‘train’ data. These correlations tell us how well the Hindcast (using the GEE images) condition results match field observations, compared to results obtained in the 2017 Basin Plan Evaluation (which used images provided by GA). The second set of correlations tell us how well the hindcast results match the 2017 Basin Evaluation results, giving us an indication of whether the more accessible hindcast method could be used for future condition assessments.

3.2.1 Correlation to 2017 Evaluation

Due to the different processing methods used to create the input percentile images (e.g. cloud masking), there are several differences in the SCT condition output images. For example, the number of pixels used to calculate the percentile at any point could be different due to the different sensitivities of the cloud masks, or the decision by MDBA to remove SLC-error affected Landsat 7 images. Due to the different sizes of the pixels it was not possible to compare on a pixel by pixel basis for each input image. Therefore, it was decided to compare instead how well the SCT model fit to the field data for each of the years.

The 2017 Evaluation data consistently performs better than Hindcast data, but this difference is always less than 8% of the R-squared. 'Train' data usually performed better than 'test' data, most likely because the model was trained with the 'train' data and should therefore fit it better. The year 2016 performed the worst out of all the years tested for the Hindcast and 2017 Evaluation data.

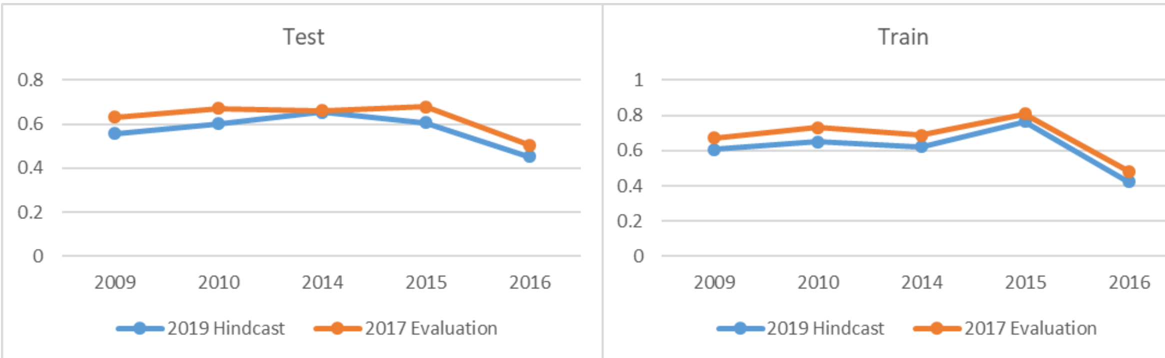


Figure 8. Graph showing the difference between the Hindcast (performed in 2019) and the 2017 Evaluation condition R^2 values for test data (left) and train data (right).

3.2.2 Comparison to 2017 Evaluation

This set of comparisons tell us how well the hindcast condition results correlate with the 2017 Basin Evaluation results. River red gum has a standard deviation of approximately 4%, while coolibah and black box had a much larger deviation of approximately 11% and 13%. The 2017 Evaluation results for coolibah and black box were put into two bins (0-6 and >6-10), as opposed to the five bins used for river red gum.

The larger bins used for those two species could account for the greater variation between Earth Engine results and the original results.

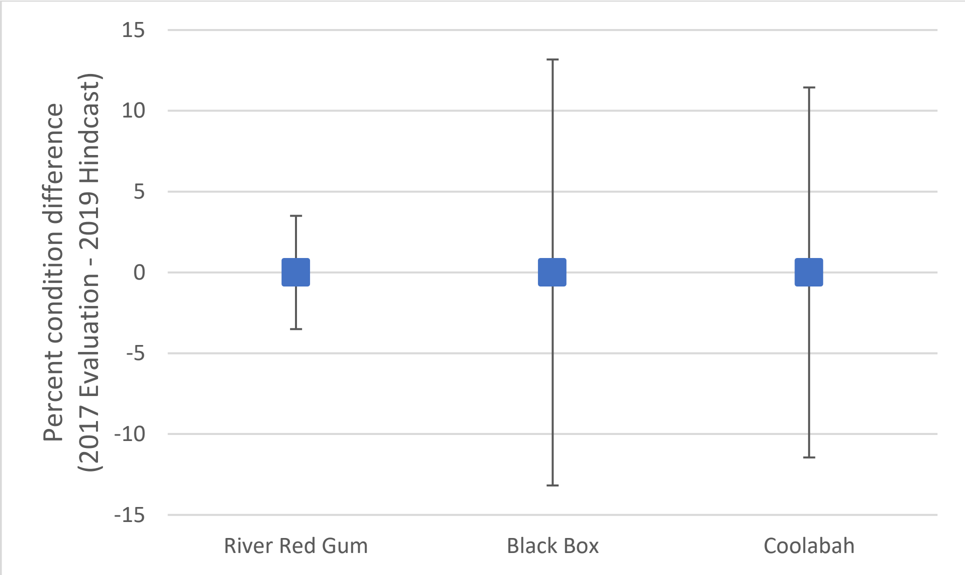
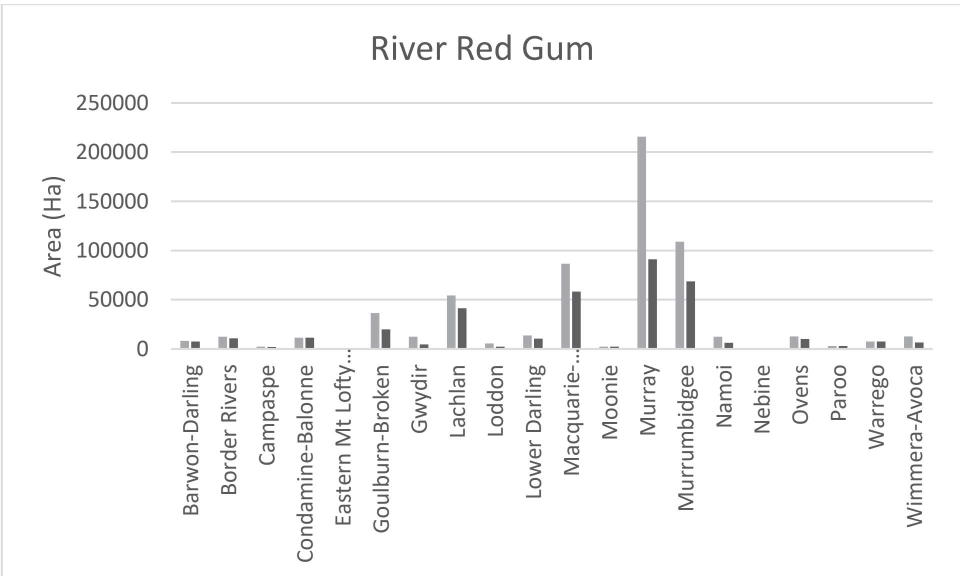
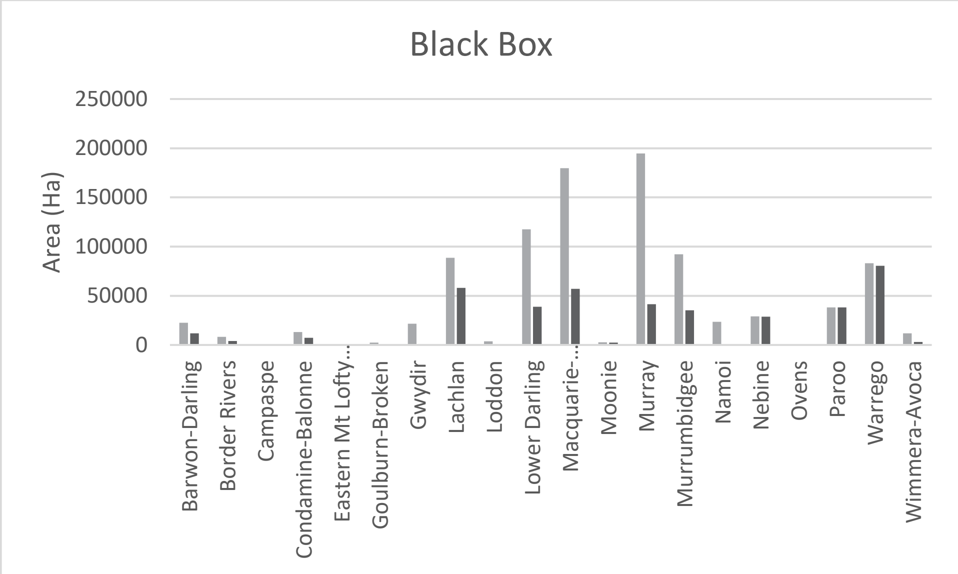


Figure 9. The differences between the 2017 Evaluation condition results and our Hindcast condition results.

Additionally, the 2017 results were clipped by the managed floodplain (MDBA 2017a) layer to exclude areas that can't receive environmental water within current operational constraints. However, the Hindcast results were clipped to a combination of floodplain and forest layers (see Chapter 2.1.2.3). River red gum usually occurs closer to the river and would therefore is less likely to be excluded by using any floodplain mask. However, when black box and coolibah occur higher on the floodplain are more likely to be excluded in floodplain extent masks.





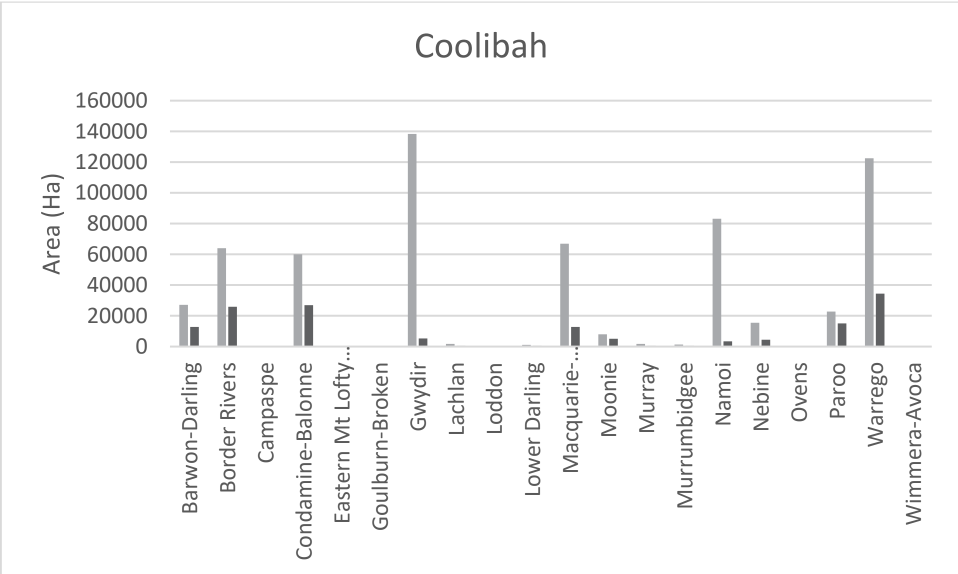


Figure 10. The coverage (Ha) of the three tree species in each BWS Region used in the 2017 Evaluation and the 2019 Hindcast data.

3.2.3 NDVI check

Average NDVI values were compared to the combined moderate to good condition scores i.e. percent of area with a condition score greater than 6. Black box had the strongest correlation (R^2=0.43), followed by coolibah (R^2=0.29) and river red gum with the weakest correlation (R^2=0.12).

River red gum congruence between NDVI and condition appeared to depend strongly on location, which determines total area. For example, the Macquarie-Castlereagh had an R^2 of 0.56 (1622 km^2) while the Warrego had one of 0.03 (86 km^2). Smaller areas are likely to have a smaller spread of values therefore a reasonable correlation between NDVI and condition cannot be drawn. Furthermore, the values of river red gums tend to be higher for both NDVI and condition, and this congregating effect can also make it more difficult to draw out a reasonable correlation. In contrast, black box NDVI to condition congruence does not appear to be affected by location. In general, coolibah also appears unaffected by location, except for the Lower-Darling BWS Region.

In conclusion it is difficult to summarise the correlation results between NDVI and condition scores as they depend on the species and location. Additionally, the effect of comparing percent area condition to average NDVI across the whole area is likely having an impact and that trend rather than absolute values is more important. The differences in NDVI and condition are probably best considered on a case by case basis.

