

2020.9.4.5632

A replacements model

The Oats model is constructed from the following list of software components. Details of the implementation and model parameterisation are provided in the following sections.

List of Plant Model Components.

Component Name	Component Type
Arbitrator	Models.PMF.OrganArbitrator
Phenology	Models.PMF.Phen.Phenology
Structure	Models.PMF.Struct.Structure
Grain	Models.PMF.Organs.ReproductiveOrgan
Leaf	Models.PMF.Organs.Leaf
Stem	Models.PMF.Organs.GenericOrgan
Root	Models.PMF.Organs.Root
Panicle	Models.PMF.Organs.GenericOrgan
MortalityRate	Models.Functions.Constant

1 The APSIM Oats Model

Allan Peake, Hamish Brown, Rob Zyskowski, Edmar I. Teixeira, Neil Huth

The APSIM oats model has been developed using the Plant Modelling Framework (PMF) of [Brown et al., 2014](#). This new framework provides a library of plant organ and process submodels that can be coupled, at runtime, to construct a model in much the same way that models can be coupled to construct a simulation. This means that dynamic composition of lower level process and organ classes (e.g. photosynthesis, leaf) into larger constructions (e.g. maize, wheat, sorghum) can be achieved by the model developer without additional coding.

The oats model consists of:

- a phenology model to simulate development through sequential growth phases
- a structure model to simulate plant morphology
- a collection of organs to simulate the various plant parts
- an arbitrator to allocate resources (N, biomass) to the various plant organs

This work builds upon an earlier APSIM Oats model that was constructed in 2007 and then published at the Australian Agronomy Conference in 2008 using data from Gatton, Tarlee and Pinery ([Peake et al., 2008](#)). However numerous changes have been made in the APSIM NextGen version, so simulations run under APSIM 7.10 and previous versions will give different predictions when run with APSIM NextGen.

The Oats model is broadly based on APSIM Wheat models such as NWheat ([S Asseng et al., 2002](#), [BA](#)

Keating, 2001), NWheatS (S Asseng et al., 1998), Cropmod-Wheat (Wang et al., 2002), and the earlier versions developed within the Plant Modelling Framework (Brown et al., 2014).

2 Arbitrator

The Arbitrator class determines the allocation of dry matter (DM) and Nitrogen between each of the organs in the crop model. Each organ can have up to three different pools of biomass:

- **Structural biomass** which is essential for growth and remains within the organ once it is allocated there.
- **Metabolic biomass** which generally remains within an organ but is able to be re-allocated when the organ senesces and may be retranslocated when demand is high relative to supply.
- **Storage biomass** which is partitioned to organs when supply is high relative to demand and is available for retranslocation to other organs whenever supply from uptake, fixation, or re-allocation is lower than demand.

The process followed for biomass arbitration is shown in Figure 2. Arbitration calculations are triggered by a series of events (shown below) that are raised every day. For these calculations, at each step the Arbitrator exchange information with each organ, so the basic computations of demand and supply are done at the organ level, using their specific parameters.

1. **doPotentialPlantGrowth.** When this event occurs, each organ class executes code to determine their potential growth, biomass supplies and demands. In addition to demands for structural, non-structural and metabolic biomass (DM and N) each organ may have the following biomass supplies:
2. **Fixation supply.** From photosynthesis (DM) or symbiotic fixation (N)
3. **Uptake supply.** Typically uptake of N from the soil by the roots but could also be uptake by other organs (eg foliage application of N).
4. **Retranslocation supply.** Storage biomass that may be moved from organs to meet demands of other organs.
5. **Reallocation supply.** Biomass that can be moved from senescing organs to meet the demands of other organs.
6. **doPotentialPlantPartitioning.** On this event the Arbitrator first executes the DoDMSetup() method to gather the DM supplies and demands from each organ, these values are computed at the organ level. It then executes the DoPotentialDMAAllocation() method which works out how much biomass each organ would be allocated assuming N supply is not limiting and sends these allocations to the organs. Each organ then uses their potential DM allocation to determine their N demand (how much N is needed to produce that much DM) and the arbitrator calls DoNSetup() to gather the N supplies and demands from each organ and begin N arbitration. Firstly DoNReallocation() is called to redistribute N that the plant has available from senescing organs. After this step any unmet N demand is considered as plant demand for N uptake from the soil (N Uptake Demand).
7. **doNutrientArbitration.** When this event occurs, the soil arbitrator gets the N uptake demands from each plant (where multiple plants are growing in competition) and their potential uptake from the soil and determines how much of their demand that the soil is able to provide. This value is then passed back to each plant instance as their Nuptake and doNUptakeAllocation() is called to distribute this N between organs.
8. **doActualPlantPartitioning.** On this event the arbitrator call DoNRetranslocation() and DoNFixation() to satisfy any unmet N demands from these sources. Finally, DoActualDMAAllocation is called where DM allocations to each organ are reduced if the N allocation is insufficient to achieve the organs minimum N concentration and final allocations are sent to organs.

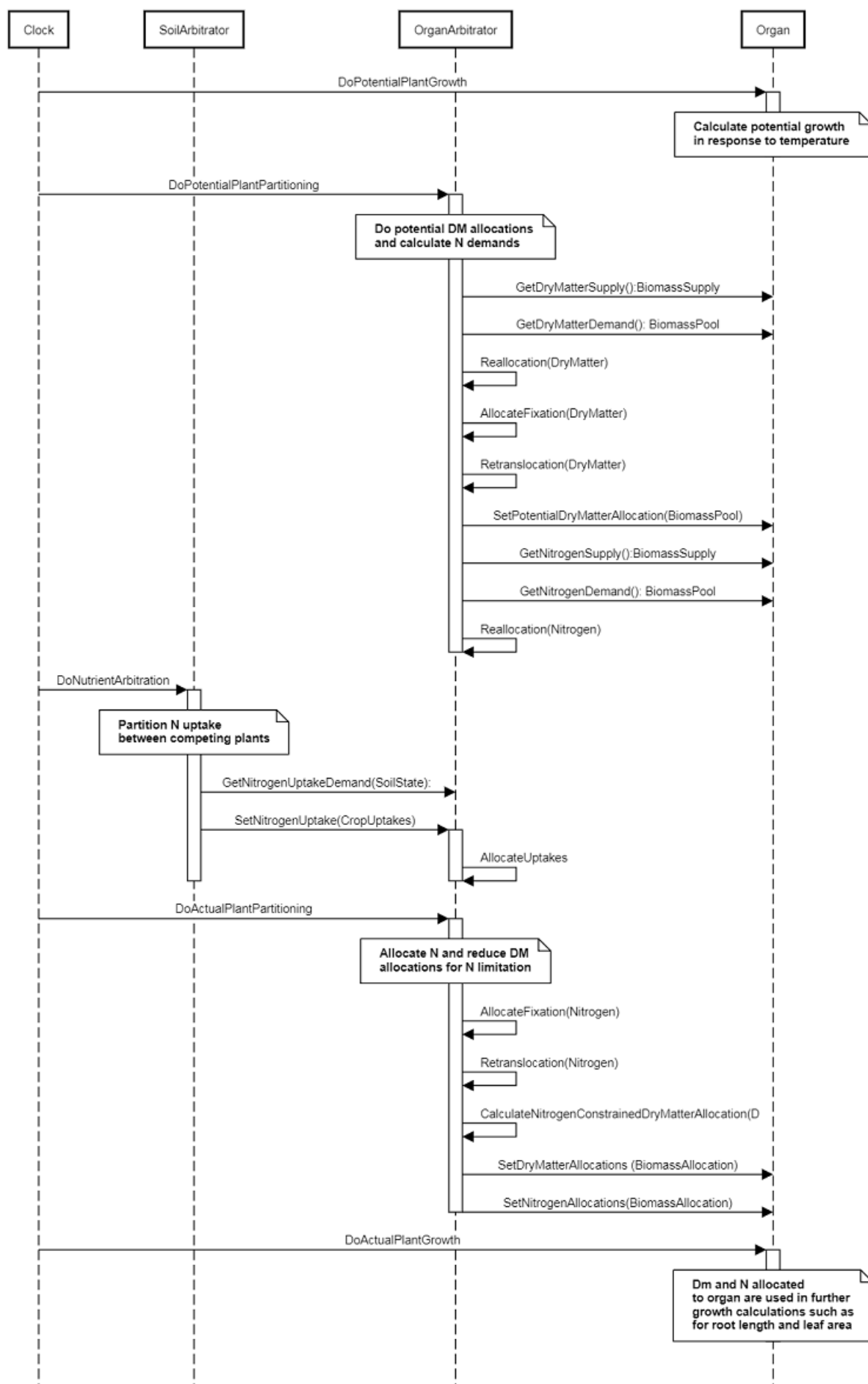


Figure 2: Schematic showing the procedure for arbitration of biomass partitioning. Pink boxes represent events that occur every day and their numbering shows the order of calculations. Blue boxes represent the methods that are called when these events occur. Orange boxes contain properties that make up the organ/arbitrator interface. Green boxes are organ specific properties.

3 Phenology

This model simulates the development of the crop through successive developmental *phases*. Each phase is bound by distinct growth *stages*. Phases often require a target to be reached to signal movement to the next phase. Differences between cultivars are specified by changing the values of the default parameters shown below.

Oat crops exhibit a range of developmental responses to environment and these are strongly influenced by genotype characteristics. Temperature effects development increasing development rates and decreasing phase durations as temperatures increase. These effects are captured by thermal time. However, Oat crops also exhibit vernalisation and photoperiod sensitivities in their Vegetative phase and further photoperiod sensitivity in the EarlyReproductive phase. Photoperiod responses are seen as a reduction in the length of a phase for a photoperiod sensitive genotype in response to a longer photoperiod. Vernalisation responses are more complicated as they are driven by temperature but interact with photoperiod. For vernalisation sensitive varieties (Winter types) exposure to cool temperatures or short photoperiods during the Vegetative phase will reduce the thermal time duration of the vegetative phase.

As used in the APSIM wheat model, we draw on the Kirby Framework to capture these vernalisation and photoperiod responses. This framework assumes that the timing of anthesis is a result of the timing of flag leaf and an additional thermal time passage from there to anthesis. It also assumes the timing of flag leaf is a result of the Final Leaf Number which sets a target, and leaf appearance rate, which sets the rate of progress toward this target. Leaf appearance rate is a function of Thermal time and a genotype specific Phyllochron which changes with Haun stage as described by [Jamieson et al., 1995](#).

Final Leaf Number (FLN) is modelled as the sum of three numbers:

$$FLN = MinLeafNumber + VernalLeaves + PhotoLeaves$$

Where MinLeafNumber is the number of leaves that a wheat crop will produce when vernalisation is satisfied early in the crop's duration (before 2nd true leaf) and it is grown in a long photoperiod. VernalLeaves are the number of leaves that are added due to vernalisation effects. For insensitive varieties this will always be zero but this is potentially a larger number for sensitive varieties and the number progressively decreases as the crop encounters more vernalisation. PhotoLeaves are the number of leaves that are added to the minimum leaf number as a result of short day exposure. For insensitive varieties this will be zero but is potentially larger for more sensitive varieties and decreases as day length increases. More detailed explanations of the components of phenology are provided below.