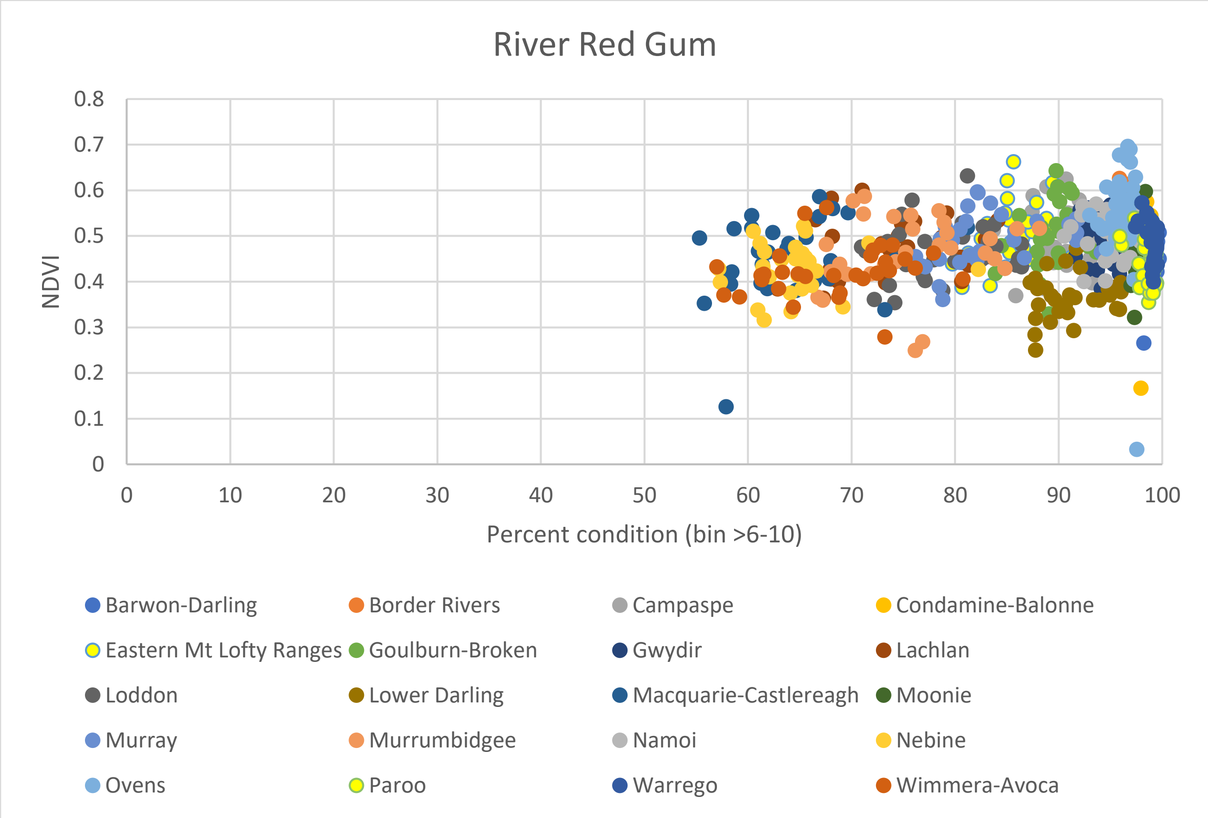


Figure 11. The correlation between NDVI and river red gum stands with a condition greater than six.

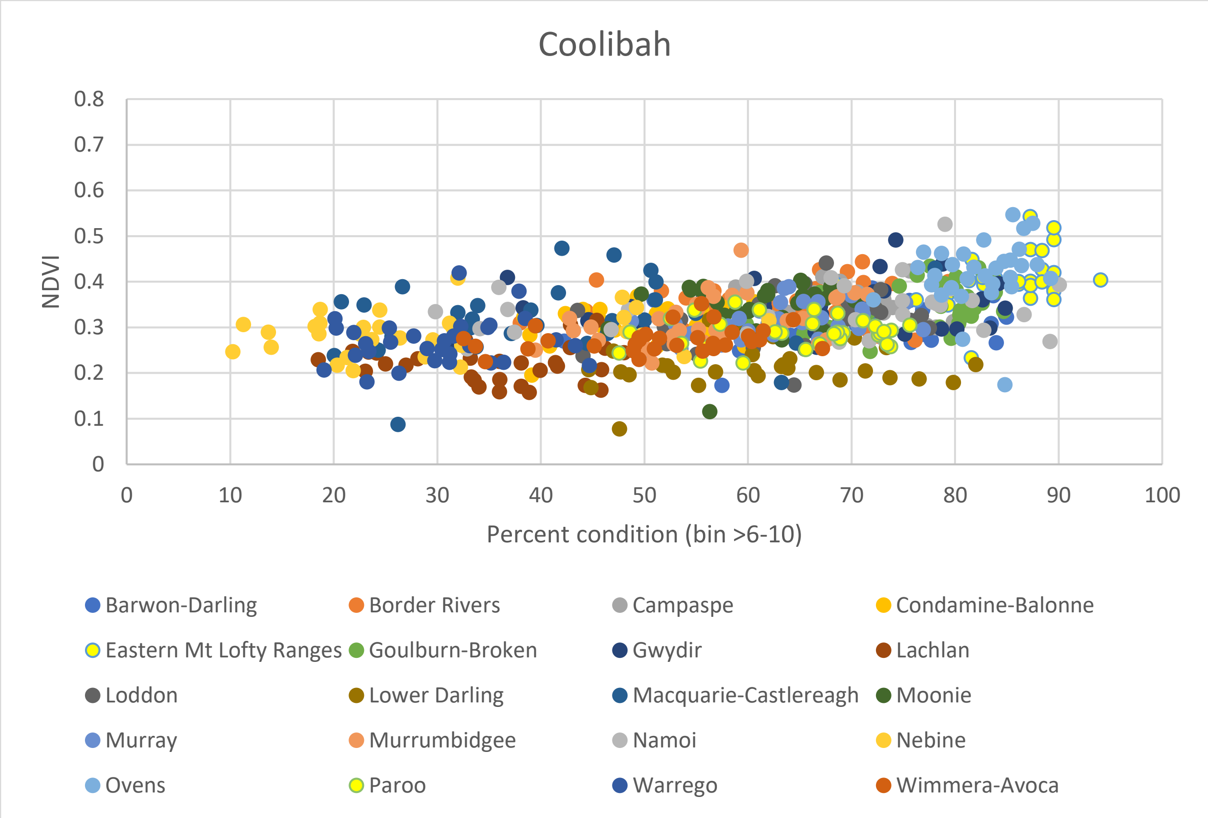


Figure 12. The correlation between NDVI and coolibah stands with a condition greater than six

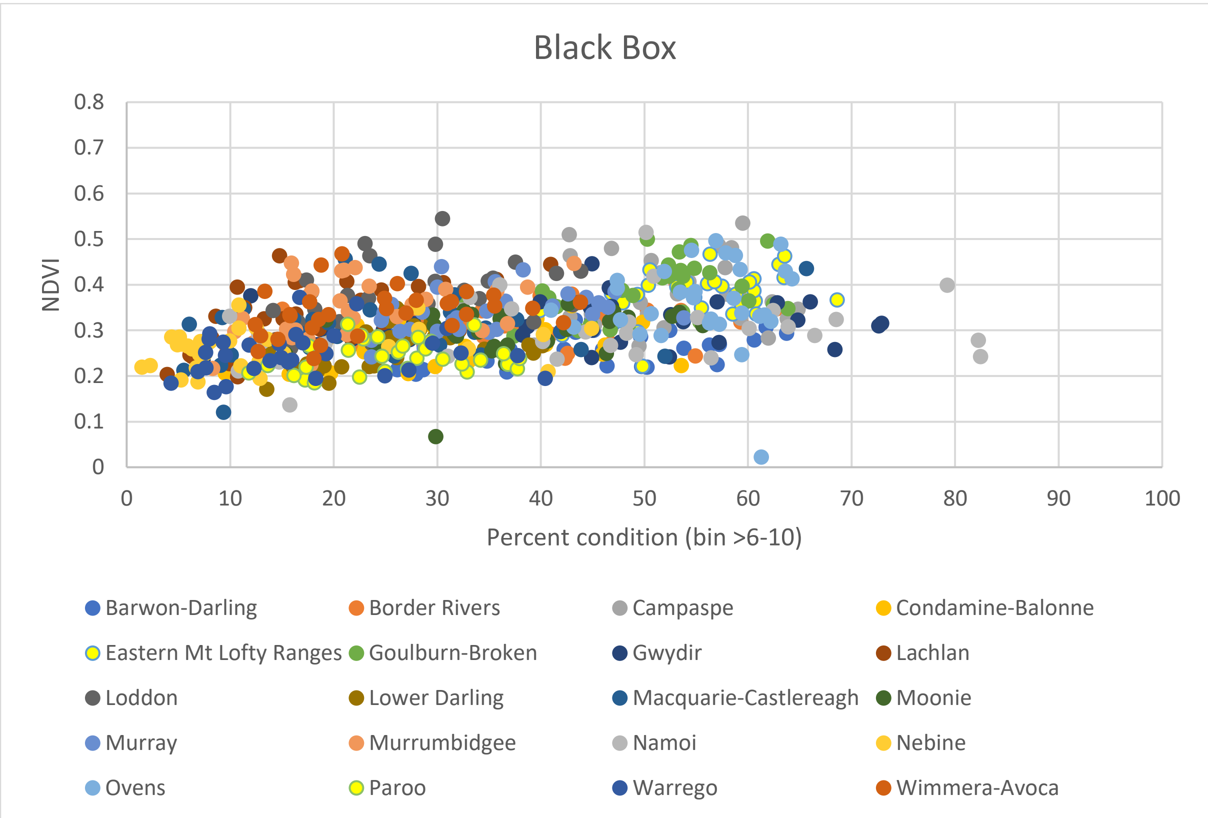


Figure 13. The correlation between NDVI and black box stands with a condition greater than six

4. Discussion

As previously stated, this report aimed to answer the following questions:

1. How well does the Hindcast output match field observations, compared to the 2017 Evaluation method output? (i.e. comparison of the models)
2. How well does the Hindcast output match the 2017 Evaluation output? (i.e. comparison of the modelled results)
3. Is the Hindcast output comparable to other measures of condition, like the NDVI?

The analysis showed that the hindcast output (using GEE derived imagery) performed well against the field observations used as test data. Similarly, the hindcast output performed well in comparison to the 2017 evaluation method output (which used imagery supplied by Geoscience Australia). This suggests that running the SCA tool using GEE derived imagery produces results comparable to those supplied by Geoscience Australia, therefore providing a more cost-effective method for assessing native vegetation condition, for the MDBA and other Commonwealth and state agencies.

There are a number of issues related to the SCA tool that have been known and documented over the last few years. The overarching goal of this discussion is to help tease apart the intertwined issues with the SCT which should help guide prioritisation of tasks to improve the use of the SCT. Having trustworthy results from the SCT can then be used to report tree stand condition, as part of but not limited to, assessing Basin Plan objectives, future Basin environmental watering strategy updates, JVSC/MDBA key evaluation questions and other reporting requirements.

These issues are applicable to the SCA tool itself, and not specifically to the GA imagery method used as part of this hindcast analysis. These are broadly divided into technical issues that affect the accuracy and reliability of the output, and into operational issues that affect the effectiveness of the tool to inform decision-making about environmental watering. These are covered respectively in the sections below. The final section aims to prioritise the tasks suggested to address the issues.

Previous assessments have been made, including after the 2017 Interim Basin Plan Evaluation, of the underlying issues and potential improvements with the SCT. Numerous possible improvements to the SCT have been identified. In light of recent technological advances, the MDBA's enhanced capacity to operate the tool and the learnings gained through producing the hindcast timeseries; the program of potential improvements and refinements have been reconsidered. Broadly these include but are not limited to the discussion below.

4.1 Technical Issues and limitations

There are several technical issues, however, the majority of these are inherent to the SCT itself, and not a direct issue with the hindcast method.

4.1.1 Field data

As noted by Newell *et al.* (2017) mathematical models of any kind rely heavily upon the quality of the dependent data. If the data provide a poor representation of the ranges and expressions of the features that are to be modelled, then the output will not be useful. The 2017 version of SCT has been trained with field data from 2009, 2010, 2014, 2015 and 2016. Field data collected outside the initial training data set can be used in three ways:

1. To understand how well the SCT output models the conditions on ground, either by
    a. using the 'Validate' option provided in the SCT, or
    b. manually intersecting field data with model outputs, to generate correlation and R^2 values.
2. To post-process the SCT output so that the model better fits the field data. This is done by using the 'Validate' option, then using the 'Apply Map Correction' option. The apply map correction option uses inverse-linear regression which will generally stretch the upper and lower boundaries of the model values to the corresponding field data values, while not substantially changing the intermediate modelled values (Newell *et al.* 2017) (although small sample sets can result in worse model fit see Results – Post-processing 3.1).
3. In the future a newly trained algorithm may be developed using a larger more diverse set of field data which should improve the accuracy of the SCT.

From this it is evident that the SCT does not replace the need to collect field data, but it is significantly improved by, and enhances the utility of the field data collected. Given the variability of the climate in the Murray–Darling Basin, there may be a high chance that the model, which was trained with only six years' worth of field data, does not reflect all possible conditions observed on the ground. Validation data will be required on an ongoing basis to ensure the model is continuously improved.

To this end we suggest that more field data capturing various conditions continue to be collected. In order to ensure field data with a spread of conditions is recorded, careful consideration of field sites perhaps involving stratification is necessary bearing in mind that the SCT is trying to correlate field data to what the satellite can detect. If there is insufficient field data collected for a certain condition, the model will naturally struggle to predict that condition.

Another issue to consider is how the field data is interpreted by the model. The model uses satellite imagery to make inferences about the condition, however, the criteria used for these inferences may lead to spurious correlations. For example, pixels containing small and healthy trees may be erroneously classed as being in poorer condition, due to a higher percentage of bare soil cover. Similarly, pixels dominated by green undergrowth may 'hide' unhealthy trees, resulting in a condition score that is erroneously positive. Discrepancies in field data which does not consider the undergrowth, and satellite-derived models which do are undoubtedly going to occur. However, while these issues might be important to consider when creating a model, they might not matter so long as the model fits the field data.

4.1.2.1 Plant Area Index

Although PAI has been defined differently in a variety of literature they appear to be in the same vein, e.g. from Bennetts and Jolly (2017) it is defined as 'the area of leaves and stems relative to a unit of land'. PAI is important because it is thought to reflect a plants' biophysical and ecological processes (Asner et al., 2003). For the purposes of this report, and use in the SCT, PAI is determined by taking a hemispherical photo of canopy then calculating the area of plant per square meter using the LAI option in winphot (ter Steege, 1996) (for more information on calculating PAI see the field protocol by Cunningham *et al.* 2018).

There are a few issues and misunderstandings with the PAI in relation to the SCT:

1. Equivalency to canopy cover
2. Scaling
3. Standardizing

Since PAI is the area of plant material per square meter, it can be greater than one for example if there is overlapping canopy (Figure 14). Therefore, although PAI and canopy are inherently linked, the relationship is not linear.