List of stages and phases used in the simulation of crop phenological development

Phase Number	Phase Name	Initial Stage	Final Stage
1	Germinating	Sowing	Germination
2	Emerging	Germination	Emergence
3	Vegetative	Emergence	TerminalSpikelet
4	StemElongation	TerminalSpikelet	FlagLeaf
5	EarlyReproductive	FlagLeaf	Flowering
6	GrainDevelopment	Flowering	StartGrainFill
7	GrainFilling	StartGrainFill	EndGrainFill
8	Maturing	EndGrainFill	Maturity
9	Ripening	Maturity	HarvestRipe
10	ReadyForHarvesting	HarvestRipe	Unused

3.1 Phenological Phases

3.1.1 Germinating Phase

The model assumes that germination will be completed on the day after sowing, provided that the extractable soil water is greater than zero.

3.1.2 Emerging Phase

This phase goes from Germination to Emergence and simulates time to emergence as a function of sowing depth. The *ThermalTime Target* for ending this phase is given by:

$$Target = SowingDepth \times ShootRate + ShootLag$$

Where:

$$ShootRate = 4 \text{ (deg day/mm),}$$

$$ShootLag = 40 \text{ (deg day),}$$

and *SowingDepth* (mm) is sent from the manager with the sowing event.

Progress toward emergence is driven by Thermal time accumulation, where thermal time is calculated as:

$$ThermalTime = [Phenology].ThermalTime$$

3.1.3 Vegetative Phase

This *phase* goes from Emergence to TerminalSpikelet. It uses a *Target* to determine the duration between development *Stages*. Daily *progress* is accumulated until the *Target* is met and remaining fraction of the day is forwarded to the next phase.

The duration of the vegetative phase is set by a target Haun Stage for the occurrence of Terminal Spikelet and the rate of leaf appearance.

HaunStageTerminalSpikelet decreases as vernalisation is accumulated and as photoperiod extends, capturing the effect of these responses on the duration of the Vegetative phase

The *Target* and the daily *Progression* toward TerminalSpikelet are described as follow:

$$Progression = [Phenology].ThermalTime$$

3.1.3.1 Target

$$Target = [Structure].MeanPhyllochron \times [Structure].HaunStageTerminalSpikelet$$

3.1.4 StemElongation Phase

This phase goes from TerminalSpikelet to FlagLeaf.

It continues until the final main-stem leaf has finished expansion. The duration of this phase is determined by leaf appearance rate (Structure.Phyllochron) and the number of leaves produced on the mainstem (Structure.FinalLeafNumber). As such, the model parameterisation of leaf appearance and final leaf number (set in the Structure model) are important for predicting the duration of the crop correctly.

The Final leaf number is fixed at Terminal Spikelet and leaves continue to appear at a rate set by thermal time and phyllochron until flag leaf ligule appears and this phase is completed.

3.1.5 EarlyReproductive Phase

This *phase* goes from FlagLeaf to Flowering. It uses a *Target* to determine the duration between development *Stages*. Daily *progress* is accumulated until the *Target* is met and remaining fraction of the day is forwarded to the next phase.

The *Target* and the daily *Progression* toward Flowering are described as follow:

$$Progression = [Phenology].ThermalTime$$

3.1.5.1 Target

$$Target = [Structure].Phyllochron.BasePhyllochron \times Phyllochrons$$

Where:

3.1.5.1.1 Phyllochrons

$$\text{Phyllochrons} = [\text{Phenology}].\text{EarlyReproductiveLongDayBase} + \text{IncreaseDueToShortPhotoPeriod}$$

Where:

3.1.5.1.1.1 IncreaseDueToShortPhotoPeriod

$$\text{IncreaseDueToShortPhotoPeriod} = \text{PhotoPeriodResponse} \times [\text{Phenology}].\text{EarlyReproductivePpSensitivity}$$

Where:

3.1.6 GrainDevelopment Phase

This *phase* goes from Flowering to StartGrainFill. It uses a *Target* to determine the duration between development *Stages*. Daily *progress* is accumulated until the *Target* is met and remaining fraction of the day is forwarded to the next phase.

The *Target* and the daily *Progression* toward StartGrainFill are described as follow:

3.1.6.1 Target

$$\text{Target} = 20$$

$$\text{Progression} = [\text{Phenology}].\text{ThermalTime}$$

3.1.7 GrainFilling Phase

This *phase* goes from StartGrainFill to EndGrainFill. It uses a *Target* to determine the duration between development *Stages*. Daily *progress* is accumulated until the *Target* is met and remaining fraction of the day is forwarded to the next phase.

The *Target* and the daily *Progression* toward EndGrainFill are described as follow:

3.1.7.1 Target

$$\text{Target} = 650$$

$$\text{Progression} = [\text{Phenology}].\text{ThermalTime}$$

3.1.8 Maturing Phase

This *phase* goes from EndGrainFill to Maturity. It uses a *Target* to determine the duration between development *Stages*. Daily *progress* is accumulated until the *Target* is met and remaining fraction of the day is forwarded to the next phase.

The *Target* and the daily *Progression* toward Maturity are described as follow:

3.1.8.1 Target

$$\text{Target} = 30$$

$$\text{Progression} = [\text{Phenology}].\text{ThermalTime}$$

3.1.9 Ripening Phase

This *phase* goes from Maturity to HarvestRipe. It uses a *Target* to determine the duration between development *Stages*. Daily *progress* is accumulated until the *Target* is met and remaining fraction of the day is forwarded to the next phase.

The *Target* and the daily *Progression* toward HarvestRipe are described as follow:

3.1.9.1 Target

$$\text{Target} = 100$$

$$\text{Progression} = [\text{Phenology}].\text{ThermalTime}$$

It is the end phase in phenology and the crop will sit, unchanging, in this phase until it is harvested or removed by other method

ThermalTime = [*Phenology*].*ThermalTime*

3.2 MeanTemp

MeanTemp = *TempSum* / 2

Where:

3.2.1 TempSum

TempSum = [*Weather*].*MaxT* + [*Weather*].*MinT*

3.3 ThermalTime

Thermal time determines the rate of developmental progress through many of the crops phases and is used by organs to determining potential growth rates.

3.4 DailyVernalisation

Vernalisation responses are based on those described by [Brown et al., 2013](#). Vernalisation is assumed to be related to the expression of the Vrn1 Gene. Its experssion is accumulated daily and daily upregulation as a function of development (*DeltaHaunStage*) and a *TempResponseProfile* that declines exponentially from a maximum at 1°C to zero at 0°C and at 20°C.

DailyVernalisation = [*Structure*].*DeltaHaunStage* × *TempResponseProfile*

Where:

3.5 Photoperiod

Returns the duration of the day, or photoperiod, in hours. This is calculated using the specified latitude (given in the weather file) and twilight sun angle threshold. If a variable called *ClimateControl.PhotoPeriod* is found in the simulation, it will be used instead.

Twilight = -6 (degrees)

3.6 PerceivedPhotoPeriod

3.6.1 ApexBelowGround

The value of *PerceivedPhotoPeriod* from Germination to Emergence is calculated as follows:

Photoperiod = 0 (h)

3.6.2 ApexAboveGround

The value of *PerceivedPhotoPeriod* from Emergence to HarvestRipe is calculated as follows:

Photoperiod = [*Phenology*].*Photoperiod*

PerceivedPhotoPeriod has a value of zero for phases not specified above

3.7 Vrn1

Vrn1 accumulation begins when Vrn4 experssion is down-regulated to zero and stops, assuming vernalisation saturation, at a value of 1

Vrn1 = *minimum* (*Saturation*, *CurrentExpression*)

Where:

Saturation = 1 (*Relative Experssion*)

3.7.1 CurrentExpression

CurrentExpression is a daily accumulation of the values of functions listed below between the Germination and TerminalSpikelet stages. Function values added to the accumulate total each day are:

3.7.1.1 Vernalisation

IF [Phenology].Vrn4 < [Phenology].VernLag THEN

Return_Zero = 0

ELSE

elseReturn_Vernalisation = [Phenology].DailyVernalisation

3.8 Vrn4

Vrn4 = minimum (MaxExpression, CurrentExpression)

Where:

MaxExpression = [Phenology].VernLag

3.8.1 CurrentExpression

CurrentExpression is a daily accumulation of the values of functions listed below between the Germination and TerminalSpikelet stages. Function values added to the accumulate total each day are:

VernalisationConditioning = [Phenology].DailyVernalisation

3.9 MinimumLeafNumber

MinimumLeafNumber = 7 (Leaves)

Number of leaves the plant will produce when fully vernalised early and grown in long photoperiod

3.10 VrnSensitivity

VrnSensitivity = 0 (Leaves)

The difference between Inherent earliness leaf number and the number of leaves produced when the plant is unvernalsed and grown in long photoperiod

3.11 VernLag

VernLag = 1 (Unitless)

The amount of vernalising temperature required before vernalisation response will be evident. Value of relative to the amount off vernalisation temperature required from the end of the lag until vernalisation saturation. A value of 0 means there is no lag, a value of 1 means the lag is the same (in vernal time) as the vernalisation requirement and a value of 2 means the lag is twice as long as the vernalisation response phase

3.12 PpSensitivity

PpSensitivity = 3 (Leaves)

The reduction in leaf number going from 8 to 16 h Pp

3.13 EarlyReproductivePpSensitivity

EarlyReproductivePpSensitivity = 3

The phyllochrons duration for the plant to go from flag leaf ligual appearance at 16 h Pp compared to the phyllochron duration for the same phase at 8 h Pp.

3.14 EarlyReproductiveLongDayBase

EarlyReproductiveLongDayBase = 3

The phyllochrons duration for the plant to go from flag leaf ligual appearance at 16 h Pp.

3.15 Zadok

This model calculates a Zadok growth stage value based upon the current phenological growth stage within the model. The model uses information regarding germination, emergence, leaf appearance and tiller appearance for early growth stages (Zadok stages 0 to 30). The model then uses simulated phenological growth stages for Zadok stages 30 to 100.

Growth Phase	Description
Germinating	$ZadokStage = 5 \times FractionThroughPhase$
Emerging	$ZadokStage = 5 + 5 \times FractionThroughPhase$
Vegetative	$ZadokStage = 10 + Structure.LeafTipsAppeared$
Reproductive	ZadokStage is interpolated from values of
	stage number using the following table.

Growth Stage	ZadokStage
3.9	30
4.9	33
5.0	39
6.0	65
7.0	71
8.0	87
9.0	90

3.16 FlagLeafDAS

A function is used to provide the date of flag leaf (fully emerged) as days after sowing(DAS).

3.17 FloweringDAS

A function is used to provide flowering date as days after sowing(DAS).

3.18 MaturityDAS

A function is used to provide maturity date as days after sowing(DAS).

4 Structure

The structure model simulates morphological development of the plant to inform the Leaf class when and how many leaves appear and to provide a high estimate for use in calculating potential transpiration.