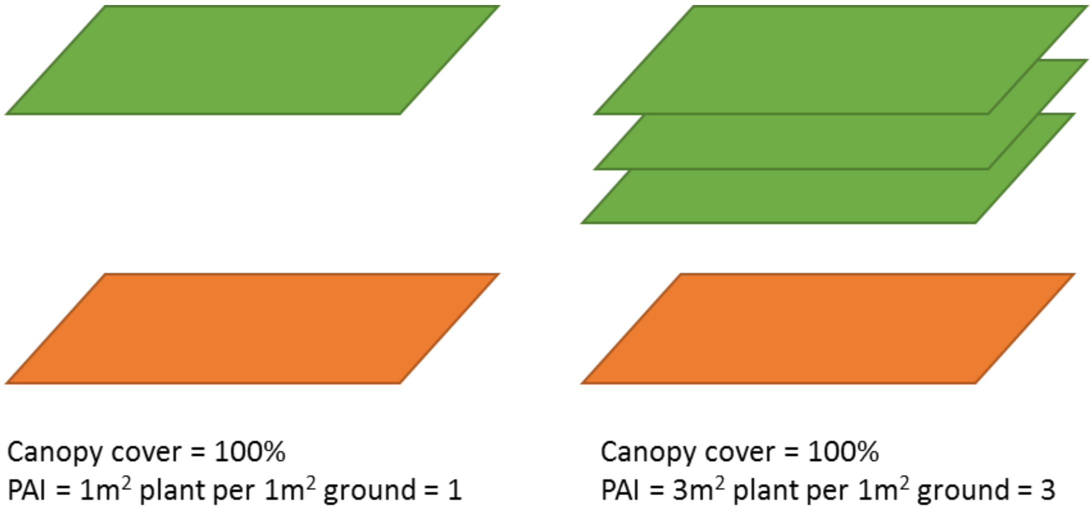


Figure 14. Schematic showing simplified PAI calculations.

In Newell *et al.* (2017) it is stated that the 'PAI is scored on a 0-2 scale', however in the data provided by ARI for 2009, 2010, 2013, 2014, 2015 and 2016 the PAI is between 0 and 2.18. Therefore, it appears that the raw PAI produced by winphot was used. In the winphot manual (ter Steege 1997) it appears that winphot outputs PAI between 0 and (at least) 8. Generally, the PAI is below 2, most likely due to the sparse nature of the chosen native trees. For further evidence, Fire, Flood and Flora was engaged by the North Central Catchment Management Authority to collect data, including PAI, in Gunbower Forest 2017 MDBA's The Living Murray (TLM) program (Bennetts & Jolly 2017) which was fed into the development of the SCT. In this case they do not mention any scaling process. For the Hindcast we used the data provided by ARI which includes PAI with values over 2.

In the past the PAI scores were scaled by forest type (e.g. Cunningham *et al.* 2013b). As the 2017 SCT (*Murray–Darling Vegetation Monitor User's Guide* 2017) is trained on unstandardized PAI values (Newell *et al.* 2017) it would be inappropriate to use standardized PAI values in the tool, due to the internal decision tree architecture of the SCT. The reasoning behind this standardisation was 'to account for the historical reduction in PAI owing to the decline in productivity associated with reduced rainfall, flooding and increased evaporation downstream along the Murray River floodplain (Bioregion) and local differences in water availability within a floodplain (Forest Type)' (Cunningham *et al.* 2013b). However, this standardisation makes assumptions about the maximum PAI a region or forest can reach, and also may not reflect the reality of what is actually on the ground and being detected by the satellite. For example, we may expect river red gum forests to be denser than black box and for the satellite to detect that, if we standardise PAI such that they are then equivalent PAIs this could result in unintended consequences. Another reason to standardise quoted is to account for 'water availability' and standardising for that, but of we might naturally expect trees in areas with low water availability to have lower condition or PAI than trees with higher water availability and so whether this should be standardised needs to be carefully considered. While it may be beneficial to get the model retrained to better account for species or bioregional differences, this should form just a part of a larger conversation about improvements and ecological reasoning to inform a future SCT.

4.1.2.2 Crown Extent

Crown extent (CE) is defined as 'the percentage of the assessable crown in which there are live leaves. This includes branches that have leaves at their base and middle but not at their tips, including epicormic growth' (Cunningham *et al.* 2018). The field protocol also states that 'as recommended by the Newell *et al.* (2017) Crown extent is assessed and recorded to the nearest 5% of the assessable crown (previous 20% categories are now considered too coarse for accurate modelling)'. However, in the data provided to MDBA by GA/ARI for 2009, 2010, 2013, 2014, 2015 and 2016 the crown extent appears to be the result of scaling raw crown extent. For example, 65.2% equals 3.26 units. Therefore, our understanding is that the current SCT was trained with raw CE scores. In the future this needs to be clarified.

Other aspects of the field protocol for collecting crown extent could pose an issue. For example, to record crown extent for a plot, the plot needs to contain 30 trees. However, since the model is not restricted to modelling condition only for areas with more than 30 trees, areas with low crown extent due to low numbers of trees are unlikely to be modelled correctly. As with issues relating to PAI this needs to be considered in improvements and ecological reasoning to inform a future SCT.

4.1.2.3 Live Basal Area

Live basal area (LBA) is relatively well understood compared to PAI and CE, conceptually. However, it is unclear the extent to which satellite imagery can differentiate between live and dead basal area. Future work should focus on investigating this, as this may add act as a source of unquantified error.

4.1.2.3 Stand Condition Score

The stand condition score (SCS) used to train the 2017 SCT (*Murray–Darling Vegetation Monitor User's Guide* 2017) is defined in Newell *et al* (2017) as:

$$\text{Stand Condition Score (SCS)} = \frac{(2\text{CE}+5\text{PAI}+0.1\text{LBA})}{3} \qquad \text{Eq 2}$$

Where:

CE = Crown Extent, scaled from 0 to 5 therefore multiplied by 2.
PAI = Plant Area Index, scaled from 0 to 2 therefore multiplied by 5.
LBA = Live Basal Area, scaled between 0-10 it is multiplied by 1.

Depending on whether or not the field data has been scaled as defined above before use in the SCS equation the equation may need to be altered. For example, when LBA is scaled between 0-10 it is multiplied by 1 but when it is measured as a percentage it must by multiplied by 0.1. However, this alteration would result in a mathematically identical outcome. Simply, the three field data variables must be on the same scale before being averaged to generate the stand condition score.

Table 11 Stand Condition Score Categories

Stand Condition Score Range	Condition Category
>8 - 10	Good
>6 – 8	Moderate
>4 – 6	Poor
>2 – 4	Degraded
0 -2	Severely Degraded

Another issue surrounding the SCS is the assignment of meaning to the bins, currently done as per Table 11. In this case, we consider SCS as absolute so we can determine whether condition is being maintained in good or poor condition etc, and we can compare forests from different locations, e.g. forest 1 is in good condition compared to forest 2. However, some forests are likely to be naturally lower in some of the field variables, due to the nature and location of different tree species. If we considered SCS to be only relative then this issue would not arise, however it would only be possible then to determine whether condition had improved, declined or remained the same for a given area and compare the change in condition rather than the absolute condition between forests. If we are to keep condition as an absolute variable, then some thought may need to be given about having different standards for different species, and potentially areas (similar to the issue of standardising by PAI maxima). Given the confusion surrounding the field data and stand condition score, a reassessment of these is necessary to move forward.

4.1.2.4 Benchmarking

Another issue is that the model classifies condition across all regions using the same criteria. This method does not consider regional differences that could influence condition. For example, river red gum that occurs in arid regions within the Basin is not expected to have high PAI, CE or LBA (resulting in a low condition scores) because of naturally less favorable conditions than river red gums further upstream.

In other words, the method does not control for contextual factors that affect condition, which are independent of environmental water delivery. This may lead to erroneous interpretations, for example river red gums in an arid region reported as being ‘poor’ condition, when even without water management intervention they would never have the same condition as river red gums elsewhere. Consequently, the reporting state may suffer legislative repercussions based on unrealistic expectations of change.

For investigating condition of a given species, there are several ways this issue could be dealt with:

1. Benchmarking:
    a. Spatial benchmarking:
        i. Based on the understanding of condition of a given species at different locations.
        ii. This allows absolute condition of a given species to be compared across the Basin. For example, river red gum condition is good in Region A but poor in Region B. In the above example, if the maximum condition a river red gum could reach was 7, and the tree received an SCT of 7, then this would be an SCT of 10 (7/7 = 100%, scaled to 0-10).
        iii. Will have to incorporate assumptions of condition of trees across the Basin. For example, river red gum in arid regions may have had higher condition scores before water management over a hundred years ago but we are currently unable to tell.
    b. Temporal benchmarking:
        i. Based on the condition of tree stands at different times in history.
        ii. The Hindcast provides condition back to 1987. This allows the condition at most points in time (at an annual level) to be determined. For example, one may wish to consider condition against the average of previous years, or since the millennium drought, or inception of the Basin Plan.
        iii. This method can allow for relative change since the chosen time to be investigated. Choosing the earliest or highest value as a proxy for maximum achievable condition (the goal of spatial benchmarking) may not actually be the highest possible (setting the bar too low), as river regulation has been occurring decades earlier than 1987.
2. Use relative change:
    a. This does not require assumptions about maximum condition to be made, and as such is less subjective and thus may offer more confidence with interpretation.
    b. Can compare relative change across the basin. For example, river red gum condition is declining in Region A but increasing in Region B.

There is certainly a need to determine which of these options is necessary to take in order to report on legislative requirements. Where reporting is based on 'improving' or 'maintaining' relative change or temporal benchmarking (e.g. improved since time A) may suffice. However, where it does not significant investment into spatial benchmarking, or alternatives to the SCT, will be required. For example, NSW Long Term Watering Plans (LTWPs) have ecological objectives stating 'maintain the extent & improve the condition of river red gum communities' which could be answered using temporal benchmarking. However, each ecological objective is broken down to targets which use language such as 'over a 5-year rolling period, maintain the proportion of river red gum communities closely fringing river channels (within 50 m) that are in moderate or good condition' where condition is defined as in Cunningham *et al.* (2009). This requires the values of condition to be correlated with meaning (e.g. 'good' condition), which may require spatial benchmarking if the classes defined by Cunningham *et al.* (2009) are not suitable alone. Separate to this is the issue of comparing different species, either by PAI or condition.

4.1.2 Constant data

4.1.2.1 Vegetation masks

The current species masks within the SCT are in the form of probabilities (see Methods – Data 2.1). This allows the user to have flexibility over whether they would prefer tending towards errors of omission or commission. However, it introduces the issue that different agencies, or indeed people within the same agency, report on different areas resulting in a confused public outlook. For this report in order to avoid setting arbitrary thresholds we used the species map from Cunningham *et al.* (2013c). Due to this uncertainty, we consider that developing a standard tree species map across the basin is of the upmost importance. Newell *et al.* (2017) states that the current masks were based heavily on Victorian data, and so uncertainty in the accuracy of this mask varies regionally and that 'improving these 'masks' or limits to the current model are perhaps the most direct and immediate approach that the MDBA could take to minimising the risks of model commission.'

Overall, there are two main issues with the current species map options:

1. Accuracy
2. Dynamics

Since the current map is in probability of a species occurring, different species overlap, and it is not obvious which species is at any given point. There are also cases of species with a high probability occurring in places such as farm dams or irrigation plots. Aggregated inaccuracies could have a significant impact on how the condition of tree species in a certain area is reported.

Secondly, because changes in the landscape occur there needs to be a dynamic map. Due to the challenges in achieving this, we would consider the creation of annual maps to be a significant success. This would allow reporting on extent as well as condition which is a requirement of the Basin Plan (2012). This would also help avoid issues such as reporting on condition for an area that used to be one species but has changed, perhaps to another species but more likely changing land use, e.g. deforested for grazing.

Overall accurate annual species maps would greatly benefit this project and also have broader uses across federal and state governments.

4.1.3 Remotely sensed data

New satellites have become available since the inception of the SCT, such as Sentinel 1 and 2, and future planned satellites will soon be available e.g. NISAR (National Aeronautics and Space Administration 2019). Newell *et al.* (2017) saw the potential of new satellite imagery, with higher resolution, revisit rates and radar which can provide information on vegetation structure to have significant potential to support various environmental monitoring. In particular reference to the SCT, we see that new satellite imagery could be used in two ways they could also be incorporated into future SCT algorithms be and used to create accurate and dynamic vegetation masks.

4.1.3.1 Epoch

The reasons for having a 16-month epoch are not clearly documented. An internal MDBA email states that "each annual mosaic with the exception of 2013 to be a composite of 16 months of Time

Series (TS) images starting from 1 September of the previous year and the 31 December of the desired year. For example, time frame for 2009 mosaic would be from 1/09/2008 to 31/12/2009. As Landsat 8 image acquisition began around June 2013, the time frame for 2013 mosaic would be of 12 months starting from June 2013 to June 2014”. The composites provided by GA, through ARI, had file names which also confirmed this, e.g. 20080901_20100101 (in coding the last date is usually not inclusive and therefore is the same as saying to 31st of December 2009). From these two lines of evidence, we then also drew the conclusion that any field data would relate to the centre year, e.g. in “20080901_20100101” the centre year would be 2009. However, it is also worth noting that Graeme Newell (ARI) has suggested that the SCT is fairly broad and simple, therefore any image could be applied, and theoretically give reasonable results.

4.1.4 SCT software

4.1.4.1 Usability

The current version of SCT software tool has a straightforward user interface, and some simple automation capabilities, however there are some elements of the implementation that create barriers to its use in a corporate environment (i.e. government) and as part of a data workflow.

- The current tool uses a largely unsupported version of Microsoft Office database tools (x64) making it incompatible with most corporate standard operating environments.
- Input raster files for processing and the output results are in Erdas RE Mapper image format, with a very specific format description text file. This format is largely unsupported by current major GIS processing tools (e.g. ESRI) and make it impossible to integrate the tool into an automated workflow using our current image processing systems (for example EE or the GA DataCube).
- It is not possible to batch run extra processes (such as validation).

4.1.4.2 Tool suitability and flexibility

Historically the current SCT tool has been a capable platform for processing and requires very little system resources to run. The simplicity of the tool architecture comes at a price in terms of flexibility and adaptability to more powerful IT processing environments.

- Currently the implementation of the decision tree algorithm is fixed, and in a ‘black box’ design. There is no ability to alter and optimise parameters for different types of input data (temporal/seasonal and area (i.e. different parameters for regions). Even the spatial extent is fixed regardless of the input data extents.
- Because the current algorithm is embedded in the design of the tool there is no ability to run these algorithms on much more capable platforms available (such as EE). This limits the capabilities of the tool to the resources of simple desktop systems, now with higher resolution imagery and more sophisticated modelling requirements there is no possibility that the desktop environment will provide enough processing capability to run these models.

4.1.4.3 Error checking

The SCT appears to have no in-built error checking mechanism nor does it specify when an operation has successfully completed. We encountered two issues with this:

1. For the epoch 20050901-20070101 the model appeared to have not completed running so there was a strip of missing data alone the bottom of the raster.
2. For the epoch 20170901-20190101 the model appeared to have completed running, but on closer inspection there were unaccounted for lines in the condition image (Figure 15). This would have been missed without detailed manual inspection, which is generally not feasible for dealing with large quantities of data.

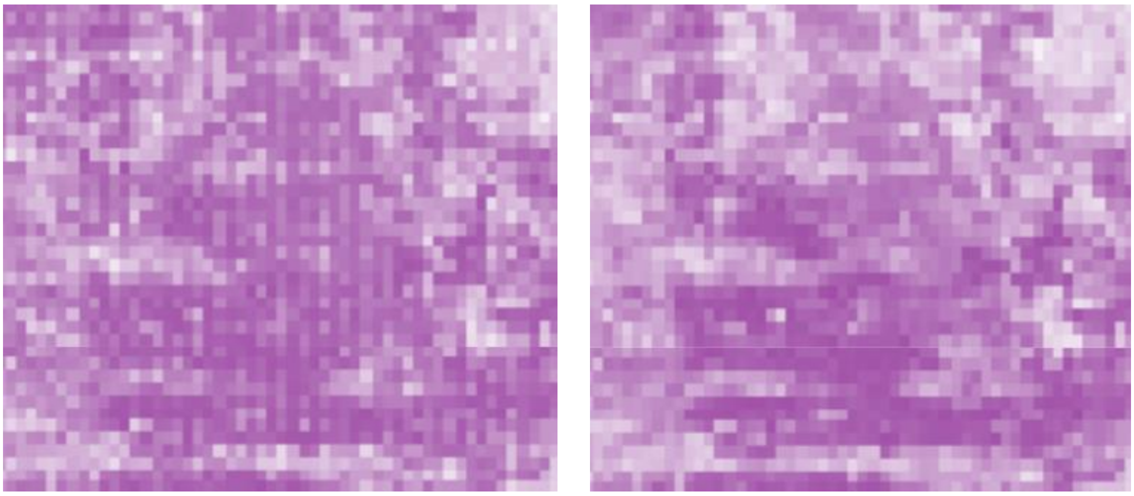


Figure 15. Images showing an error causing artifical stripiness in the raster (left) and the correctly processed raster (right).

4.1.4.4 Model output coverage

To complete the hind-cast we used the GA tile schema, interestingly though the model always outputs results for a set area (Figure 16). Some areas had data added in (condition present but not covered by GA tile schema), while other areas had all data removed (covered by GA tile schema but no condition results). The GA tile schema did not cover the entire basin (Figure 17) and in the future it would be preferable to do so.

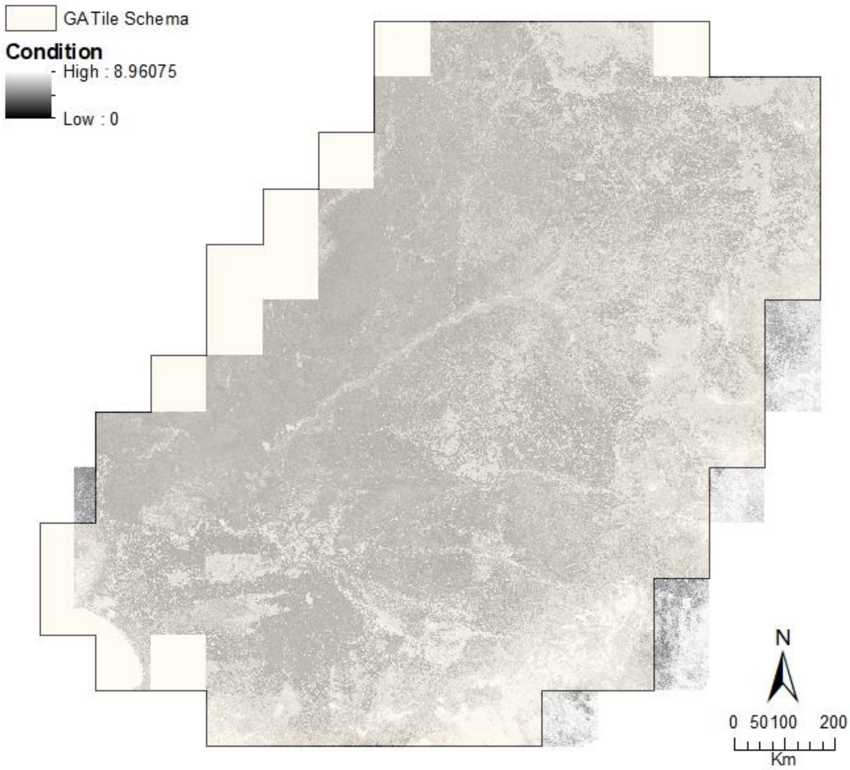


Figure 16. The GA tile schema and the model output.

4.1.4.5 Post-processing

Post-processing included in the SCT to help deal with extreme condition values, without altering the average values. It would be preferable to incorporate better spread condition field data to develop a new underlying algorithm that can cope with modelling all condition circumstance and thus avoid the need to post-process altogether.

4.1.4.6 Improvements

The sole recommendation that will solve all the issues outlined above is to move the current SCT processing algorithm to an open platform such as Python or EE JavaScript.
Moving the implementation of the tool to either of these platforms will enable:

- Easy integration into currently available high-performance computing environments (e.g. EE and GA Cube) with no issues in terms of system compatibility (using Jupyter Notebooks or JavaScript modules).
- Flexibility to modify the parameters and process flow easily.
- Scalability to use higher resolution imagery and support more complex models.

4.1.5 Models

The SCT has three in-built models, all of which are bagged decision trees. In this project we considered only ‘Veg Monitor Map All Years’. For the five years with field data the average R^2 was 0.574 which is less than ideal. In particular, for the 2016 centre-year model R^2 is only 0.454. This could be due to the large amount of rainfall that was experienced that year, causing an extreme in condition which was not able to be adequately modelled, most likely due to a lack of field data collected and used to train the 'Veg Monitor Map All Years' algorithm under similar conditions. Rectifying this would require collecting more field data and retraining the underlying algorithm. These results also cast doubt about the other years modelled but without any field data the model fit cannot be calculated.

This suggests that the 'Veg Monitor Map All Years' model needs to be re-developed to better account for variability in condition. As part of doing this, it could be beneficial to consider incorporating new field data, satellite imagery, the epochs used, and explore new modelling techniques. For example, Sentinel 2 and Sentinel 1 have a higher resolution than Landsat and could be used to create more precise condition maps. Veg Monitor Map All Years' appears to require 16-month epochs (see Epoch 2.1.3.1), however a model that could be used for single dates or shorter time period could be beneficial. Although there is a 'Quick Veg Map All Years' and 'Veg 2016 and 2017 Only' model in the SCT which are supposed to accept one image, 'Quick Veg Map All Years' is insinuated to perform poorer than 'Veg Monitor Map All Years' and 'Veg 2016 and 2017 Only' should only be used for those years (*Murray–Darling Vegetation Monitor User's Guide* 2017). Newell *et al.* (2017) also highlighted the need to consider new machine 'deep learning' methods. They also suggest investigating methods which can incorporate contextual information (e.g. adjacency to other forest, woodland, agriculture, surface water, preceding climatic conditions, etc.), as the surrounding environment often influences the condition of nearby forests.

In its current form the SCT algorithms cannot be easily viewed or understood. One way around this would be to rewrite the tool as a python script, which would allow the user complete transparency and oversight of the process, allowing informed decisions about what would be reasonable adjustments or inputs to use. There are many different methods of machine learning and new methods of modelling are constantly being developed, e.g. TensorFlow, and any re-development of the SCT algorithm should be used as an opportunity to explore these to achieve the best possible outcome.

4.2 Development & Application

The SCT tool uses a mix of field data and remote sensing imagery to make inferences about the condition of woody vegetation. The outputs generated by the tool can be used to assess changes in vegetation condition over time, and to help water planners and managers to make decisions about environmental watering to improve outcomes. The reliability and usefulness of the tool is dependent on the integrity of the information that used to train the model, and consistent field data collection procedures and handling across multiple State and Federal government agencies. Secondly, successful application of the tool for purposes such as condition monitoring requires a unique set of expert skills and coordination across those agencies.

4.2.1 Information Integrity

A critical consideration when applying the SCT is the integrity of the information it generates. The SCT can be used to infer tree condition at a point in time and space however a more robust application of the tool is to utilize its outputs to elucidate trends in tree stand condition through time and space as illustrated in Figure 18.

Spatial Scale

Temporal scale

	Site	Region	Basin Wide
Season	Low	Low	Medium
Year	Low	Medium	High
Multi-year	Medium	High	Very High

Figure 17 Relative confidence in Stand Condition Tool outputs at various scales

There are two key questions when evaluating the condition of riverine woody vegetation in the MDB:

1. Has condition of woody vegetation been maintained or improved?
2. Has recruitment of woody vegetation been maintained or improved?

The SCT can inform whether the condition of the target species have improved or declined, providing that it is modelled well to the field data. However, it is unable to inform questions around recruitment. Regardless, the information generated by the SCT has the highest utility when combined with other sources in a multiple lines of evidence approach. When considered in concert with site-based assessments of recruitment and ecosystem condition the SCT tool becomes powerful, the only truly basin-wide line of evidence.

There are several knowledge gaps in the process that limit the utility of the SCT's outputs including:

- No standardised Basin-wide classification of woody vegetation ecosystems;
- Basin-wide lack of target species recruitment stage demographic data to enable reporting; and
- A lack of an accurate evidence-based ecological definition of woody vegetation reference/target condition at multiple scales.

Some of these gaps are being address through the JVSC 'improving evaluation and report on woody vegetation condition outcomes, at region and Basin scales' project described in detail in Appendix B. The proposed activities will improve the knowledge base on woody vegetation population demographics and will significantly improve the confidence and ability to evaluate and report woody vegetation condition outcomes at multiple scales; using an improved evidence-base along with standardised and consistent understanding and methodologies.

4.2.2 Improvements in application

In addition to the technical limitations discussed above, the application of the tool is limited by expert knowledge, data accessibility and coordination. Because the tool operates at the intersection of ecology (vegetation condition), data science (remote sensing/machine learning) and public policy (the Basin Plan), there is a limited number of individuals with an in depth understanding of both how the tool operates and the context of its application. This and the interjurisdictional nature of the tool have led to difficulty in having properly informed strategic discussions about the development and operation of the tool.

There is currently no officer level interjurisdictional SCT working group. As such there is no forum through which staff applying the SCT in different jurisdictions can easily communicate. The 'case study' element of the JVSC project has highlighted the need for such a group, while the VTAG provided a useful forum the expertise of the group is heavily biased toward vegetation ecology. The purpose of a SCT working group will be two-fold, firstly to collaboratively operationalize the tool, secondly to inform strategic discussions around improvements and resource allocation.

The suggested participating organizations are listed below:

- Department of Environment, Land, Water and Planning (Victoria),
- South Australian Department for Environment and Water (South Australia),
- Department of Industry - Water Branch (New South Wales)
- Office for Environment and Heritage (New South Wales)
- Department of Environment and Science (Queensland)
- Murray–Darling Basin Authority
- Geoscience Australia

Ideally this group would utilize collaborative software design platform such as GitHub or Jupyter Notebooks to test, operate and refine the tool. Such a group would enable improvements to the SCT to be undertaken through a collaborative and continual process leveraging existing expertise across the member organisations to foster greater collective understanding of all elements of the tool. This will also mean SCT improvements would not be dependent on the necessary funding and expertise being available concurrently as has been the case in the past.

4.2.3 Prioritisation

The numerous tasks for improving the SCT need to be prioritised and based on the work described in this report an assessment detailed below in Table 11 has been made. The assessment is preliminary only and represents the internal view of the MDBA. Any work program to refine the tool will require input from all relevant agencies.