4.1 Plant and Main-Stem Population

The *Plant.Population* is set at sowing with information sent from a manager script in the Sow method. The *PrimaryBudNumber* is also sent with the Sow method and the main-stem population (*MainStemPopn*) for the crop is calculated as: $MainStemPopn = Plant.Population \times PrimaryBudNumber$ Primary bud number is > 1 for crops like potato and grape vine where there are more than one main-stem per plant

4.2 Main-Stem leaf appearance

Each day the number of main-stem leaf tips appeared (*LeafTipsAppeared*) is calculated as: $LeafTipsAppeared += DeltaTips$ Where $DeltaTips$ is calculated as: $DeltaTips = ThermalTime / Phyllochron$ Where *Phyllochron* is the thermal time duration between the appearance of leaf tip given by:

4.2.1 Phyllochron

This is the thermal time between the emergence of leaf tips. The model used here is based on [Jamieson et al., 1998](#) where leaf appearance could be described by a base phyllochron determined between leaves 2 and 7 and a phyllochron that was 70% of base phyllochron for leaves < 2 and 130% of base phyllochron for leaves > 7

$$Phyllochron = LeafStageFactor \times BasePhyllochron \times PhotoPeriodEffect$$

Where:

4.2.1.1 LeafStageFactor

$$BasePhyllochron = 90 \text{ (oC.d)}$$

4.2.1.2 PhotoPeriodEffect

and *ThermalTime* is given by:

$$ThermalTime = [Phenology].ThermalTime$$

LeafTipsAppeared continues to increase until *FinalLeafNumber* is reached where *FinalLeafNumber* is calculated as:

4.2.2 FinalLeafNumber

4.2.2.1 FinalNodeNumber

$$FinalNodeNumber = [Phenology].MinimumLeafNumber + VernalLeaves + PhotoPLeaves$$

Where:

4.2.2.1.1 VernalLeaves

$$VernalLeaves = [Phenology].VrnSensitivity \times VernalisationReductionFactor$$

Where:

4.2.2.1.1.1 VernalisationReductionFactor

$$VernalisationReductionFactor = 1 - [Phenology].Vrn1$$

4.2.2.1.2 PhotoPLeaves

$$PhotoPLeaves = [Phenology].PpSensitivity \times PhotoPeriodReductionFactor$$

Where:

4.3 Branching and Branch Mortality

The total population of stems (*TotalStemPopn*) is calculated as: $TotalStemPopn = MainStemPopn + NewBranches - NewlyDeadBranches$ Where $NewBranches = MainStemPopn \times BranchingRate$ and *BranchingRate* is given by:

4.3.1 BranchingRate

Potential branching rate is determined by the commonly observed pattern of tillering in wheat, in which each tiller emerges with the third leaf on its parent axis (e.g. first tiller emerges at the same time as the third leaf on the main stem, the first secondary tiller appears with the third leaf on tiller 1). This is described as a simple function of main stem leaf number. This function is varied occasionally between Oat cultivars due to the wide range of plant habits.

$$BranchingRate = PotentialBranchingRate \times StressFactors$$

Where:

4.3.1.1 PotentialBranchingRate

4.3.1.1.1 Vegetative

The value of PotentialBranchingRate from Emergence to TerminalSpikelet is calculated as follows:

4.3.1.1.2 Reproductive

The value of PotentialBranchingRate from TerminalSpikelet to HarvestRipe is calculated as follows:

$$Zero = 0$$

PotentialBranchingRate has a value of zero for phases not specified above

4.3.1.2 StressFactors

$$StressFactors = \text{minimum} (NitrogenEffect, CoverEffect, WaterStressEffect)$$

Where:

4.3.1.2.1 NitrogenEffect

Inadequate Nitrogen supply is assumed to affect tillering prior to any effect on photosynthesis or leaf size.

4.3.1.2.2 CoverEffect

Tillering is said to cease once a threshold fraction of incoming radiation has been achieved. This captures shading effects on tillering described by [Evers et al., 2006](#).

4.3.1.2.3 WaterStressEffect

NewlyDeadBranches is calculated as: $NewlyDeadBranches = (TotalStemPopn - MainStemPopn) \times BranchMortality$ where *BranchMortality* is given by:

4.3.2 BranchMortality

4.3.2.1 MortalityPhase

The value of BranchMortality from TerminalSpikelet to Flowering is calculated as follows:

4.3.2.1.1 Mortality

$$Mortality = MortalityPerDegDay \times [Phenology].ThermalTime$$

Where:

4.3.2.1.1.1 MortalityPerDegDay

BranchMortality has a value of zero for phases not specified above

4.4 Height

The Height of the crop is calculated by the *HeightModel*:

4.5 HeightModel

Height is used by the MicroClimate model to calculate the aerodynamic resistance used for calculation of potential transpiration. Calculates the potential height increment and then multiplies it by the smallest of any children functions (Child functions represent stress).

4.6 MeanTillerGrowthRate

MeanTillerGrowthRate is calculated from a moving average of TillerGrowthRate over a series of 5 days.

This is calculated to represent the supply of assimilate to individual tillers over the past 5 days to use as an index to determine tiller mortality. When tiller numbers are high or total crop assimilate supply is small there will not be enough assimilate to maintain all of the tillers so the model will senesce some.

4.6.1 TillerGrowthRate

$$TillerGrowthRate = [Arbitrator].DM.TotalFixationSupply / [Phenology].ThermalTime / [Structure].TotalStemPopn$$

4.7 MeanPhyllochron

MeanPhyllochron is calculated from a moving average of Phyllochron over a series of 300 days.

$$Phyllochron = [Structure].Phyllochron$$

4.8 HaunStageTerminalSpikelet

The Haun stage at which Terminal Spikelet occurs is determined from final leaf number using the approach described in [Brown et al., 2013](#).

$$HaunStageTerminalSpikelet = ([Structure].FinalLeafNumber - 2.85)/1.1$$

4.9 HaunStage

$$HaunStage = [Leaf].ExpandedCohortNo + [Leaf].NextExpandingLeafProportion$$

4.10 StemSenescenceAge

$$StemSenescenceAge = 0$$

5 Grain

This organ uses a generic model for plant reproductive components. Yield is calculated from its components in terms of organ number and size (for example, grain number and grain size).

As with Barley, Oats grain is typically harvested by farmers still in its husk (i.e. the palea and lemma remain attached to the kernel/caryopsis), while the kernel proper is typically smaller than a wheat grain grown under similar conditions. For simplicity (and in the absence of an extensive grain-oats data set) the Oats model uses the same grain size model as wheat, with the assumption that the weight of the oats kernel includes the weight of the husk (which is not included within kernel weight in the wheat model).

Grain number prediction is also derived directly from the wheat model, estimated as a constant number of grains (default=26) per unit of stem+spike dry matter at anthesis. This approach has known limitations and investigation is underway to improve this function in the future.

5.1 NumberFunction

5.1.1 GrainNumber

$$GrainNumber = GrainsPerGramOfStem \times [StemPlusPanicle].Wt$$

Where:

$$GrainsPerGramOfStem = 26 \text{ (grains)}$$

5.2 InitialGrainProportion

$$InitialGrainProportion = 0.05$$

5.3 MaximumPotentialGrainSize

$$MaximumPotentialGrainSize = 0.032 \text{ (g)}$$

5.4 DMDemandFunction

5.4.1 InitialPhase

The value of DMDemandFunction from Flowering to StartGrainFill is calculated as follows:

5.5 FillingRateFunction

Filling rate is calculated from grain number, a maximum mass to be filled and the duration of the filling process.

$$NumberFunction = [Grain].NumberFunction$$

$$FillingDuration = [Phenology].GrainDevelopment.Target$$

$$ThermalTime = [Phenology].ThermalTime$$

5.5.1 PotentialSizeIncrement

$$PotentialSizeIncrement = [Grain].InitialGrainProportion \times [Grain].MaximumPotentialGrainSize$$

5.5.2 LinearPhase

The value of DMDemandFunction from StartGrainFill to EndGrainFill is calculated as follows:

5.6 FillingRateFunction

Filling rate is calculated from grain number, a maximum mass to be filled and the duration of the filling process.

$$NumberFunction = [Grain].NumberFunction$$

$$FillingDuration = [Phenology].GrainFilling.Target$$

$$ThermalTime = [Phenology].ThermalTime$$

5.6.1 PotentialSizeIncrement

$$PotentialSizeIncrement = ProportionLinearPhase \times [Grain].MaximumPotentialGrainSize$$

Where:

5.6.1.1 ProportionLinearPhase

$$ProportionLinearPhase = 1 - [Grain].InitialGrainProportion$$

DMDemandFunction has a value of zero for phases not specified above

5.7 MinimumNConc

$$MinimumNConc = 0.0123 \text{ (g/g)}$$

5.8 MaxNConcDailyGrowth

$$MaxNConcDailyGrowth = 0.03$$

5.9 MaximumNConc

$$MaximumNConc = 0.03 \text{ (g/g)}$$

5.10 NFillingRate

5.10.1 GrainFilling

The value of NFillingRate from FlagLeaf to EndGrainFill is calculated as follows:

5.11 FillingRateFunction

Filling rate is calculated from grain number, a maximum mass to be filled and the duration of the filling process.

$$NumberFunction = [Grain].NumberFunction$$

5.11.1 FillingDuration

$$FillingDuration = [Phenology].GrainDevelopment.Target + [Phenology].GrainFilling.Target$$

$$ThermalTime = [Phenology].ThermalTime$$

5.11.2 PotentialSizeIncrement

$$PotentialSizeIncrement = [Grain].MaximumPotentialGrainSize \times [Grain].MaximumNConc$$

NFillingRate has a value of zero for phases not specified above

5.12 WaterContent

$$WaterContent = 0.12 \text{ (g/g)}$$

Water content used to calculate a fresh weight.

5.13 AccumThermalTime

AccumThermalTime is a daily accumulation of the values of functions listed below between the FlagLeaf and Maturity stages. Function values added to the accumulate total each day are:

$$DailyThermalTimeValue = [Phenology].ThermalTime$$

5.14 Protein

$$Protein = ([Grain].Live.N + [Grain].Dead.N) / ([Grain].Live.Wt + [Grain].Dead.Wt) \times 100 \times 5.71$$

5.15 BiomassRemovalDefaults

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

Prune, graze and thin have been set to zero as they will vary depending on the needs of the user, and they can be overwritten in the manager script.

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	100	100	0	0
Cut	100	100	0	0
Prune	0	0	0	0
Graze	0	0	0	0
Thin	0	0	0	0

5.16 DMConversionEfficiency

$$DMConversionEfficiency = 1$$

5.17 remobilisationCost

$$remobilisationCost = 0$$

5.18 CarbonConcentration

$$CarbonConcentration = 0.4$$

6 Leaf

The leaves are modelled as a set of leaf cohorts and the properties of each of these cohorts are summed to give overall values for the leaf organ. A cohort represents all the leaves of a given main-stem node position including all of the branch leaves appearing at the same time as the given main-stem leaf ([Lawless et al., 2005](#)). The number of leaves in each cohort is the product of the number of plants per m² and the number of branches per plant. The *Structure* class models the appearance of main-stem leaves and branches. Once cohorts are initiated the *Leaf* class models the area and biomass dynamics of each. It is assumed all the leaves in each cohort have the same size and biomass properties. The modelling of the status and function of individual cohorts is delegated to *LeafCohort* classes.

6.1 Dry Matter Fixation

The most important DM supply from leaf is the photosynthetic fixation supply. Radiation interception is calculated from LAI using an extinction coefficient of:

6.1.1 ExtinctionCoeff

ExtinctionCoeff = 0.45

Using extinction coefficient as per measured at Gatton in 2007 (0.45).

6.2 Photosynthesis

Biomass fixation is modelled as the product of intercepted radiation and its conversion efficiency, the radiation use efficiency (RUE) ([Monteith et al., 1977](#)). This approach simulates net photosynthesis rather than providing separate estimates of growth and respiration. The potential photosynthesis calculated using RUE is then adjusted according to stress factors, these account for plant nutrition (FN), air temperature (FT), vapour pressure deficit (FVPD), water supply (FW) and atmospheric CO₂ concentration (FCO₂). NOTE: RUE in this model is expressed as g/MJ for a whole plant basis, including both above and below ground growth.

RUE = 1.6

WHOLE PLANT RUE = bigger than RUE calculated using above ground biomass. 1.35 measured for Oats at Gatton 2007 (Above ground biomass only) so assume 20% more for root biomass.

6.2.1 FCO₂

This model calculates the CO₂ impact on RUE using the approach of [Reyenga et al., 1999](#).

RadnInt = [*Leaf*].*RadIntTot*

6.3 CO₂internal

CO2internal = (163 - [*IWeather*].*MeanT*)/(5 - 0.1x[*IWeather*].*MeanT*)

6.4 StomatalConductanceCO₂Modifier

StomatalConductanceCO2Modifier = [*Leaf*].*Photosynthesis.FCO2* / *RelativeCO2Gradient*

Where:

6.4.1 RelativeCO₂Gradient

RelativeCO2Gradient = ([*IWeather*].*CO2* - [*Leaf*].*CO2internal*)/(350 - [*Leaf*].*CO2internal*)

Area = 200

Area = 0

6.5 Potential Leaf Area index

Leaf area index is calculated as the sum of the area of each cohort of leaves. The appearance of a new cohort of leaves occurs each time *Structure.LeafTipsAppeared* increases by one. From tip appearance the area of each cohort will increase for a certain number of degree days defined by the *GrowthDuration*

6.5.1 GrowthDuration

GrowthDuration = *Multiplier* × [*Structure*].*Phyllochron* × *AgeFactor*

Where:

$$\text{Multiplier} = 1.75$$

6.5.1.1 AgeFactor

If no stress occurs the leaves will reach a Maximum area (*MaxArea*) at the end of the *GrowthDuration*. The *MaxArea* is defined by:

6.5.2 MaxArea

$$\text{MaxArea} = \text{AreaLargestLeaves} \times \text{AgeFactor}$$

Where:

$$\text{AreaLargestLeaves} = 4000 \text{ (mm}^2\text{)}$$

Default as per NZ Lincoln data set. Amended between varieties due to large variation in Oats plant type.

6.5.2.1 AgeFactor

In the absence of stress the leaf will remain at *MaxArea* for a number of degree days set by the *LagDuration* and then area will senesce to zero at the end of the *SenescenceDuration*

6.5.3 LagDuration

$$\text{LagDuration} = \text{AgeFactor} \times \text{LastLeafDuration}$$

Where:

6.5.3.1 LastLeafDuration

$$\text{LastLeafDuration} = \text{ThermalTimeToRipe} - [\text{Leaf}].\text{CohortParameters}.\text{SenescenceDuration}$$

Where:

6.5.3.1.1 ThermalTimeToRipe

$$\begin{aligned} \text{ThermalTimeToRipe} = & [\text{Phenology}].\text{EarlyReproductive}.\text{Target} + \\ & [\text{Phenology}].\text{GrainDevelopment}.\text{Target} + [\text{Phenology}].\text{GrainFilling}.\text{Target} + \\ & [\text{Phenology}].\text{Maturing}.\text{Target} + [\text{Phenology}].\text{Ripening}.\text{Target} \end{aligned}$$

6.5.4 SenescenceDuration

$$\text{SenescenceDuration} = \text{Multiplier} \times [\text{Structure}].\text{Phyllochron}$$

Where:

$$\text{Multiplier} = 6$$

Mutual shading can cause premature senescence of cohorts if the leaf area above them becomes too great. Each cohort models the proportion of its area that is lost to shade induced senescence each day as:

6.5.5 ShadeInducedSenescenceRate

$$\text{ShadeInducedSenescenceRate} = \text{CoverEffect} \times [\text{Phenology}].\text{ThermalTime}$$

Where:

6.6 Stress effects on Leaf Area Index

Stress reduces leaf area in a number of ways. Firstly, stress occurring prior to the appearance of the cohort can reduce cell division, so reducing the maximum leaf size. Leaf captures this by multiplying the *MaxSize* of each cohort by a *CellDivisionStress* factor which is calculated as:

6.6.1 CellDivisionStress

$$\text{CellDivisionStress} = \text{minimum} (\text{WaterStressFactor}, \text{NitrogenStressFactor})$$

Where:

6.6.1.1 NitrogenStressFactor

Leaf.FN quantifies the N stress status of the plant and represents the concentration of metabolic N relative the maximum potential metabolic N content of the leaf calculated as $(Leaf.NConc - MinimumNConc)/(CriticalNConc - MinimumNConc)$.

Leaf.FW quantifies water stress and is calculated as $Leaf.Transpiration/Leaf.WaterDemand$, where $Leaf.Transpiration$ is the minimum of $Leaf.WaterDemand$ and $Root.WaterUptake$

Stress during the *GrowthDuration* of the cohort reduces the size increase of the cohort by multiplying the potential increase by a *ExpansionStress* factor:

6.6.2 ExpansionStress

$ExpansionStress = \text{minimum}(WaterStress, TemperatureEffect, NitrogenStressFactor)$

Where:

6.6.2.1 NitrogenStressFactor

Stresses can also accelerate the onset and rate of senescence in a number of ways. Nitrogen shortage will cause N to be retranslocated out of lower order leaves to support the expansion of higher order leaves and other organs. When this happens the lower order cohorts will have their area reduced in proportion to the amount of N that is remobilised out of them.

Water stress hastens senescence by increasing the rate of thermal time accumulation in the lag and senescence phases. This is done by multiplying thermal time accumulation by *DroughtInducedLagAcceleration* and *DroughtInducedSenescenceAcceleration* factors, respectively:

6.6.3 DroughtInducedLagAcceleration

$DroughtInducedLagAcceleration = 1 + Stress$

Where:

6.6.3.1 Stress

$Stress = StressResponseCoefficient \times StressFactor$

Where:

$StressResponseCoefficient = 1$

6.6.4 DroughtInducedSenAcceleration

$DroughtInducedSenAcceleration = 1 + Stress$

Where:

6.6.4.1 Stress

$Stress = StressResponseCoefficient \times StressFactor$

Where:

$StressResponseCoefficient = 1$

6.7 Dry matter Demand

Leaf calculates the DM demand from each cohort as a function of the potential size increment (*DeltaPotentialArea*) an specific leaf area bounds. Under non stressed conditions the demand for non-storage DM is calculated as *DeltaPotentialArea* divided by the mean of *SpecificLeafAreaMax* and *SpecificLeafAreaMin*. Under stressed conditions it is calculated as *DeltaWaterConstrainedArea* divided by *SpecificLeafAreaMin*.

6.7.1 SpecificLeafAreaMax

Initial SLA is determined as the average of max and min SLA

6.7.2 SpecificLeafAreaMin

Non-storage DM Demand is then separated into structural and metabolic DM demands using the *StructuralFraction*:

6.7.3 StructuralFraction

$$\text{StructuralFraction} = 0.3$$

The storage DM demand is calculated from the sum of metabolic and structural DM (including today's demands) multiplied by a *NonStructuralFraction*:

Unknown child name: NonStructuralFraction

6.8 Nitrogen Demand

Leaf calculates the N demand from each cohort as a function of the potential DM increment and N concentration bounds. Structural N demand = *PotentialStructuralDMAAllocation* * *MinimumNConc* where:

6.8.1 MinimumNConc

Metabolic N demand is calculated as *PotentialMetabolicDMAAllocation* * (*CriticalNConc* - *MinimumNConc*) where:

6.8.2 CriticalNConc

$$\text{CriticalNConc} = \text{CriticalNConcAt350ppm} \times \text{CO2Factor}$$

Where:

6.8.2.1 CO2Factor

Storage N demand is calculated as the sum of metabolic and structural wt (including today's demands) multiplied by *LuxuryNconc* (*MaximumNConc* - *CriticalNConc*) less the amount of storage N already present. *MaximumNConc* is given by:

6.9 Drymatter supply

In addition to photosynthesis, the leaf can also supply DM by reallocation of senescing DM and retranslocation of storage DM: Reallocation supply is a proportion of the metabolic and non-structural DM that would be senesced each day where the proportion is set by:

6.9.1 DMReallocationFactor

$$\text{DMReallocationFactor} = 1$$

Retranslocation supply is calculated as a proportion of the amount of storage DM in each cohort where the proportion is set by :

6.9.2 DMRetranslocationFactor

$$\text{DMRetranslocationFactor} = 0.8$$

6.10 Nitrogen supply

Nitrogen supply from the leaf comes from the reallocation of metabolic and storage N in senescing material and the retranslocation of metabolic and storage N. Reallocation supply is a proportion of the Metabolic and Storage DM that would be senesced each day where the proportion is set by:

6.10.1 NReallocationFactor

$$\text{NReallocationFactor} = 1$$

Retranslocation supply is calculated as a proportion of the amount of storage and metabolic N in each cohort where the proportion is set by :

6.10.2 NRetranslocationFactor

$$\text{NRetranslocationFactor} = 0.03$$

6.10.3 MaintenanceRespirationFunction

MaintenanceRespirationFunction = 0

6.10.4 DetachmentLagDuration

DetachmentLagDuration = 1000000 (deg day)

6.10.5 DetachmentDuration

DetachmentDuration = 1000000 (deg day)

6.10.6 RelativeBranchLeafSize

RelativeBranchLeafSize = 1

6.10.7 InitialNConc

InitialNConc = 0

6.10.8 StorageFraction

StorageFraction = 0

6.10.9 LeafSizeShapeParameter

LeafSizeShapeParameter = 0.3

6.10.10 SenessingLeafRelativeSize

SenessingLeafRelativeSize = 1 (0-1)

6.10.11 LagDurationAgeMultiplier

LagDurationAgeMultiplier = 1 1 1

6.10.12 SenescenceDurationAgeMultiplier

SenescenceDurationAgeMultiplier = 1 1 1

6.10.13 LeafSizeAgeMultiplier

LeafSizeAgeMultiplier = 1 1 1 1 1 1 1 1 1 1 1 1

6.10.14 RemobilisationCost

RemobilisationCost = 0

6.10.15 CarbonConcentration

CarbonConcentration = 0.4

6.11 FrostFraction

FrostFraction = 0

ThermalTime = [Phenology].ThermalTime

6.12 FRGRFunction

FRGRFunction = minimum (RUE_FT, Others)

Where:

RUE_FT = [Leaf].Photosynthesis.FT

6.12.1 Others

Others = minimum (RUE_FN, RUE_FVPD)

Where:

$$RUE_FN = [Leaf].Photosynthesis.FN$$

$$RUE_FVPD = [Leaf].Photosynthesis.FVPD$$

6.13 BiomassRemovalDefaults

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

Prune, graze and thin have been set to zero as they will vary depending on the needs of the user, and they can be overwritten in the manager script.

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	0	0	80	80
Cut	80	50	10	10
Prune	0	0	0	0
Graze	0	0	0	0
Thin	0	0	0	0

6.14 StructuralFraction

$$StructuralFraction = 0.5$$

6.15 DMConversionEfficiency

$$DMConversionEfficiency = 1$$

6.16 remobilisationCost

$$remobilisationCost = 0$$

6.17 CarbonConcentration

$$CarbonConcentration = 0.4$$

6.18 WidthFunction

$$WidthFunction = 0$$

$$DepthFunction = [Leaf].Height$$

7 Stem

This organ is simulated using a GenericOrgan type. It is parameterised to calculate the growth, senescence, and detachment of any organ that does not have specific functions.

7.1 Dry Matter Demand

The dry matter demand for the organ is calculated as defined in DMDemands, based on the DMDemandFunction and partition fractions for each biomass pool.