Table 12 SCT Refinment tasks

	Priority	Task	Justification	**Dependencies (on other tasks)**	Limitations
1	Moderate	Collect field data	Field data for a given time can only be collected at that time, whereas the models themselves can be created to model condition in the past at any time. Therefore, field data should be regularly collected.	Ideally Task 5 will have been completed to determine the necessary field work. However, given the time sensitivity it may be necessary to collect something	Additional funding

				rather than nothing.	
2	High	Data handling	MDBA needs to improve internal processes of data collection and collation.	None	Included in business as usual
3	High	Develop a dynamic species map Consider new and upcoming satellite imagery	This would have multiple uses not just for this project, but across government. e.g. Sentinel, NISAR	None	Additional funding
4	Moderate	Determine/Define PAI in relation to condition score	In the condition equation it states that PAI is scaled between 0 and 2, however it is unclear whether it has been or if simply the raw output of winphot was used.	This task may need to be completed in conjunction with Task 5.	Additional funding
5	Moderate	Determine suitability of current SCS		To determine suitability, Task 4 may need to be completed.	Additional funding
6	Low	Determine local PAI maxima	If the review of the SCS (Task 5) finds that PAI needs to be standardised by maxima, then this will become necessary. Field data would inform this, possibly this could be incorporated as part of Task 1.	None	Additional funding
7	Low	Develop a new underlying algorithm for the SCT Consider new and upcoming satellite imagery	This should not be done without first reassessing the field data variables that feed into the SCS, and indeed the SCS itself. e.g. Sentinel, NISAR	Task 5	Additional funding
9	Low	Spatial and temporal benchmarking	Since the SCT outputs maps of condition across the basin and across years, determining the condition at any point or region can be done by most GIS users.	None	Included in BAU

4. Conclusion

Throughout the course of this project the MDBA developed the capability to run the SCT using satellite images processed in-house using Google Earth Engine. The aims of this hindcast report were to test this new capability to assess the performance of the hindcast method against the 2017 Basin Plan Evaluation method. While it was found that the 2017 Evaluation data consistently performed better than the Hindcast data, the difference was uniformly less than 8% of the R-squared and given the known satellite processing differences between the two projects this was decidedly reasonable. Given the results of the hindcast, the SCT was applied to the entire Landsat imagery archive and these results will be used to inform the 2020 Evaluation.

Despite the technical limitations outlined in Chapter 4.1.4, the SCT is currently the only basin-wide measure of tree stand condition and provides valuable information on ecosystems at the scale required to guide water management in the Murray–Darling Basin. Addressing the technical and accessibility limitations will help to improve the reliability of the evidence base used to inform decisions around the management of water dependent ecosystems across the Basin.

1. References

Asner, GP, Scurlock, JMO & Hicke, JA 2003, 'Global synthesis of leaf area index observations: implications for ecological and remote sensing studies', *Global Ecology & Biogeography*, 12, 191–205.

Department of Environment and Energy 2017, *Australian National Aquatic Ecosystem (ANAE) classification for the Murray Darling Basin – Wetlands*, viewed 22nd November 2019 <http://www.environment.gov.au/fed/catalog/search/resource/details.page?uuid=%7BBF0F8CB3-9903-4D5B-85D8-7940D9B3BDFE%7D>

Bennetts, K & Jolly, K 2017, *Gunbower Forest TLM Stand Condition Assessments 2017*. Unpublished report for the North Central Catchment Management Authority, Fire, Flood and Flora, Cape Woolamai, Victoria.

Cunningham, SC, Mac Nally, R, White, M, Read, J, Baker, PJ, Thomson, J and Griffoen, P 2007, *Mapping the current condition of river red gum (Eucalyptus camandulensis* Dehnh.*) stands along the Victorian Murray river floodplain*, Monash University, Victoria.

Cunningham SC, MacNally R, Griffioen P & White M 2009, *Mapping the condition of River Red Gum and Black Box stands in The Living Murray icon sites, A Milestone Report to the Murray−Darling Basin Authority as part of Contract MD1114*, Murray−Darling Basin Authority, Canberra.

Cunningham, SC, Griffioen, P, White, M & Mac Nally, R 2013a, *Mapping the Condition of River Red Gum (Eucalyptus camaldulensis Dehnh.) and Black Box (Eucalyptus largiflorens F.Muell.) Stands in The Living Murray Icon Sites. Comparison of the predictive power of Landsat and Rapideye imagery, and validation of future predictions based on imagery only,* Murray–Darling Basin Authority, Canberra.

Cunningham, SC, Griffioen, P, White, M and Mac Nally, R 2013b, *Mapping the Condition of River Red Gum (Eucalyptus camaldulensis Dehnh.) and Black Box (Eucalyptus largiflorens F.Muell.) Stands in The Living Murray Icon Sites. Stand Condition Report 2012,* Murray–Darling Basin Authority, Canberra.

Cunningham, SC, White, M, Griffioen, P, Newell, G and Mac Nally, R 2013c, *Mapping Floodplain Vegetation Types across the Murray–Darling Basin Using Remote Sensing*, Murray–Darling Basin Authority, Canberra.

Cunningham, SC & Griffioen, P (2013) Murray Stand Condition Tool User's Guide. Murray–Darling Basin Authority, Canberra.

Cunningham SC, Griffioen P, White M and Mac Nally R, (2014) A Tool for Mapping Stand Condition across the Floodplain Forests of The Living Murray Icon Sites. Murray–Darling Basin Authority, Canberra.

Cunningham, White, Bowen, Dillewaard, Butler, Ryan & Driver, 2018, *Field protocol for assessing the 'stand condition' of floodplain forests and woodlands in the Murray–Darling Basin,* Murray–Darling Basin Authority, Canberra.

Department for Environment and Water 2019, *Chowilla Floodplain*, viewed 22nd November 2019, <https://www.environment.sa.gov.au/topics/river-murray/improving-river-health/the-living-murray-program/chowilla-floodplain>

Geoscience Australia 2018, *Surface Reflectance*, Digital Earth Australia, viewed 4th November 2019, <https://docs.dea.ga.gov.au/data/data.html>

Google Earth Engine 2019a, *FAQ*, Google Earth Engine, viewed 6th November 2019, <https://earthengine.google.com/faq/>

Google Earth Engine 2019b, *Managing Assets,* Google Earth Engine, viewed 6th November 2019, <https://developers.google.com/earth-engine/image_upload>

Murray Darling Basin Authority 2014,*Basin-Wide Environmental Watering Strategy* https://www.mdba.gov.au/sites/default/files/pubs/qld-mdba-basin-wide-environmental-watering-strategy-2014_3.pdf

Murray–Darling Basin Authority 2017a, *Managed floodplain with current constraints*, data.gov.au, viewed on 25th November 2019, <https://data.gov.au/dataset/ds-dga-9f97aebc-d89e-48ae-8586-67120a8cfcfa/distribution/dist-dga-c1d8bfaf-1aa6-4d05-b5eb-5c1de0032674/details?q=>

Murray–Darling Basin Authority 2017b, *Murray–Darling Basin managed floodplain,* data.gov.au, viewed on 27th November 2019, <https://data.gov.au/dataset/ds-dga-9f97aebc-d89e-48ae-8586-67120a8cfcfa/details?q=floodplain>

Murray–Darling Basin Authority 2018, *Canberra, Basin-Wide Environmental Watering Strategy Regions for Vegetation Outcomes*, data.gov.au, viewed on 31 October 2019, <https://data.gov.au/dataset/ds-dga-d226be5f-cae1-4e67-9827-03e5b7097216/details?q=mdba%20regions>

Murray–Darling Vegetation Monitor User's Guide 2017, Arthur Rylah Institute, viewed 4th November.

National Aeronautics and Space Administration 2019, *NASA-ISRO SAR Mission (NISAR)*, viewed 4th November 2019, <https://nisar.jpl.nasa.gov/>

Native vegetation 2017, Murray–Darling Basin Authority, viewed 4th November 2019, <https://www.mdba.gov.au/sites/default/files/pubs/Native-vegetation-report-Feb18.pdf>

Newell, G, White, M & Griffioen, P 2017, *Development of a Stand Condition Monitoring Tool for the Murray–Darling Basin,* Murray–Darling Basin Authority, Canberra.

Overton, IC, Jolly, ID, Slavich, PG, Lewis, MM & Walker, GR 2005, 'Modelling vegetation health from the interaction of saline groundwater and flooding on the Chowilla floodplain, South Australia', *Australian Journal of Botany*, 54, 2, 207-220.

Rouse, JW, Haas, RH, Scheel, JA & Deering, DW 1974, 'Monitoring Vegetation Systems in the Great Plains with ERTS.' Proceedings, 3rd Earth Resource Technology Satellite (ERTS) Symposium, vol. 1, pp. 48-62.

Stand Condition Assessment of Forests and Woodlands of Gunbower Forest – 2013 2013, Murray–Darling Basin Authority.

ter Steege, H 1996, *Winphot, a programme to analyse vegetation indices, light and light quality from hemispherical photographs*, Tropenbos-Guyana Reports 97-3

United States Geological Survey 2019a, *Landsat Surface Reflectance Quality Assessment*, USGS, viewed 4th November 2019, <https://www.usgs.gov/land-resources/nli/Landsat/Landsat-surface-reflectance-quality-assessment?qt-science_support_page_related_con=0#qt-science_support_page_related_con>

United States Geological Survey 2019b, *CFMask Algorithm*, USGS, viewed 4th November 2019, <https://www.usgs.gov/land-resources/nli/Landsat/cfmask-algorithm>

United States Geological Survey 2019c, *Landsat Levels of Processing*, USGS, viewed 4th November 2019, <https://www.usgs.gov/land-resources/nli/Landsat/Landsat-levels-processing>

Appendix A

JVSC Report

The SCT was applied as part of the 2017 Basin Plan Evaluation and reported on condition stands of river red gum (*Eucalyptus camaldulensis*), black box (*Eucalyptus largiflorens*) and coolibah (*Eucalyptus coolabah*) occurring within the managed floodplain (MDBA 2017a) for each of the BWS regions for 6 years, 2009, 2010, 2012, 2014, 2015 and 2016. A collective assessment of these findings by Basin States lead to a Joint Venture Monitoring & Evaluation Project titled 'Improving evaluation and reporting on woody vegetation condition outcomes, at regional and Basin scales'.

The overall aims of this project were to:

- Improve the knowledge base and refine tools to enable consistent and accurate condition assessments of river red gum, black box and coolibah-dominated forests and woodlands across the Murray–Darling Basin; and in turn
- Improve evaluation and reporting on woody vegetation condition outcomes at a range of spatial scales across the Basin (e.g. State priority assets, catchments, States, and Basin-wide)

Additionally, each of the activities of this project have individual aims, including:

1. **SCA Tool familiarisation and case study demonstration**
    - Facilitate knowledge transfer of the current and potential capabilities of the SCA Tool, to improve the confidence, and support the interpretation and application of outputs for State-based vegetation evaluation and reporting; and
    - Identify and test priority State-based case study sites to apply the SCA Tool to improve the understanding and accuracy of outputs, and target on-ground surveys on woody vegetation condition.
2. **Improving knowledge of woody vegetation population demographics**
    - Review the existing (largely disparate) knowledge base of the demographics and population structure of floodplain trees (River Red Gum, Black Box, Coolibah and River Cooba); and
    - Address key State and Basin-wide knowledge gaps through the development of an agreed method and approach for reporting on woody vegetation demographics (as part of woody vegetation condition outcomes).
3. **Stand Condition model refinement**
    - Update the Stand Condition model with data from previous woody vegetation training data collection
    - Apply the Stand Condition model to the historic earth observation data to create a spatially explicit Stand Condition Model time-series. This will enable a long-term dynamic/historic perspective on the condition of the floodplain forests.
    - Using the time-series of temporal models and statistics derived from these data, develop (in consultation) an agreed approach to site-specific (or pixel specific) Stand Condition model benchmarking, especially at case study locations.

- Use appropriate data, e.g. national land-use and land cover data, MDBA floodplain delineations, and State-based vegetation mapping of woody vegetation extent to delimit the (multi-temporal) Stand Condition models
- Workshop to determine/agreed benchmarks and land-use/veg-type to the time-series.
- Apply agreed benchmarks and land-use/veg-type to the time-series.
- Undertake a workshop of model refinement outcomes (with VTAG members, MDBA and other relevant State and Commonwealth staff).

Appendix B

Google Cloud SDK

The Google Cloud Software Development Kit (SDK) is a set of tools for use with Google Cloud Platform (more information available at https://cloud.google.com/sdk/).

In the Google Cloud SDK Shell:

1. Run **gcloud init** to authorise the connection (see https://cloud.google.com/sdk/docs/authorizing).

2. Log in using your Google Cloud storage account email and password.

3. Run **gcloud config** to set proxy settings (see https://cloud.google.com/sdk/docs/proxy-settings).

4. Use gsutil to copy files from Google Cloud to local storage (see https://cloud.google.com/storage/docs/gsutil/commands/help).
    - **gsutil –m cp –r gs://your_cloud_bucket/percentile_images*.tif C:\your_folder**

5. After successfully downloading all images they can be deleted from the cloud.
    - **gsutil –m rm –r gs://your_cloud_bucket/ percentile_images*.tif**

Appendix C

Joint Venture Monitoring and Evaluation -VTAG project

YEAR 1 (2018/19)

1. Familiarisation and demonstration of the utility of the Stand Condition Assessment (SCA) Tool

This one-off task, undertaken by each of the Basin-States, aims to enhance confidence in the Stand Condition Assessment (SCA) Tool, and improve understanding of the current and potential capabilities of the SCA Tool. This activity aims to improve confidence and interpretation of SCA outputs for State priority assets, and its outputs for use in State and Basin-wide vegetation evaluation and reporting.

Key tasks of this activity include:

- State familiarisation with the SCA Tool and its outputs
  - Workshop guided by ARI and MDBA on the technical detail of the SCA Tool, and the post-processing, interpretation and application in the evaluation of woody vegetation outcomes, respectively.
- State selections and investigations of case study sites (with a case study site selected in each State) to test and validate SCA Tool outputs.
  - VTAG workshop to identify overarching principles for the criteria to identify State case study sites (e.g. 2017 evaluation lessons learnt, gap analysis outputs, priority State-based monitoring, evaluation and reporting activities)
  - State investigations of the SCA Tool outputs (i.e. cross-referencing SCA Tool outputs with State site-data and expert knowledge).
  - Development of a State-based case study report to share lessons learnt, use and application of the SCA Tool.
- Where required (given there might already be suitable data), States undertake agreed field investigations to support the outputs of both the gap analysis for model refinement (gap analysis undertaken in JV SC 2017/18 Vegetation Project), case study site validation, and complement population demographic condition assessments.
  - Case study data collection, analysis and tool validation for reporting purposes.
  - Development of State case study reports (including post-processing interpretations at case study sites) to support both model refinement activities, but also inform priority State asset and Basin-wide evaluation and reporting.

2. Improving knowledge base on woody vegetation population demographics

Develop and implement a one-off study to fill key knowledge gaps to assess responses of woody vegetation condition, via population demographics (including recruitment [1]), to environmental watering (also considering the influence of groundwater), and to complement SCA Tool outputs on woody vegetation condition. This one-off retrospective study will provide valuable information on the current condition status to enable evaluation and reporting on key evaluation questions (see Attachment 2), State and Basin-wide environmental outcomes in 2020.

Key tasks of this activity include:

- Development of an agreed, coordinated and standardised approach and sampling method (through a knowledge base scan) to report on woody vegetation condition outcomes across the Basin;

- Inventory of woody vegetation population studies undertaken within the MDB and to map and compare studies;
- Review of species, locations, methods and findings;
- Build on and refine existing conceptual models linking flooding, other environmental variables, and additional stressors to changes in woody vegetation condition or to woody vegetation recruitment;
- Determine knowledge gap(s) and priorities to enable reporting;
- Development of agreed research questions, environmental and hydrological variables;
- Summary of findings report;
- VTAG workshop and review to communicate current knowledge base, identify linkages and knowledge gap(s), determine agreed approach and methods (evaluation and standardisation); and
- Workshop outcomes summary and technical report outlining agreed methods.

This activity will build existing knowledge bases, utilise data collected, value-add and complement numerous existing programs. Co-investment in this project is at State discretion and various programs (i.e. including those listed in related projects and programs below) are likely to benefit from project involvement.

Appendix D

BWS Region species coverage

The area in square kilometers of each of the three tree species, as masked, by BWS Region.

	Area (km^2)		
BWS Region	**River Red Gum**	**Coolibah**	**Black Box**
Barwon-Darling	82.4	271.8	225.6
Border Rivers	122.7	639.7	81.5
Campaspe	22.7	0.1	0.7
Condamine-Balonne	114.4	600.7	129.7
Eastern Mt Lofty Ranges	3.5	0.1	1.2
Goulburn-Broken	363.9	1.5	22.6
Gwydir	124.7	1382.6	215.6
Lachlan	544.4	16.8	885.6
Loddon	56.3	0.2	36.4
Lower Darling	136.9	10.4	1174.9
Macquarie-Castlereagh	865.2	668.5	1797.7
Moonie	22.3	79.8	25.5
Murray	2157.6	16.6	1945.8
Murrumbidgee	1089.9	12.9	922.6
Namoi	125.0	831.2	236.6
Nebine	2.0	155.5	291.1
Ovens	126.9	0.6	0.4
Paroo	28.1	228.6	383.0
Warrego	74.1	1225.9	830.5
Wimmera-Avoca	127.2	1.2	118.8