7.1.1 DMDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

7.1.1.1 Structural

$$\text{Structural} = \text{DMDemandFunction} \times \text{StructuralFraction}$$

Where:

7.1.1.1.1 DMDemandFunction

$$\text{DMDemandFunction} = \text{PartitionFraction} \quad [\text{Arbitrator}].\text{DM}.\text{TotalFixationSupply}$$

Where:

7.1.1.1.1.1 PartitionFraction

The value of PartitionFraction from Emergence to TerminalSpikelet is calculated as follows:

$$\text{StemFraction} = 0.4$$

The value of PartitionFraction from TerminalSpikelet to FlagLeaf is calculated as follows:

$$\text{StemFraction} = 0.7$$

The value of PartitionFraction from FlagLeaf to Flowering is calculated as follows:

$$\text{StemFraction} = 0.7$$

PartitionFraction has a value of zero for phases not specified above

7.1.1.1.2 StructuralFraction

7.1.1.1.2.1 VegetativeGrowth

The value of StructuralFraction from Sowing to Flowering is calculated as follows:

$$\text{Fraction} = 0.7$$

7.1.1.1.2.2 ReproductiveGrowth

The value of StructuralFraction from Flowering to HarvestRipe is calculated as follows:

$$\text{Fraction} = 0.5$$

StructuralFraction has a value of zero for phases not specified above

7.1.1.2 Metabolic

$$\text{Metabolic} = 0 \text{ (g/m}^2\text{)}$$

7.1.1.3 Storage

The partitioning of daily growth to storage biomass is based on a storage fraction.

7.1.1.3.1 StorageFraction

$$\text{StorageFraction} = 1 - [\text{Stem}].\text{DMDemands}.\text{Structural}.\text{StructuralFraction}$$

7.2 Nitrogen Demand

The N demand is calculated as defined in NDemands, based on DM demand the N concentration of each biomass pool.

7.2.1 NDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

7.2.1.1 Structural

$$\text{Structural} = [\text{Stem}].\text{minimumNconc} \times [\text{Stem}].\text{potentialDMAAllocation}.\text{Structural}$$

7.2.1.2 Metabolic

$$\text{Metabolic} = \text{MetabolicNconc} \times [\text{Stem}].\text{potentialDMAAllocation}.\text{Structural}$$

Where:

7.2.1.2.1 MetabolicNconc

$$\text{MetabolicNconc} = [\text{Stem}].\text{criticalNConc} - [\text{Stem}].\text{minimumNconc}$$

7.2.1.3 Storage

The partitioning of daily growth to storage biomass attempts to bring the organ's N content to the maximum concentration.

$$\text{Storage} = [\text{Stem}].\text{maximumNconc} \times ([\text{Stem}].\text{Live.Wt} + \text{potentialAllocationWt}) - [\text{Stem}].\text{Live.N}$$

The demand for storage N is further reduced by a factor specified by the [Stem].NitrogenDemandSwitch.

7.2.2 MinimumNConc

$$\text{MinimumNConc} = 0.0018$$

$$\text{CriticalNConc} = [\text{Stem}].\text{MinimumNConc}$$

7.2.3 MaximumNConc

The demand for N is reduced by a factor specified by the NitrogenDemandSwitch.

7.2.4 NitrogenDemandSwitch

A value of 1 is returned if phenology is between Emergence and StartGrainFill phases, otherwise a value of 0 is returned.

7.3 Dry Matter Supply

Stem does not reallocate DM when senescence of the organ occurs.

The proportion of non-structural DM that is allocated each day is quantified by the DMReallocationFactor.

7.3.1 DMRetranslocationFactor

7.3.1.1 VegetativeGrowth

The value of DMRetranslocationFactor from Emergence to StartGrainFill is calculated as follows:

$$\text{DMRetranslocationFactor} = 0$$

7.3.1.2 ReproductiveGrowth

The value of DMRetranslocationFactor from StartGrainFill to EndGrainFill is calculated as follows:

$$\text{DMRetranslocationFactor} = 0$$

DMRetranslocationFactor has a value of zero for phases not specified above

7.4 Nitrogen Supply

Stem does not reallocate N when senescence of the organ occurs.

The proportion of non-structural N that is allocated each day is quantified by the NReallocationFactor.

7.4.1 NRetranslocationFactor

7.4.1.1 VegetativeGrowth

The value of NRetranslocationFactor from Sowing to Flowering is calculated as follows:

$$\text{Fraction} = 0$$

7.4.1.2 ReproductiveGrowth

The value of NRetranslocationFactor from Flowering to HarvestRipe is calculated as follows:

$$\text{Fraction} = 0.5$$

NRetranslocationFactor has a value of zero for phases not specified above

7.5 Senescence and Detachment

The proportion of live biomass that senesces and moves into the dead pool each day is quantified by the SenescenceRate.

7.5.1 SenescenceRate

$SenescenceRate = SenescenceRate$

Where:

7.5.1.1 SenescenceRate

Stem has detachment parameterised to zero so all biomass in this organ will remain with the plant until a defoliation or harvest event occurs.

8 Root

The generic root model calculates root growth in terms of rooting depth, biomass accumulation and subsequent root length density in each soil layer.

Root Growth

Roots grow downwards through the soil profile, with initial depth determined by sowing depth and the growth rate determined by RootFrontVelocity. The RootFrontVelocity is modified by multiplying it by the soil's XF value; which represents any resistance posed by the soil to root extension. Root depth is also constrained by a maximum root depth.

Root length growth is calculated using the daily DM partitioned to roots and a specific root length. Root proliferation in layers is calculated using an approach similar to the generalised equimarginal criterion used in economics. The uptake of water and N per unit root length is used to partition new root material into layers of higher 'return on investment'.

Dry Matter Demands

A daily DM demand is provided to the organ arbitrator and a DM supply returned. By default, 100% of the dry matter (DM) demanded from the root is structural. The daily loss of roots is calculated using a SenescenceRate function. All senesced material is automatically detached and added to the soil FOM.

Nitrogen Demands

The daily structural N demand from root is the product of total DM demand and the minimum N concentration. Any N above this is considered Storage and can be used for retranslocation and/or reallocation as the respective factors are set to values other than zero.

Nitrogen Uptake

Potential N uptake by the root system is calculated for each soil layer (i) that the roots have extended into. In each layer potential uptake is calculated as the product of the mineral nitrogen in the layer, a factor controlling the rate of extraction (kNO3 or kNH4), the concentration of N form (ppm), and a soil moisture factor (NUptakeSWFactor) which typically decreases as the soil dries.

$$NO3 \text{ uptake} = NO3_i \times kNO3 \times NO3_{ppm, i} \times NUptakeSWFactor$$

$$NH4 \text{ uptake} = NH4_i \times kNH4 \times NH4_{ppm, i} \times NUptakeSWFactor$$

Nitrogen uptake demand is limited to the maximum daily potential uptake (MaxDailyNUptake) and the plants N demand. The demand for soil N is then passed to the soil arbitrator which determines how much of the N uptake demand each plant instance will be allowed to take up.

Water Uptake

Potential water uptake by the root system is calculated for each soil layer that the roots have extended into. In each layer potential uptake is calculated as the product of the available water in the layer (water above LL limit) and a factor controlling the rate of extraction (KL). The values of both LL and KL are set in the soil interface and KL may be further modified by the crop via the KLModifier function.

$$SW \text{ uptake} = (SW_i - LL_i) \times KL_i \times KLModifier$$

8.1 KLModifier

KLModifier = 1

8.2 SoilWaterEffect

SoilWaterEffect = 1

8.3 MaxDailyNUptake

MaxDailyNUptake = 20

8.4 SenescenceRate

SenescenceRate = 0.005

8.5 MaximumRootDepth

MaximumRootDepth = 1000000

8.6 MaximumNConc

MaximumNConc = 0.01

8.7 MinimumNConc

MinimumNConc = 0.01

8.8 NitrogenDemandSwitch

A value of 1 is returned if phenology is between Germination and Maturity phases, otherwise a value of 0 is returned.

8.9 BiomassRemovalDefaults

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

Prune, graze and thin have been set to zero as they will vary depending on the needs of the user, and they can be overwritten in the manager script.

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	0	0	100	100
Cut	0	0	50	50
Prune	0	0	0	0
Graze	0	0	0	0
Thin	0	0	0	0

8.10 KNO3

KNO3 = 0.02

8.11 KNH4

KNH4 = 0.01

8.12 SpecificRootLength

SpecificRootLength = 105 (m/g)

8.13 RootFrontVelocity

RootFrontVelocity = PotentialRootFrontVelocity × TemperatureFactor × WaterFactor

Where:

8.13.1 PotentialRootFrontVelocity

8.13.1.1 PreEmergence

The value of PotentialRootFrontVelocity from Germination to Emergence is calculated as follows:

Value = 5 (mm/d)

8.13.1.2 PostEmergence

The value of PotentialRootFrontVelocity from Emergence to Maturity is calculated as follows:

Value = 20 (mm/d)

PotentialRootFrontVelocity has a value of zero for phases not specified above

8.14 NUptakeSWFactor

This is modelled in the same way as the old wheat model where potential N uptake is decreased as the soil dries as described by NUptakeSWFactor

8.15 DMConversionEfficiency

DMConversionEfficiency = 1

8.16 MaintenanceRespirationFunction

MaintenanceRespirationFunction = 1

8.17 remobilisationCost

remobilisationCost = 0

8.18 CarbonConcentration

CarbonConcentration = 0.4

8.19 DMDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

8.19.1 Structural

Structural = DMDemandFunction × StructuralFraction

Where:

8.19.1.1 DMDemandFunction

DMDemandFunction = PartitionFraction [Arbitrator].DM.TotalFixationSupply

Where:

8.19.1.1.1 PartitionFraction

8.19.1.1.1.1 PreEmergence

The value of PartitionFraction from Germination to Emergence is calculated as follows:

$$Value = 0$$

8.19.1.1.1.2 PreFlowering

The value of PartitionFraction from Emergence to Flowering is calculated as follows:

8.19.1.1.1.3 PostFlowering

The value of PartitionFraction from Flowering to EndGrainFill is calculated as follows:

$$Value = 0.2$$

PartitionFraction has a value of zero for phases not specified above

$$StructuralFraction = 1$$

8.19.2 Metabolic

$$Metabolic = 0 \text{ (g/m}^2\text{)}$$

8.19.3 Storage

The partitioning of daily growth to storage biomass is based on a storage fraction.

8.19.3.1 StorageFraction

$$StorageFraction = 1 - [Root].DMDemands.Structural.StructuralFraction$$

8.20 NDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

8.20.1 Structural

$$Structural = [Root].minimumNconc \times [Root].potentialDMAAllocation.Structural$$

8.20.2 Metabolic

$$Metabolic = MetabolicNconc \times [Root].potentialDMAAllocation.Structural$$

Where:

8.20.2.1 MetabolicNconc

$$MetabolicNconc = [Root].criticalNConc - [Root].minimumNconc$$

8.20.3 Storage

The partitioning of daily growth to storage biomass attempts to bring the organ's N content to the maximum concentration.

$$Storage = [Root].maximumNconc \times ([Root].Live.Wt + potentialAllocationWt) - [Root].Live.N$$

The demand for storage N is further reduced by a factor specified by the [Root].NitrogenDemandSwitch.

$$CriticalNConc = [Root].MinimumNConc$$

8.21 InitialWt

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

8.21.1 Structural

$$Structural = 0.005 \text{ (g/plant)}$$

8.21.2 Metabolic

$$Metabolic = 0 \text{ (g/m}^2\text{)}$$

8.21.3 Storage

Storage = 0 (g/m²)

8.22 RootDepthStressFactor

RootDepthStressFactor = 1

9 Panicle

This organ is simulated using a GenericOrgan type. It is parameterised to calculate the growth, senescence, and detachment of any organ that does not have specific functions.

9.1 Dry Matter Demand

The dry matter demand for the organ is calculated as defined in DMDemands, based on the DMDemandFunction and partition fractions for each biomass pool.

9.1.1 DMDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

9.1.1.1 Structural

Structural = DMDemandFunction × StructuralFraction

Where:

9.2 DMDemandFunction

Demand is calculated from the product of growth rate, thermal time and population.

ThermalTime = [Phenology].ThermalTime

9.2.1 ExpansionStress

ExpansionStress = minimum (WaterStress, TemperatureEffect, NitrogenStressFactor)

Where:

9.2.1.1 NitrogenStressFactor

OrganPopulation = [Panicle].HeadNumber

StartStage = 4.8

Stage when organ growth starts

MaximumOrganWt = 0.4 (g)

Size individual organs will grow to when fully supplied with DM

9.2.2 GrowthDuration

GrowthDuration = [Phenology].EarlyReproductive.Target + [Phenology].GrainDevelopment.Target

StructuralFraction = 0.9

9.2.2.1 Metabolic

Metabolic = 0 (g/m²)

9.2.2.2 Storage

The partitioning of daily growth to storage biomass is based on a storage fraction.

9.2.2.2.1 StorageFraction

StorageFraction = 1 - [Panicle].DMDemands.Structural.StructuralFraction

9.3 Nitrogen Demand

The N demand is calculated as defined in NDemands, based on DM demand the N concentration of each biomass pool.

9.3.1 NDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

9.3.1.1 Structural

$$Structural = [Panicle].minimumNconc \times [Panicle].potentialDMAAllocation.Structural$$

9.3.1.2 Metabolic

$$Metabolic = MetabolicNconc \times [Panicle].potentialDMAAllocation.Structural$$

Where:

9.3.1.2.1 MetabolicNconc

$$MetabolicNconc = [Panicle].criticalNConc - [Panicle].minimumNconc$$

9.3.1.3 Storage

The partitioning of daily growth to storage biomass attempts to bring the organ's N content to the maximum concentration.

$$Storage = [Panicle].maximumNconc \times ([Panicle].Live.Wt + potentialAllocationWt) - [Panicle].Live.N$$

The demand for storage N is further reduced by a factor specified by the [Panicle].NitrogenDemandSwitch.

9.3.2 MinimumNConc

$$MinimumNConc = 0.02$$

$$CriticalNConc = [Panicle].MinimumNConc$$

9.3.3 MaximumNConc

$$MaximumNConc = 0.025$$

The demand for N is reduced by a factor specified by the NitrogenDemandSwitch.

9.3.4 NitrogenDemandSwitch

A value of 1 is returned if phenology is between Emergence and StartGrainFill phases, otherwise a value of 0 is returned.

9.4 Dry Matter Supply

Panicle does not reallocate DM when senescence of the organ occurs.

The proportion of non-structural DM that is allocated each day is quantified by the DMReallocationFactor.

9.4.1 DMRetranslocationFactor

9.4.1.1 VegetativeGrowth

The value of DMRetranslocationFactor from Emergence to StartGrainFill is calculated as follows:

$$DMRetranslocationFactor = 0$$

9.4.1.2 ReproductiveGrowth

The value of DMRetranslocationFactor from StartGrainFill to EndGrainFill is calculated as follows:

$$DMRetranslocationFactor = 0.5$$

DMRetranslocationFactor has a value of zero for phases not specified above

9.5 Nitrogen Supply

Panicle does not reallocate N when senescence of the organ occurs.

Panicle will retranslocate 30% of non-structural N each day.

9.6 Senescence and Detachment

Panicle has senescence parameterised to zero so all biomass in this organ will remain alive.

Panicle has detachment parameterised to zero so all biomass in this organ will remain with the plant until a defoliation or harvest event occurs.

10 AboveGround Biomass

This is a composite biomass class, representing the sum of 1 or more biomass objects.

AboveGround summarises the following biomass objects:

- [Leaf].Live
- [Leaf].Dead
- [Stem].Live
- [Stem].Dead
- [Panicle].Live
- [Panicle].Dead
- [Grain].Live
- [Grain].Dead

11 AboveGroundLive Biomass

This is a composite biomass class, representing the sum of 1 or more biomass objects.

AboveGroundLive summarises the following biomass objects:

- [Leaf].Live
- [Stem].Live
- [Panicle].Live
- [Grain].Live

12 AboveGroundDead Biomass

This is a composite biomass class, representing the sum of 1 or more biomass objects.

AboveGroundDead summarises the following biomass objects:

- [Leaf].Dead
- [Stem].Dead
- [Panicle].Dead
- [Grain].Dead

13 BelowGround Biomass

This is a composite biomass class, representing the sum of 1 or more biomass objects.

BelowGround summarises the following biomass objects:

- [Root].Live
- [Root].Dead

14 Total Biomass

This is a composite biomass class, representing the sum of 1 or more biomass objects.

Total summarises the following biomass objects:

- [Leaf].Live

- [Leaf].Dead
- [Stem].Live
- [Stem].Dead
- [Root].Live
- [Root].Dead
- [Panicle].Live
- [Panicle].Dead
- [Grain].Live
- [Grain].Dead

15 TotalLive Biomass

This is a composite biomass class, representing the sum of 1 or more biomass objects.

TotalLive summarises the following biomass objects:

- [Leaf].Live
- [Stem].Live
- [Root].Live
- [Panicle].Live
- [Grain].Live

16 TotalDead Biomass

This is a composite biomass class, representing the sum of 1 or more biomass objects.

TotalDead summarises the following biomass objects:

- [Leaf].Dead
- [Stem].Dead
- [Root].Dead
- [Panicle].Dead
- [Grain].Dead

17 Ear Biomass

This is a composite biomass class, representing the sum of 1 or more biomass objects.

The Ear composite biomass object includes the Live and dead pools of all organs within the Ear to allow output for total dry matter and Nutrient contents, including total structural, non-structural and metabolic pools.

Ear summarises the following biomass objects:

- [Panicle].Live
- [Grain].Live
- [Panicle].Dead
- [Grain].Dead

18 StemPlusPanicle Biomass

This is a composite biomass class, representing the sum of 1 or more biomass objects.

StemPlusPanicle summarises the following biomass objects:

- [Stem].Live
- [Panicle].Live
- [Stem].Dead
- [Panicle].Dead

19 Cultivars

Most crop parameters can be specified individually for cultivars. Customized cultivar parameters are reported in this section.

Aladdin, Algerian, Bond, Brusher, Comet, Coolibah, Drover, Drummond, Drummond_orig, Genie, Hokonui, Iapar61Ibipora, Kangaroo, Milton, Nugene, Stampede, Taipan, Wintaroo, Wizard

19.1 Aladdin

This cultivar is defined by overriding some of the base parameters of the plant model.

Aladdin makes the following changes:

```
[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 10
[Phenology].PpSensitivity.FixedValue = 7.5
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 3
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 3
[Structure].Phyllochron.BasePhyllochron.FixedValue = 90
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 8000
```

19.2 Algerian

This cultivar is defined by overriding some of the base parameters of the plant model.

Algerian makes the following changes:

```
[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 12
[Phenology].PpSensitivity.FixedValue = 6
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 4
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 3
[Structure].Phyllochron.BasePhyllochron.FixedValue = 105
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 5000
[Structure].BranchingRate.StressFactors.CoverEffect.XYPairs.X = 0,0.4,0.7
```

19.3 Bond

This cultivar is defined by overriding some of the base parameters of the plant model.

Bond makes the following changes:

```
[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 10
[Phenology].PpSensitivity.FixedValue = 7.5
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 3
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 3
[Structure].Phyllochron.BasePhyllochron.FixedValue = 90
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 8000
```

19.4 Brusher

This cultivar is defined by overriding some of the base parameters of the plant model.

Brusher makes the following changes:

```
[Phenology].MinimumLeafNumber.FixedValue = 7
[Phenology].VrnSensitivity.FixedValue = 1
[Phenology].PpSensitivity.FixedValue = 1
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 0
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 2
[Structure].Phyllochron.BasePhyllochron.FixedValue = 80
```

19.5 Comet

This cultivar is defined by overriding some of the base parameters of the plant model.

Comet makes the following changes:

```
[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 10
[Phenology].PpSensitivity.FixedValue = 7.5
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 3
```

[Phenology].EarlyReproductiveLongDayBase.FixedValue = 3
[Structure].Phyllochron.BasePhyllochron.FixedValue = 90
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 8000

19.6 Coolibah

This cultivar is defined by overriding some of the base parameters of the plant model.

Coolibah makes the following changes:

[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 5.3
[Phenology].PpSensitivity.FixedValue = 3.0
[Structure].Phyllochron.BasePhyllochron.FixedValue = 90
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 5000
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 0
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 2
[Structure].BranchingRate.StressFactors.CoverEffect.XYPairs.X = 0,0.25,0.5

19.7 Drover

This cultivar is defined by overriding some of the base parameters of the plant model.

Drover makes the following changes:

[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 4
[Phenology].PpSensitivity.FixedValue = 9
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 6
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 3
[Structure].Phyllochron.BasePhyllochron.FixedValue = 80
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 8000

19.8 Drummond

This cultivar is defined by overriding some of the base parameters of the plant model.

Drummond makes the following changes:

[Phenology].MinimumLeafNumber.FixedValue = 8.5
[Phenology].VrnSensitivity.FixedValue = 7
[Phenology].PpSensitivity.FixedValue = 3
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 3
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 2
[Structure].Phyllochron.BasePhyllochron.FixedValue = 75
[Structure].BranchingRate.StressFactors.CoverEffect.XYPairs.X = 0,0.2,0.35
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 4000
[Structure].BranchingRate.PotentialBranchingRate.Vegetative.PotentialBranchingRate.XYPairs.X = 1,2,3.99,4,5,6,7,8,9
[Structure].BranchingRate.PotentialBranchingRate.Vegetative.PotentialBranchingRate.XYPairs.Y = 0,0,0,1,2,4,7,12,20
[Leaf].CohortParameters.GrowthDuration.AgeFactor.XYPairs.Y = 1,1,0.1
[Grain].MaximumPotentialGrainSize.FixedValue = 0.05
[Grain].NumberFunction.GrainNumber.GrainsPerGramOfStem.FixedValue = 40

19.9 Drummond_orig

This cultivar is defined by overriding some of the base parameters of the plant model.

Drummond_orig makes the following changes:

[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 0
[Phenology].PpSensitivity.FixedValue = 10
[Structure].Phyllochron.BasePhyllochron.FixedValue = 80
[Structure].BranchingRate.StressFactors.CoverEffect.XYPairs.X = 0,0.2,0.35
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 4000
[Structure].BranchingRate.PotentialBranchingRate.Vegetative.PotentialBranchingRate.XYPairs.X = 1,2,3.99,4,5,6,7,8,9

19.10 Genie

This cultivar is defined by overriding some of the base parameters of the plant model.