Appendix D

Example SCT Outputs

Macquarie-Castlereagh

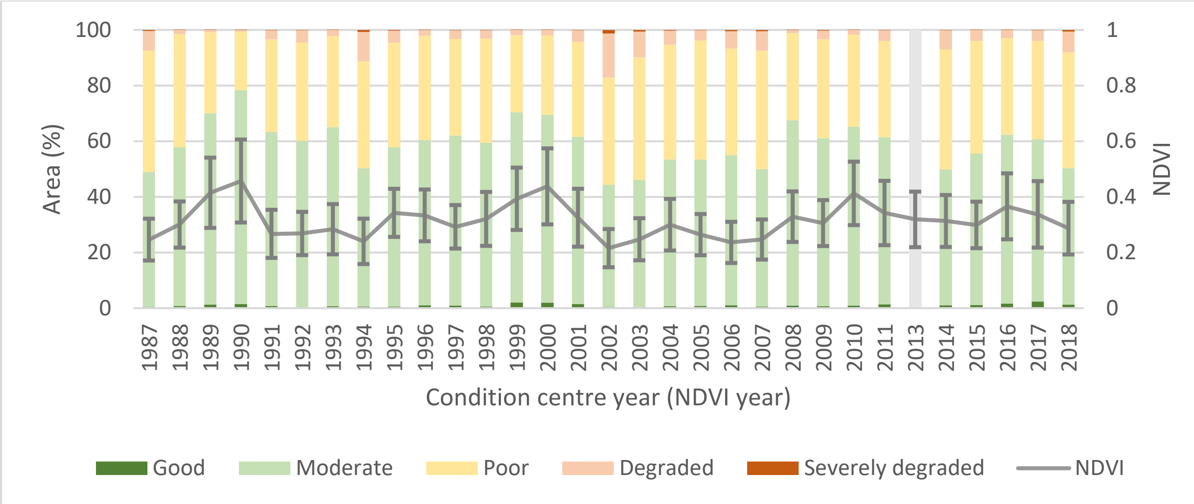


Figure 18: River Red Gum in Macquarie-Castlereagh

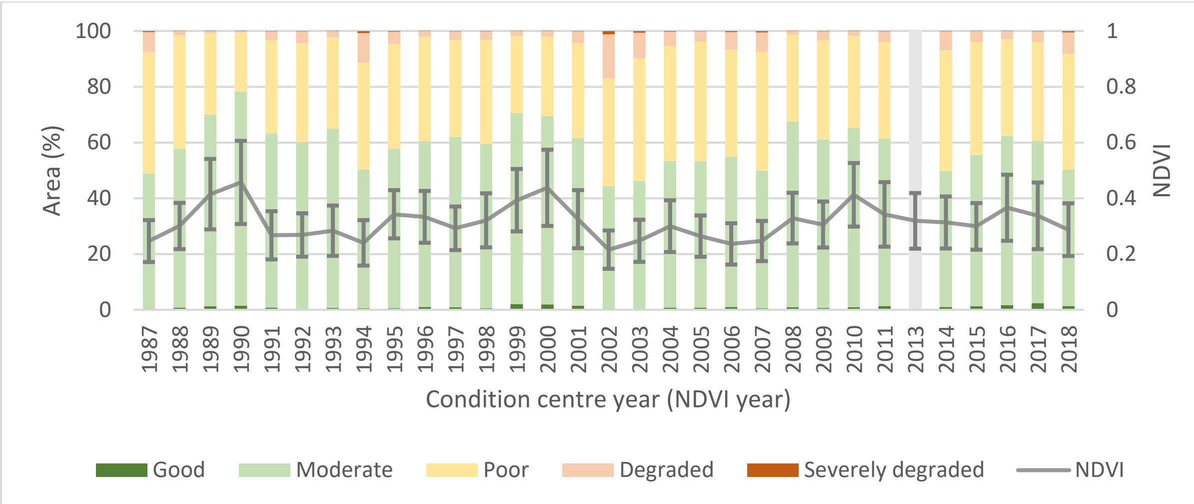


Figure 19: Black Box in Macquarie-Castlereagh

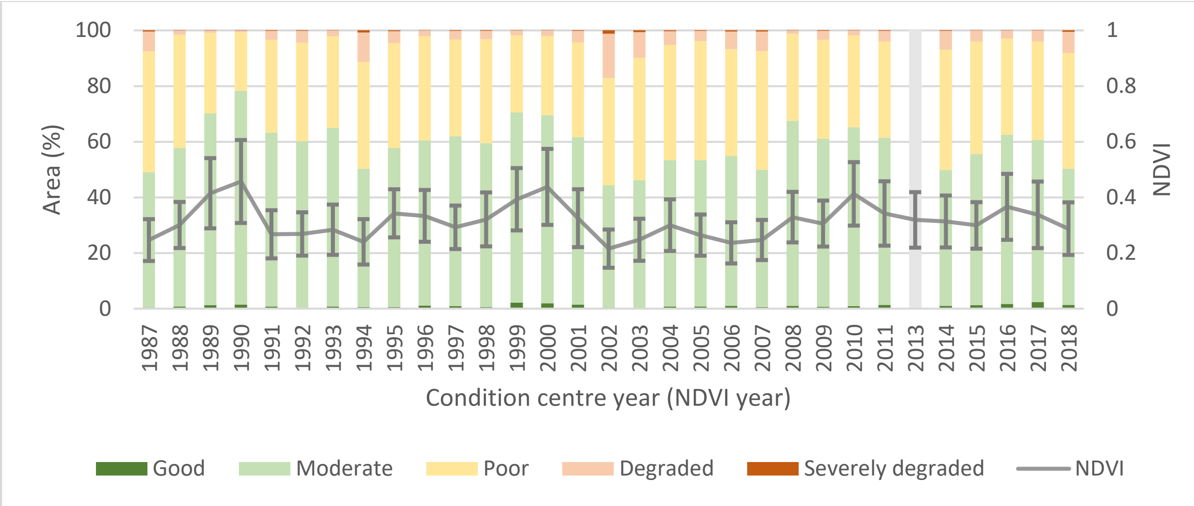


Figure 20: Coolibah in Macquarie-Castlereagh

Barwon-Darling

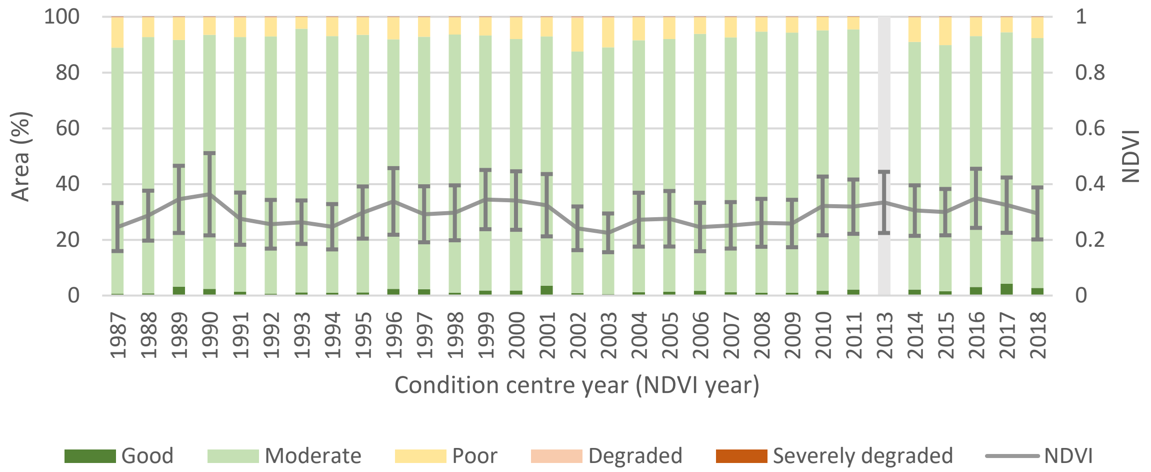


Figure 21: River Red Gum

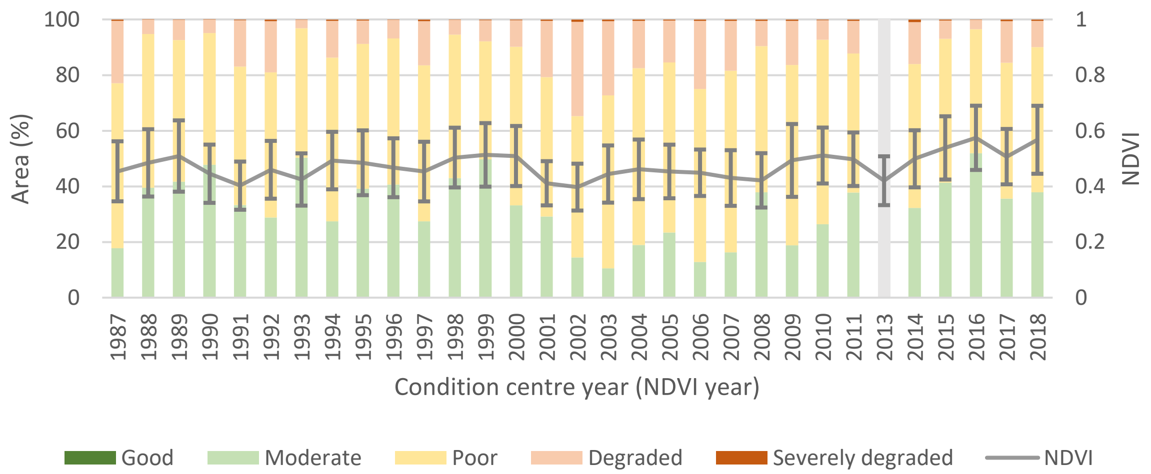


Figure 22: Black Box in the Barwon-Darling

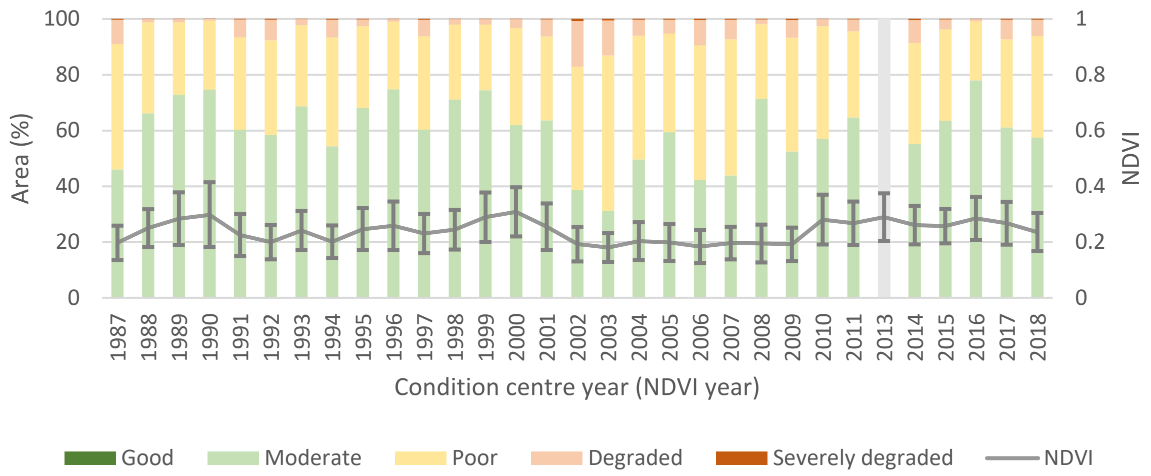


Figure 23: Coolibah in the Barwon-Darling

Appendix E

Script to create SCT inputs

```
var L8sr = ee.ImageCollection('LANDSAT/LC08/C01/T1_SR');

var L7sr = ee.ImageCollection('LANDSAT/LE07/C01/T1_SR');

var L7sr = ee.ImageCollection('LANDSAT/LT05/C01/T1_SR');



//Map.addLayer(table3);

//Map.addLayer(table4);

//var start = ee.String('2017-09-01'); //'2008-09-01'

//var end = ee.String('2019-01-01'); //'2010-01-01'

var bounds = table2.geometry().union(table3.geometry()).dissolve(); // just the BWS regions from
assets

var percentile = 75;          // which percentile to use

var year = 22;

//var description = 'P_' + year + '_50'; // Description of export, please use
'datefrom_dateto_percentile_collection_bands' eg 20170901_20190101_P50_L8

Map.addLayer(bounds);

////////////////////////////////////////////////////////////////////////

Map.centerObject(bounds);


//Mask cloud
```

```
var cloudMaskL457 = function(image) {
  var qa = image.select('pixel_qa');
  // If the cloud bit (5) is set and the cloud confidence (7) is high
  // or the cloud shadow bit is set (3), then it's a bad pixel.
  var cloud = qa.bitwiseAnd(1 << 5)
          .and(qa.bitwiseAnd(1 << 7))
          .or(qa.bitwiseAnd(1 << 3))
  // Remove edge pixels that don't occur in all bands
  var mask2 = image.mask().reduce(ee.Reducer.min());
  return image.updateMask(cloud.not()).updateMask(mask2);
};

// Function to cloud mask from the pixel_qa band of Landsat 8 SR data.
function maskL8sr(image) {
  // Bits 3 and 5 are cloud shadow and cloud, respectively.
  var cloudShadowBitMask = 1 << 3;
  var cloudsBitMask = 1 << 5;

  // Get the pixel QA band.
  var qa = image.select('pixel_qa');

  // Both flags should be set to zero, indicating clear conditions.
  var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
      .and(qa.bitwiseAnd(cloudsBitMask).eq(0));

  // Return the masked image, scaled to TOA reflectance, without the QA bands.
  return image.updateMask(mask)//.divide(10000) // Why is this here?
      .select("B[0-9]*")
      .copyProperties(image, ["system:time_start"]);
```

```
}

// rename bands
var renameBands8 =  function(img){
  return img.select(['B2','B3','B4','B5','B6','B7']).rename(['blue','green','red','nir','swir1','swir2'])
};
var renameBands57 =  function(img){
  return img.select(['B1','B2','B3','B4','B5','B7']).rename(['blue','green','red','nir','swir1','swir2'])
};

//Collections
var L8col = L8sr
        .filterBounds(bounds)
        .map(maskL8sr)
        //.filter(ee.Filter.lessThan('CLOUD_COVER',40))
        .map(renameBands8);
var L7col = L7sr.filterDate('1999-01-01','2003-05-31')
        .filterBounds(bounds)
        .map(cloudMaskL457)
        .map(renameBands57);
var L5col = L5sr.filterDate('1986-09-01','2012-05-05')
        .filterBounds(bounds)
        .map(cloudMaskL457)
        .map(renameBands57);
var col = L8col.merge(L7col).merge(L5col);
print(col.first());

var L5sorted = L5col.sort('system:time_start',false);
print(L5sorted.first());
```

```
// make list of years
//filter for each month, sum, and then sum over the basin
var timeframe = ee.String('year');
var start_list = ee.Date('1986-09-01');
var end_list = ee.Date('2017-09-01');
var months = end_list.difference(start_list, timeframe);
var start_datelist = ee.List.sequence(0, months.int()).map(function(count) {
  return start_list.advance(count, timeframe);
});
//print('start_datelist',start_datelist);

var timeframe = ee.String('year');
var start_list = ee.Date('1988-01-01');
var end_list = ee.Date('2019-01-01');
var months = end_list.difference(start_list, timeframe);
var end_datelist = ee.List.sequence(0, months.int()).map(function(count) {
  return start_list.advance(count, timeframe);
});
print('end_datelist',end_datelist);

//print(ee.String(ee.Date(start_datelist.get(0)).format('YYYY-MM-dd')));

// Make percentile
var p = col.filterDate(ee.String(ee.Date(start_datelist.get(year)).format('YYYY-MM-
dd')),ee.String(ee.Date(end_datelist.get(year)).format('YYYY-MM-dd')))
        .reduce(ee.Reducer.percentile([percentile]));

print('Date range:',ee.Date(start_datelist.get(year)).format('YYYY-MM-
dd'),'to',ee.Date(end_datelist.get(year)).format('YYYY-MM-dd'));

// Add amp to layer
```

```
Map.addLayer(p.clip(bounds),{bands:['red_p'+percentile+'','green_p'+percentile+'','blue_p'+percentil
e+''],min:0,max:4000});


//Cumulative iteration

var algorithm = function(current,previous){

  previous = ee.List(previous);

  var n1 = ee.Number(previous.get(-1));

  //var n2 = ee.Number(previous.get(-2));

  return previous.add(n1.add(1));

};

//Compute 10 iterations

var numIteration = ee.List.repeat(1,30);

print('numIteration:',numIteration);

var go = [0,1];

var sequence = numIteration.iterate(algorithm,go);

print(sequence);


// For loop https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for

sequence.evaluate(function(ids){

  for (var i=0; i<ids.length; i++) {

    print(i);

    var image = col.filterDate(ee.String(ee.Date(start_datelist.get(i)).format('YYYY-MM-
dd')),ee.String(ee.Date(end_datelist.get(i)).format('YYYY-MM-dd')))

    .reduce(ee.Reducer.percentile([percentile]))

    .clip(bounds);

            Export.image.toCloudStorage({

            image: image,

            description: 'MDB_P'+ percentile + '_' + i + '_',

            bucket: 'mdba_gee_bucket',
```

```
        region: bounds,
        scale: 30,
        maxPixels: 100000000000
});
  }
});
```

Script to calculate area of condition classes

```
// 'anae' is the selected ANAE_Wetlands_2017 polygons in assets
// 'nsw' is the selected Interim_NSW polygons in assets
// 'mfpcol' is the selected floodplain layers as a collection under assets
// 'condcol' is the SCT condition outputs as a collection under assets

// Make floodplain mask
// Make ANAE forest mask
var forests = anae.eq(22)
      .add(anae.eq(30))
      .add(anae.eq(31))
      .add(anae.eq(35))
      .add(anae.eq(66))
      .gt(0)
      .selfMask();

// Make floodplain mask
var mfp = mfpcol.mosaic().gt(0);
print(mfp);
Map.addLayer(mfp,{min:0,max:1,palette:['blue']},'MFP');
```

```
// Make interim NSW forest mask
var nsw_forests = nsw .eq(1)
                  .add(nsw.eq(2))
                  .add(nsw.eq(4))
                  .add(nsw.eq(7))
                  .add(nsw.eq(10))
                  .add(nsw.eq(11))
                  .add(nsw.eq(13))
                  .add(nsw.eq(15))
                  .add(nsw.eq(16))
                  .gt(0)
                  .unmask();

// Add all masks together
var mfp_forest_add = forests.unmask()
                      .add(mfp.add(1).unmask())
                      .add(nsw_forests.unmask())
                      .selfMask()
                      .multiply(ee.Image(0));

// Make binary area image
var sppmfp = spp;
var rc = sppmfp.mask(spp.eq(3)).divide(ee.Image(3)).add(mfp_forest_add);
var rr = sppmfp.mask(spp.eq(1)).add(mfp_forest_add);
var bb = sppmfp.mask(spp.eq(2)).divide(ee.Image(2)).add(mfp_forest_add);
Map.addLayer(rr.randomVisualizer(),{},'rr species map');

// Make group image
var binCondition = function(cond){
```

```
  var bin1 =
cond.mask(cond.mask(cond.lte(20)).gt(0)).multiply(ee.Image(0)).add(ee.Image(102)).unmask();
//was gte

  var bin2 =
cond.mask(cond.mask(cond.gt(20)).mask(cond.lte(40)).gt(0)).multiply(ee.Image(0)).add(ee.Image(24
)).unmask();

  var bin3 =
cond.mask(cond.mask(cond.gt(40)).mask(cond.lte(60)).gt(0)).multiply(ee.Image(0)).add(ee.Image(46
)).unmask();

  var bin4 =
cond.mask(cond.mask(cond.gt(60)).mask(cond.lte(80)).gt(0)).multiply(ee.Image(0)).add(ee.Image(68
)).unmask();

  var bin5 =
cond.mask(cond.mask(cond.gt(80)).mask(cond.lte(100)).gt(0)).multiply(ee.Image(0)).add(ee.Image(8
10)).unmask();

  var groupImage = bin1.add(bin2).add(bin3).add(bin4).add(bin5);//should then keep in zeros
.mask(cond);

  return groupImage;

};


var groupImages = condcol.map(binCondition);

print('groupImages',groupImages)

Map.addLayer(groupImages.first().randomVisualizer(),{},'groupImage')


// Function to add species as a property

var addRRG = function(element){

  var dict = {species: 'River Red Gum'};

  var nowhereFeature = ee.Feature(null, dict);

  return element.copyProperties(nowhereFeature,['species']);

};

var addRC = function(element){

  var dict = {species: 'Coolabah'};

  var nowhereFeature = ee.Feature(null, dict);

  return element.copyProperties(nowhereFeature,['species']);

};
```

```
var addBB = function(element){
  var dict = {species: 'Black Box'};
  var nowhereFeature = ee.Feature(null, dict);
  return element.copyProperties(nowhereFeature,['species']);
};



// Calculate area by group
// River red gum
var calcAreaRR = function(groupImage){
  return rr.multiply(ee.Image.pixelArea()).addBands(groupImage).reduceRegions({
    reducer:ee.Reducer.sum().group({groupField:1}),
    collection: table,scale:rr.projection().nominalScale()})
                                 .map(addRRG)
                                 .map(function(element){
                                    var dict = {si: ee.String(groupImage.get('system:index'))};
                                    var nowhereFeature = ee.Feature(null, dict);
                                    return element.copyProperties(nowhereFeature,['si']);
});
};
var rr_area = groupImages.map(calcAreaRR).flatten();
print('rr_area',rr_area);
//print('test', ee.String(groupImages.first().get('system:index')))

// River coolabah
var calcAreaRC = function(groupImage){
  return rc.multiply(ee.Image.pixelArea()).addBands(groupImage).reduceRegions({
    reducer:ee.Reducer.sum().group({groupField:1}),
    collection: table,scale:rc.projection().nominalScale()})
                                 .map(addRC)
```

```
                                   .map(function(element){
                                       var dict = {si: ee.String(groupImage.get('system:index'))};
                                       var nowhereFeature = ee.Feature(null, dict);
                                       return element.copyProperties(nowhereFeature,['si']);
});
};
var rc_area = groupImages.map(calcAreaRC).flatten();
print('rc_area',rc_area);

// Black box
var calcAreaBB = function(groupImage){
  return bb.multiply(ee.Image.pixelArea()).addBands(groupImage).reduceRegions({
    reducer:ee.Reducer.sum().group({groupField:1}),
    collection: table,scale:bb.projection().nominalScale()})
                                   .map(addBB)
                                   .map(function(element){
                                     var dict = {si: ee.String(groupImage.get('system:index'))};
                                     var nowhereFeature = ee.Feature(null, dict);
                                     return element.copyProperties(nowhereFeature,['si']);
});
};
var bb_area = groupImages.map(calcAreaBB).flatten();
print('bb_area',bb_area);

// Combine all 3 species
var bin_area = ee.FeatureCollection(rr_area.toList(rr_area.size())
                                   .cat(rc_area.toList(rc_area.size()))
                                   .cat(bb_area.toList(bb_area.size())));
print('bin_area',bin_area);
```

```
// convert output column list to columns
var area_wide = bin_area.map(function(feature){
  var list = ee.List(ee.Feature(feature).get('groups'));
  var keys = list.map(function(o) { return ee.Number(ee.Dictionary(o).get('group')).format('%d') })
  var values = list.map(function(o) { return ee.Dictionary(o).get('sum') })

  return ee.Feature(feature.geometry(), ee.Dictionary.fromLists(keys, values))
    .copyProperties(feature,['BWS_Region','species','si'])
}).select([".*"], null, false);
print('grouped area formatted', area_wide.filter(ee.Filter.eq('BWS_Region','Barwon-Darling')));

// Export
Export.table.toCloudStorage({collection:area_wide,
                    description:'Condition_all_years_mfp_forest_add8',
                    bucket:'mdba_gee_bucket',
                    //fileNamePrefix,
                    fileFormat:'CSV'
                    //, selectors
                    });
```