Genie makes the following changes:

```
[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 10
[Phenology].PpSensitivity.FixedValue = 7.5
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 3
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 3
[Structure].Phyllochron.BasePhyllochron.FixedValue = 90
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 8000
```

19.11 Kangaroo

This cultivar is defined by overriding some of the base parameters of the plant model.

Kangaroo makes the following changes:

```
[Phenology].MinimumLeafNumber.FixedValue = 7
[Phenology].VrnSensitivity.FixedValue = 4
[Phenology].PpSensitivity.FixedValue = 4
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 0
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 2
[Structure].Phyllochron.BasePhyllochron.FixedValue = 90
```

19.12 Hokonui

This cultivar is defined by overriding some of the base parameters of the plant model.

Hokonui makes the following changes:

```
[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 0
[Phenology].PpSensitivity.FixedValue = 10
[Structure].Phyllochron.BasePhyllochron.FixedValue = 90
[Leaf].CohortParameters.GrowthDuration.AgeFactor.XYPairs.Y = 1,1,0.1
```

19.13 Milton

This cultivar is defined by overriding some of the base parameters of the plant model.

Milton makes the following changes:

```
[Phenology].MinimumLeafNumber.FixedValue = 8.5
[Phenology].VrnSensitivity.FixedValue = 7
[Phenology].PpSensitivity.FixedValue = 3
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 2
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 0
[Structure].Phyllochron.BasePhyllochron.FixedValue = 105
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 4000
[Leaf].CohortParameters.SenescenceDuration.Multiplier.FixedValue = 3
[Leaf].CohortParameters.GrowthDuration.AgeFactor.XYPairs.Y = 1,1,0.1
[Grain].MaximumPotentialGrainSize.FixedValue = 0.05
[Grain].NumberFunction.GrainNumber.GrainsPerGramOfStem.FixedValue = 50
[Leaf].ExtinctionCoeff.FixedValue = 0.6
```

19.14 Nugene

This cultivar is defined by overriding some of the base parameters of the plant model.

Nugene makes the following changes:

```
[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 10
[Phenology].PpSensitivity.FixedValue = 7.5
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 3
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 3
```

[Structure].Phyllochron.BasePhyllochron.FixedValue = 90
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 8000

19.15 Stampede

This cultivar is defined by overriding some of the base parameters of the plant model.

Stampede makes the following changes:

[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 0
[Phenology].PpSensitivity.FixedValue = 10
[Structure].Phyllochron.BasePhyllochron.FixedValue = 80
[Leaf].CohortParameters.GrowthDuration.AgeFactor.XYPairs.Y = 1,1,0.1

19.16 Taipan

This cultivar is defined by overriding some of the base parameters of the plant model.

Taipan makes the following changes:

[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 10
[Phenology].PpSensitivity.FixedValue = 7.5
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 3
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 3
[Structure].Phyllochron.BasePhyllochron.FixedValue = 90
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 8000

19.17 Wintaroo

This cultivar is defined by overriding some of the base parameters of the plant model.

Wintaroo makes the following changes:

[Phenology].MinimumLeafNumber.FixedValue = 7
[Phenology].VrnSensitivity.FixedValue = 2
[Phenology].PpSensitivity.FixedValue = 2
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 0
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 2
[Structure].Phyllochron.BasePhyllochron.FixedValue = 90

19.18 Wizard

This cultivar is defined by overriding some of the base parameters of the plant model.

Wizard makes the following changes:

[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 10
[Phenology].PpSensitivity.FixedValue = 7.5
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 3
[Phenology].EarlyReproductiveLongDayBase.FixedValue = 3
[Structure].Phyllochron.BasePhyllochron.FixedValue = 90
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 8000

19.19 lapar61Ibipora

This cultivar is defined by overriding some of the base parameters of the plant model.

Iapar61Ibipora makes the following changes:

[Phenology].MinimumLeafNumber.FixedValue = 9
[Phenology].VrnSensitivity.FixedValue = 13
[Phenology].VernLag.FixedValue = 2
[Phenology].PpSensitivity.FixedValue = 6
[Phenology].EarlyReproductivePpSensitivity.FixedValue = 2
[Structure].BranchingRate.PotentialBranchingRate.Vegetative.PotentialBranchingRate.XYPairs.Y = 0,0,0.8,0.5,0.5,0.2,0,0

[Structure].BranchingRate.StressFactors.CoverEffect.XYPairs.X = 0,0.4,0.7
[Grain].MaximumPotentialGrainSize.FixedValue = 0.015
[Leaf].CohortParameters.GrowthDuration.Multiplier.FixedValue = 4
[Leaf].CohortParameters.MaxArea.AreaLargestLeaves.FixedValue = 5000
[Leaf].CohortParameters.SpecificLeafAreaMax.XYPairs.X = 2,3.05,3.1,4,6,7,8
[Leaf].CohortParameters.SpecificLeafAreaMax.XYPairs.Y = 100,3200,45000,25000,12000,8000,6000
[Leaf].CohortParameters.SpecificLeafAreaMin.XYPairs.Y = 6000,6000
[Root].DMDemands.Structural.DMDemandFunction.PartitionFraction.PreFlowering.AgeFactor.XYPairs.Y = 0.2,0.2
[Panicum].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 0.17

20 MortalityRate

MortalityRate = 0

21 DataStore

A storage service for reading and writing to/from a database.

21.1 ObservedPred

Reads the contents of a specific sheet from an EXCEL file and stores into the DataStore.

21.2 PredictedObserved

Reads the contents of a file (in apsim format) and stores into the DataStore. If the file has a column name of 'SimulationName' then this model will only input data for those rows where the data in column 'SimulationName' matches the name of the simulation under which this input model sits. If the file does NOT have a 'SimulationName' column then all data will be input.

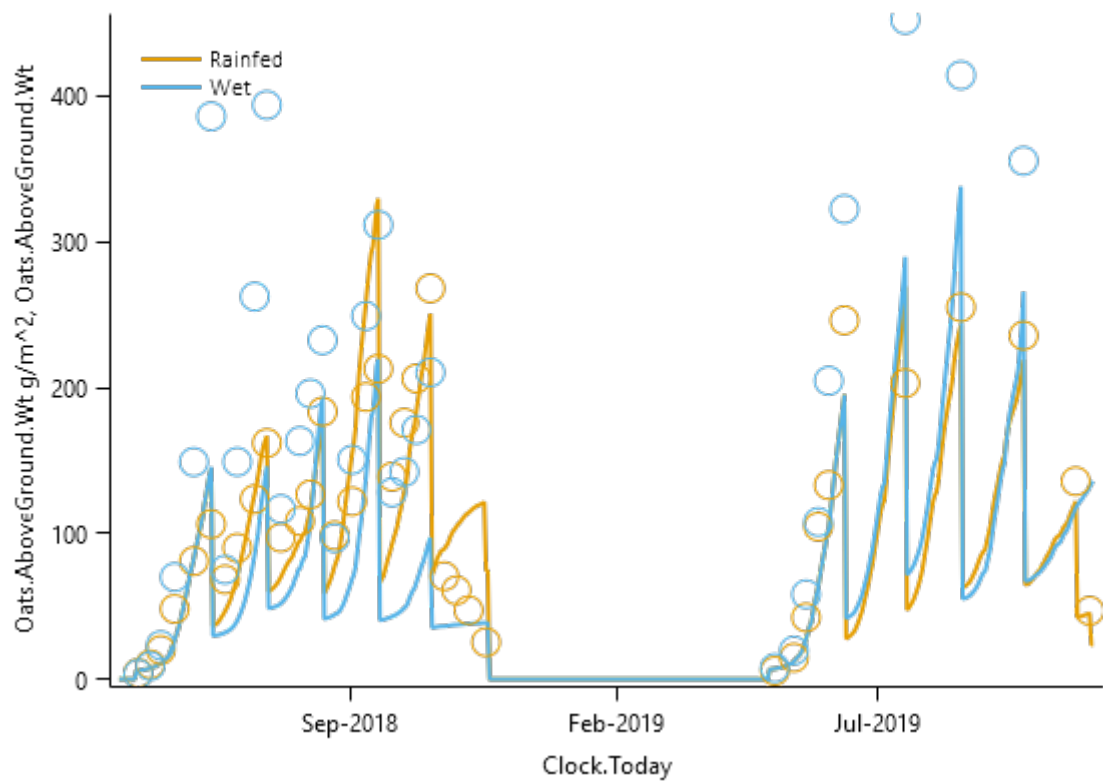
21.3 PredictedObservedSW

Reads the contents of a file (in apsim format) and stores into the DataStore. If the file has a column name of 'SimulationName' then this model will only input data for those rows where the data in column 'SimulationName' matches the name of the simulation under which this input model sits. If the file does NOT have a 'SimulationName' column then all data will be input.

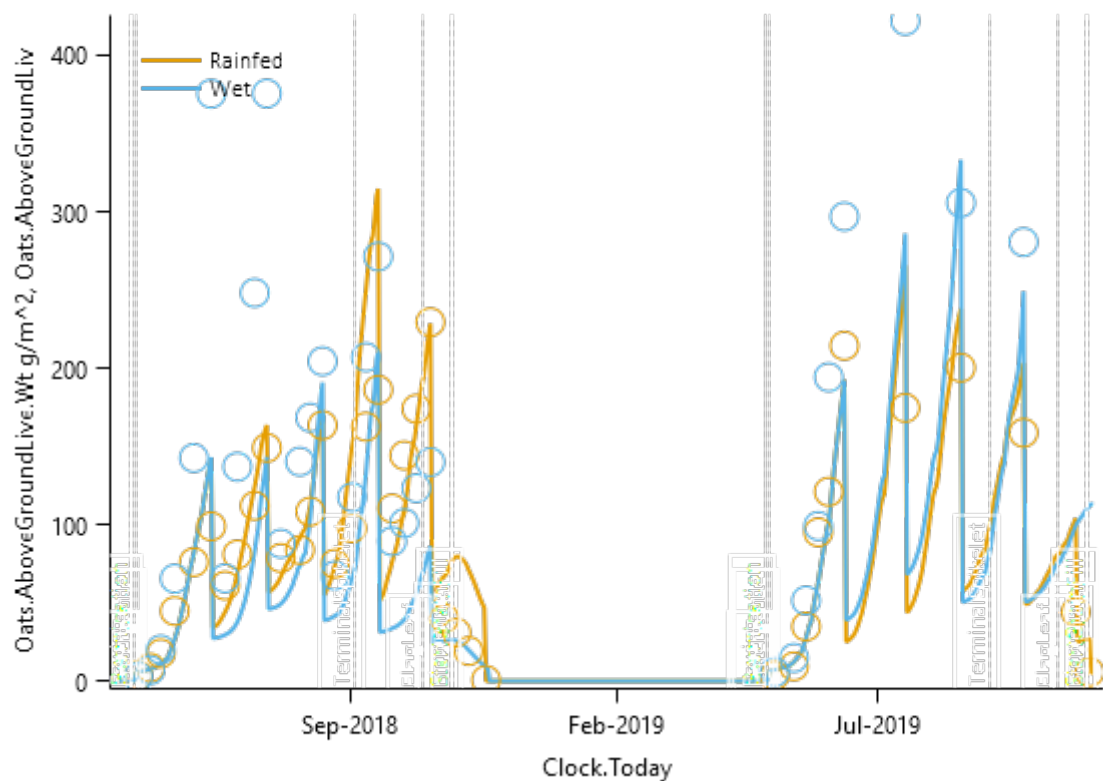
21.4 PredictedObservedH

Reads the contents of a file (in apsim format) and stores into the DataStore. If the file has a column name of 'SimulationName' then this model will only input data for those rows where the data in column 'SimulationName' matches the name of the simulation under which this input model sits. If the file does NOT have a 'SimulationName' column then all data will be input.