Script to calculate mean NDVI

```
// 'anae' is the selected ANAE_Wetlands_2017 polygons in assets
```

```
// 'nsw' is the selected Interim_NSW polygons in assets
// 'mfpcol' is the selected floodplain layers as a collection under assets

var first_date = '1986-01-01';
var last_date = '2020-01-01';
var timeframe = 'year';
var advance_overlap = 1; //
var cloud_value = 90;



// Function to cloud mask from the pixel_qa band of Landsat 8 SR data.
function maskL8sr(image) {
  // Bits 3 and 5 are cloud shadow and cloud, respectively.
  var cloudShadowBitMask = 1 << 3;
  var cloudsBitMask = 1 << 5;

  // Get the pixel QA band.
  var qa = image.select('pixel_qa');

  // Both flags should be set to zero, indicating clear conditions.
  var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
      .and(qa.bitwiseAnd(cloudsBitMask).eq(0));

  // Return the masked image, scaled to reflectance, without the QA bands.
  return image.updateMask(mask)//.divide(10000)
      .select("B[0-9]*")
      .copyProperties(image);
}

var cloudMaskL457 = function(image) {
```

```
  var qa = image.select('pixel_qa');
  // If the cloud bit (5) is set and the cloud confidence (7) is high
  // or the cloud shadow bit is set (3), then it's a bad pixel.
  var cloud = qa.bitwiseAnd(1 << 5)
          .and(qa.bitwiseAnd(1 << 7))
          .or(qa.bitwiseAnd(1 << 3))
  // Remove edge pixels that don't occur in all bands
  var mask2 = image.mask().reduce(ee.Reducer.min());
  return image.updateMask(cloud.not()).updateMask(mask2);
};

// Apply cloud mask
var l5c = l5.map(cloudMaskL457);
var l7c = l7.map(cloudMaskL457);
var l8c = l8.map(maskL8sr);

// Function to change landsat 5/7 bandnames
var bands57 = function(img){
  var renamed = img.select('B1','B2','B3','B4','B5','B7').rename('blue','green','red','nir','swir1','swir2');
  return renamed;
};

// Function to change landsat 8 bandnames
var bands8 = function(img){
  var renamed = img.select('B2','B3','B4','B5','B6','B7').rename('blue','green','red','nir','swir1','swir2');
  return renamed;
};

// Apply function to changes landsat band names
var l5f = l5c.filterDate('1984-01-01','2015-12-31').map(bands57);
```

```
var l7f = l7c.filterDate('1999-01-01','2003-05-30').map(bands57);
var l8f = l8c.filterDate('2013-04-11','2019-02-12').map(bands8);



//Function to add date - as band value
var dateBand = function(image) {
  var inx = ee.Image.constant(ee.Number.parse(image.date().format('YYYY'))).float();
  return image.addBands(inx);
};

/*
// This might be a better way of getting the date, but the current way works
//Function to add system:index
var createStepBand = function(image) {
  var inx = ee.Image.constant(ee.Number.parse(image.get('system:time_start'))); //constant_1
  return image.addBands(inx);
};
*/

// Add ndvi
var addNDVI = function(image) {
  var ndvi = image.normalizedDifference(['nir', 'red']).rename('NDVI');
  return image.addBands(ndvi);
};

// This is the image collection the video will be made from
var col = l5f.merge(l7f.merge(l8f))
        .filterBounds(bws.geometry())
        .filterMetadata('CLOUD_COVER','less_than',cloud_value)
```

```
            //.filter(ee.Filter.eq('WRS_PATH', WRS_PATH))
            //.filter(ee.Filter.eq('WRS_ROW', WRS_ROW))
            .map(dateBand)
            .map(addNDVI);
            //.map(createStepBand);
print('Example of image in image collection',col.first());



// Filter for each month, sum, and then sum over the basin
var start = ee.Date(first_date);
var end = ee.Date(last_date);
var months = end.difference(start, timeframe);
var datelist = ee.List.sequence(0, months.int()).map(function(count) {
  return start.advance(count, timeframe);
});
print('This is the list of dates that','the code will try to find images for',datelist);

// This function creates a list of images
var dummy = datelist.map(function(d) {
  var begin = ee.Date(d);
  var end = begin.advance(advance_overlap, timeframe);
  var imgcol = col.filterDate(begin, end);
  var monthsum= imgcol.median();
  return monthsum;
 });
// print('This is the list of images for all dates,','some are empty',dummy,);

// Turn list into image collection
var newcol = ee.ImageCollection(dummy);
print('This is the image collection (with empty images)',newcol);
```

```
// Get rid of empty images
var c = newcol
   .map(function(img) {
     return img.set('count', img.bandNames().length()).copyProperties(img);
   })
   .filter(ee.Filter.gt('count', 1));
print('This is the image collection (with no empty images)',col);


// Get date
var getDate = function(image){
  var date =
ee.Number(image.reduceRegion(ee.Reducer.median(),test.centroid(1),30).get('constant'))
  var dict = ee.Dictionary.fromLists(['year'],[date]);
  return image.set(dict);
};


// Add ndvi to collection
var ndviCol = c.map(getDate).select('NDVI');


print('test year',ee.Image(ndviCol.filter(ee.Filter.eq('year',1999)).first()));
print('Collection',ndviCol);
Map.addLayer(ndviCol.filter(ee.Filter.eq('system:index','13')),{},'test ndvi col');


//////////////////////////////////////////////////////////////////////////
// SCT Part //////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////
// Make floodplain mask
// Make ANAE forest mask
var forests = anae.eq(22)
```

```
            .add(anae.eq(30))
            .add(anae.eq(31))
            .add(anae.eq(35))
            .add(anae.eq(66))
            .gt(0)
            .selfMask();

// Make floodplain mask
var mfp = mfpcol.mosaic().gt(0);
print(mfp);
Map.addLayer(mfp,{min:0,max:1,palette:['blue']},'MFP');



// Make interim NSW forest mask
var nsw_forests = nsw .eq(1)
                .add(nsw.eq(2))
                .add(nsw.eq(4))
                .add(nsw.eq(7))
                .add(nsw.eq(10))
                .add(nsw.eq(11))
                .add(nsw.eq(13))
                .add(nsw.eq(15))
                .add(nsw.eq(16))
                .gt(0)
                .unmask();

// Add all masks together
var mfp_forest_add = forests.unmask()
                    .add(mfp.add(1).unmask())
                    .add(nsw_forests.unmask())
```

```
            .selfMask()
            .multiply(ee.Image(0));
Map.addLayer(mfp_forest_add,{},'mfp_forest_add');

// Make binary area image
var sppmfp = spp;
var rc = sppmfp.mask(spp.eq(3)).divide(ee.Image(3)).add(mfp_forest_add);
var rr = sppmfp.mask(spp.eq(1)).add(mfp_forest_add);
var bb = sppmfp.mask(spp.eq(2)).divide(ee.Image(2)).add(mfp_forest_add);
// Make floodplain mask
// Make ANAE forest mask
var forests = anae.eq(22)
        .add(anae.eq(30))
        .add(anae.eq(31))
        .add(anae.eq(35))
        .add(anae.eq(66))
        .gt(0)
        .selfMask();

// Make floodplain mask
var mfp = mfpcol.mosaic().gt(0);
print(mfp);
Map.addLayer(mfp,{min:0,max:1,palette:['blue']},'MFP');



// Make interim NSW forest mask
var nsw_forests = nsw .eq(1)
          .add(nsw.eq(2))
          .add(nsw.eq(4))
          .add(nsw.eq(7))
```

```
            .add(nsw.eq(10))
            .add(nsw.eq(11))
            .add(nsw.eq(13))
            .add(nsw.eq(15))
            .add(nsw.eq(16))
            .gt(0)
            .unmask();

// Add all masks together
var mfp_forest_add = forests.unmask()
              .add(mfp.add(1).unmask())
              .add(nsw_forests.unmask())
              .selfMask()
              .multiply(ee.Image(0));
Map.addLayer(mfp_forest_add,{},'mfp_forest_add');

// Make binary area image
var sppmfp = spp;
var rc = sppmfp.mask(spp.eq(3)).divide(ee.Image(3)).add(mfp_forest_add);
var rr = sppmfp.mask(spp.eq(1)).add(mfp_forest_add);
var bb = sppmfp.mask(spp.eq(2)).divide(ee.Image(2)).add(mfp_forest_add);
Map.addLayer(rr,{},'rr species map');
Map.addLayer(bb,{},'bb species map');
Map.addLayer(rc,{},'rc species map');
//////////////////////////////////////////////////////////////////////////////////////////

// Function to add species as a property
var addRRG = function(element){
  var dict = {species: 'River Red Gum'};
  var nowhereFeature = ee.Feature(null, dict);
```

```
  return element.copyProperties(nowhereFeature,['species']);
};
var addRC = function(element){
  var dict = {species: 'Coolibah'};
  var nowhereFeature = ee.Feature(null, dict);
  return element.copyProperties(nowhereFeature,['species']);
};
var addBB = function(element){
  var dict = {species: 'Black Box'};
  var nowhereFeature = ee.Feature(null, dict);
  return element.copyProperties(nowhereFeature,['species']);
};




// Function to add year as a property
    var addYear = function(element){
      var dict = {year: ee.String(test1.get('year'))};
      var nowhereFeature = ee.Feature(null, dict);
      return element.copyProperties(nowhereFeature,['year']);
    };

/*
// Function to add year as a property
    var addYear = function(element){
        return element.set({year: ee.String(test.get('year'))});
    };
    */
```

```
//Map.addLayer(rr,{},'rr')


var test1 = ndviCol.first();

Map.addLayer(test1,{min:-1,max:1});

var table = test1.mask(rr)

		.reduceRegions({collection:bws, reducer:ee.Reducer.mean().combine({ reducer2:
ee.Reducer.stdDev(), sharedInputs: true}), scale:test.projection().nominalScale()})

		.map(addRRG)

		//.set({year: ee.String(test.get('year'))})

		.map(addYear);

print('table',table.limit(5));


//print((ee.Feature(null, {species: 'Black Box'}),['species']))

//print((ee.Feature(null, {year: ee.String(test.get('year'))}),['year']))


// Calculate area by group


//var reduceMe = function(image){

var sequence = ee.List.sequence(1986, 2019, 1);

print('sequence',sequence);

// ndviCol.filter(ee.Filter.eq('year',1992))


//------------------------------------------------------------------------------------------------------------------------

// For loop https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for
```

```
sequence.evaluate(function(ids){
  for (var i=0; i<ids.length; i++) {
  //print('i',ee.Number(i).add(1986));
  var image = ndviCol.filter(ee.Filter.eq('year',ee.Number(i).add(1986))).first();
  print('image',image)
  var bin_area_rr = image.mask(rr)
        .reduceRegions({collection:bws, reducer:ee.Reducer.mean().combine({ reducer2:
ee.Reducer.stdDev(), sharedInputs: true}), scale:30})
        .map(addRRG)
        .set({year: ee.Number(i).add(1986)})
        .map(addYear);


  var bin_area_rc = image.mask(rc)
        .reduceRegions({collection:bws, reducer:ee.Reducer.mean().combine({ reducer2:
ee.Reducer.stdDev(), sharedInputs: true}), scale:30})
        .map(addRC)
        .set({year: ee.Number(i).add(1986)})
        .map(addYear);


  var bin_area_bb = image.mask(bb)
        .reduceRegions({collection:bws, reducer:ee.Reducer.mean().combine({ reducer2:
ee.Reducer.stdDev(), sharedInputs: true}), scale:30})
        .map(addBB)
        .set({year: ee.Number(i).add(1986)})
        .map(addYear);


  var reduced =
ee.FeatureCollection(bin_area_rr.toList(bin_area_rr.size()).cat(bin_area_rc.toList(bin_area_rc.size()))
.cat(bin_area_bb.toList(bin_area_bb.size())))
         .select({propertySelectors:['year','BWS_Region','species','mean','stdDev'],
                        //newProperties,
                        retainGeometry:false});
```

```
 Export.table.toCloudStorage({collection:reduced,
                    description:'Take6_annual_ndvi_lt90cc'+ '_' + i,
                    bucket:'mdba_gee_bucket',
                    //fileNamePrefix,
                    fileFormat:'CSV'
                    //, selectors
                    });

 }});

print('projection',test.projection().nominalScale())
//--------------------------------------------------------------------------------------------------------------------------

/*
//};
//var reduced = test1.map(reduceMe).flatten(); //Using flatten makes it 96 instead of 32 features
var new_reduced =
reduced.limit(10).select({propertySelectors:['year','BWS_Region','species','mean','stdDev'],
                              //newProperties,
                              retainGeometry:false})
print('Reduced',new_reduced);
//print('Reduced first',reduced.first());

/*
var test = reduced.first();
var groups = test.get('groups');
print('groups',groups)
var keys = ee.List(groups).map(function(o){return
ee.Number(ee.Dictionary(o).get('group')).format('%d');});
print('keys',keys);
```

```
var values = ee.List(groups).map(function(o){return ee.Number(ee.Dictionary(o).get('sum'))})
print('values',values);
var dict = ee.Dictionary.fromLists(keys,values);
print('dict',dict);



// convert output column list to columns - can't get to work
var area_wide = reduced.map(function(feature){
  var list = ee.List(ee.Feature(feature).get('groups'))
  var keys = list.map(function(o) { return ee.Number(ee.Dictionary(o).get('group')).format('%d') })
  var values = list.map(function(o) { return ee.Dictionary(o).get('sum') })

  return ee.Feature(feature.geometry(), ee.Dictionary.fromLists(keys, values))
    .copyProperties(feature,['species','year'])
})//.select([".*"], null, false);
print('grouped area formatted', area_wide.limit(10));
/*
// Export the table
Export.table.toDrive({collection:area_wide,
                    description:'Area_2000_2002_rr',
                    //folder, fileNamePrefix,
                    fileFormat:'CSV'//, selectors
});
*/

/*

Export.table.toCloudStorage({collection:new_reduced.limit(10),
                         description:'reducedlimit3',
                         bucket:'mdba_gee_bucket',
```

```
//fileNamePrefix,
fileFormat:'CSV'
//, selectors
});


*/
```

Office locations
Adelaide
Albury-Wodonga
Canberra
Goondiwindi
Griffith
Mildura
Murray Bridge
Toowoomba

mdba.gov.au 1800 230 067 engagement@mdba.gov.au