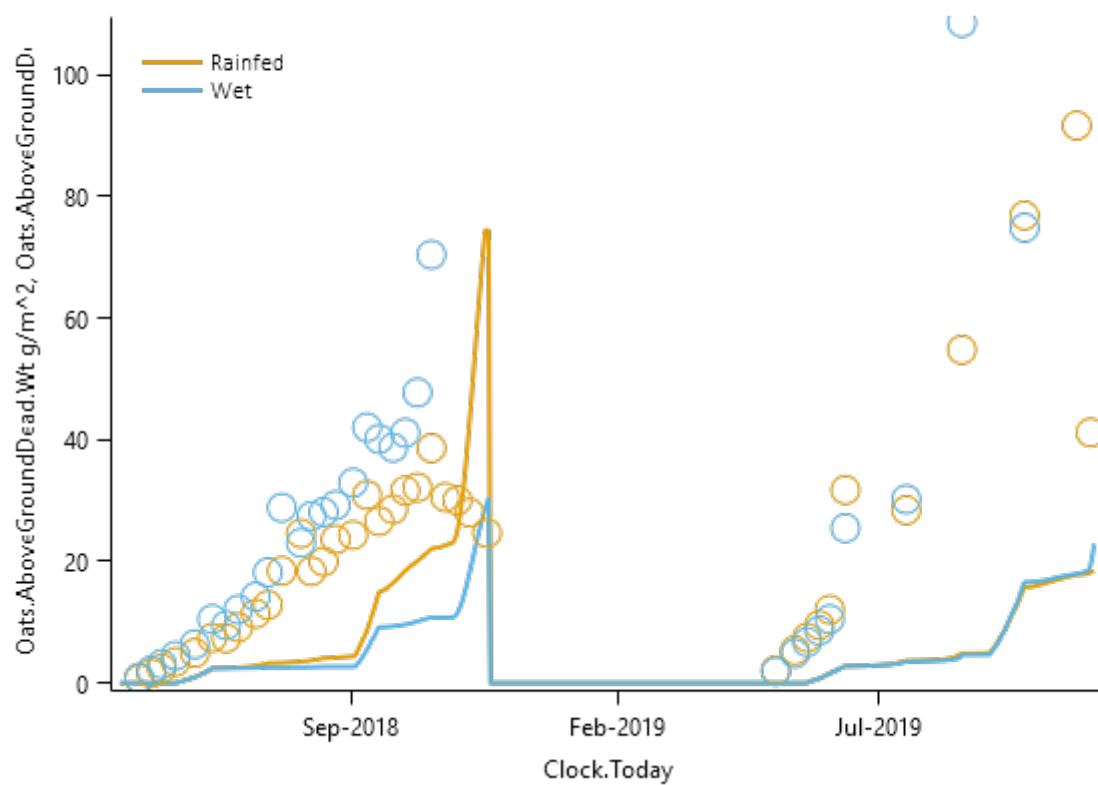
AboveGround



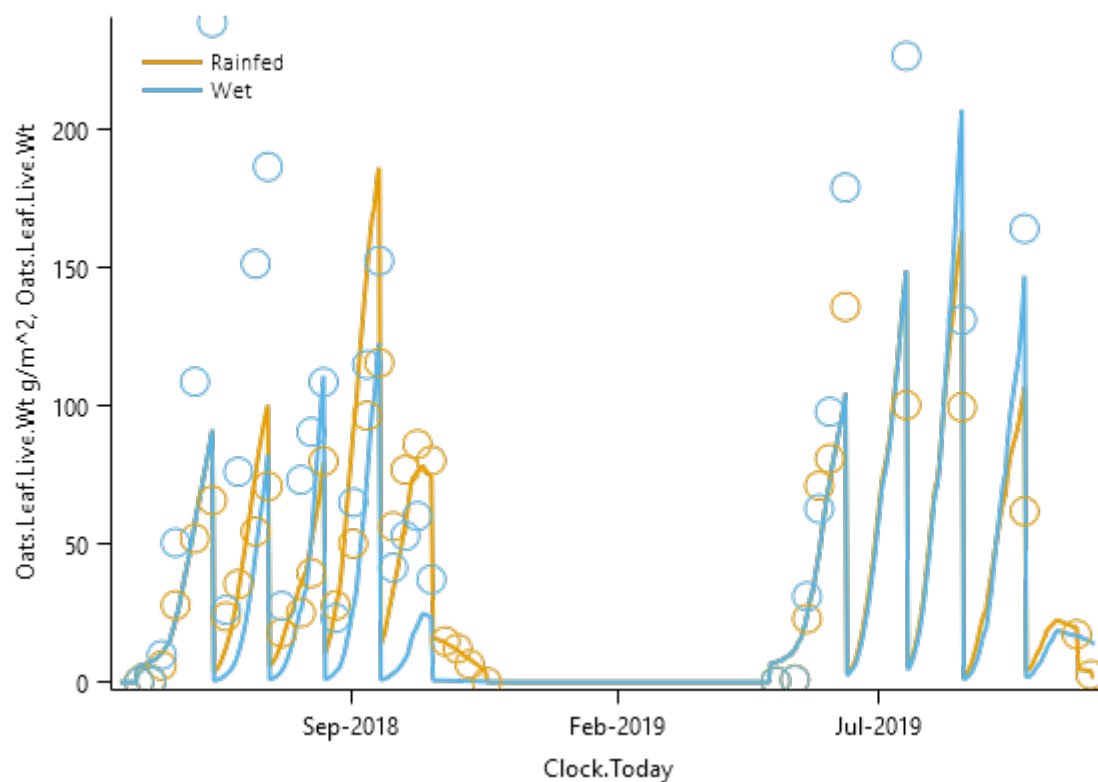
AboveGroundLive



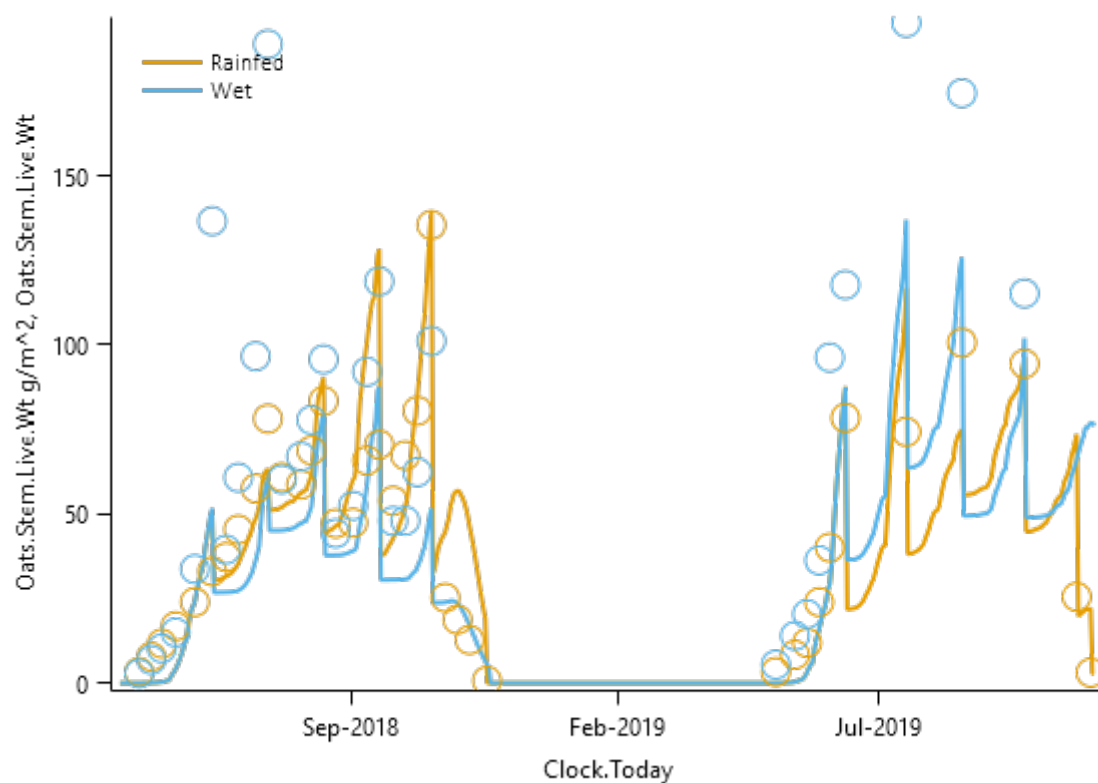
AboveGroundDead



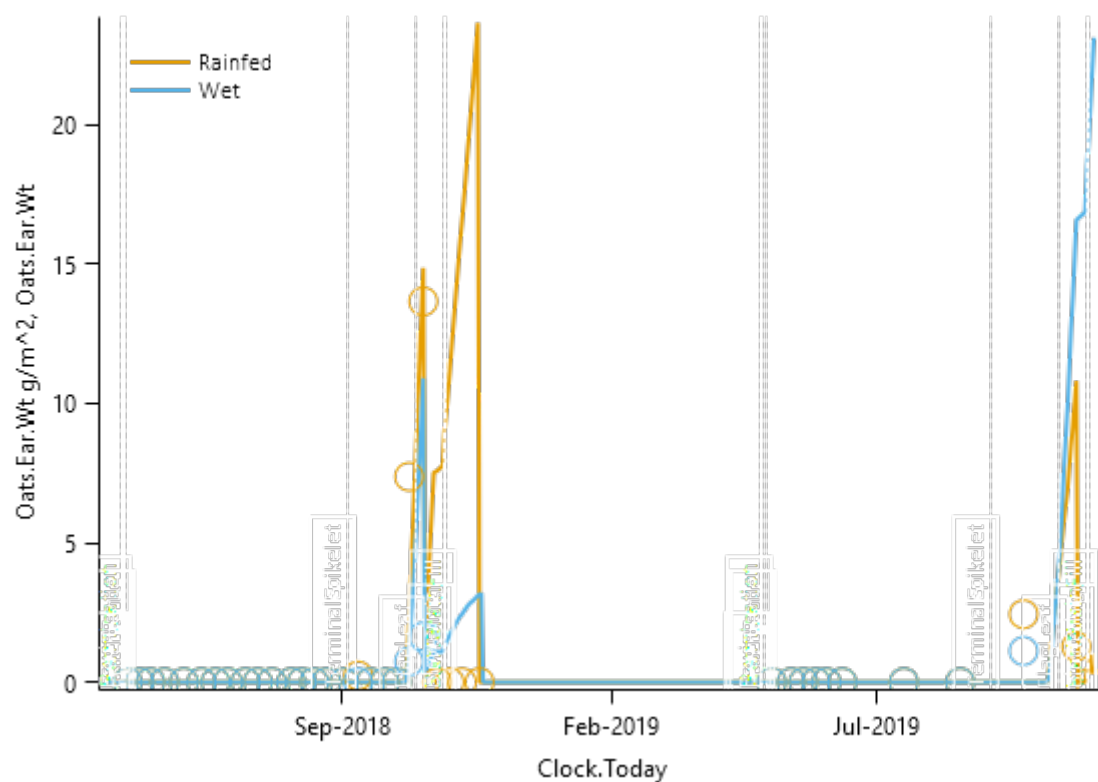
LeafLive



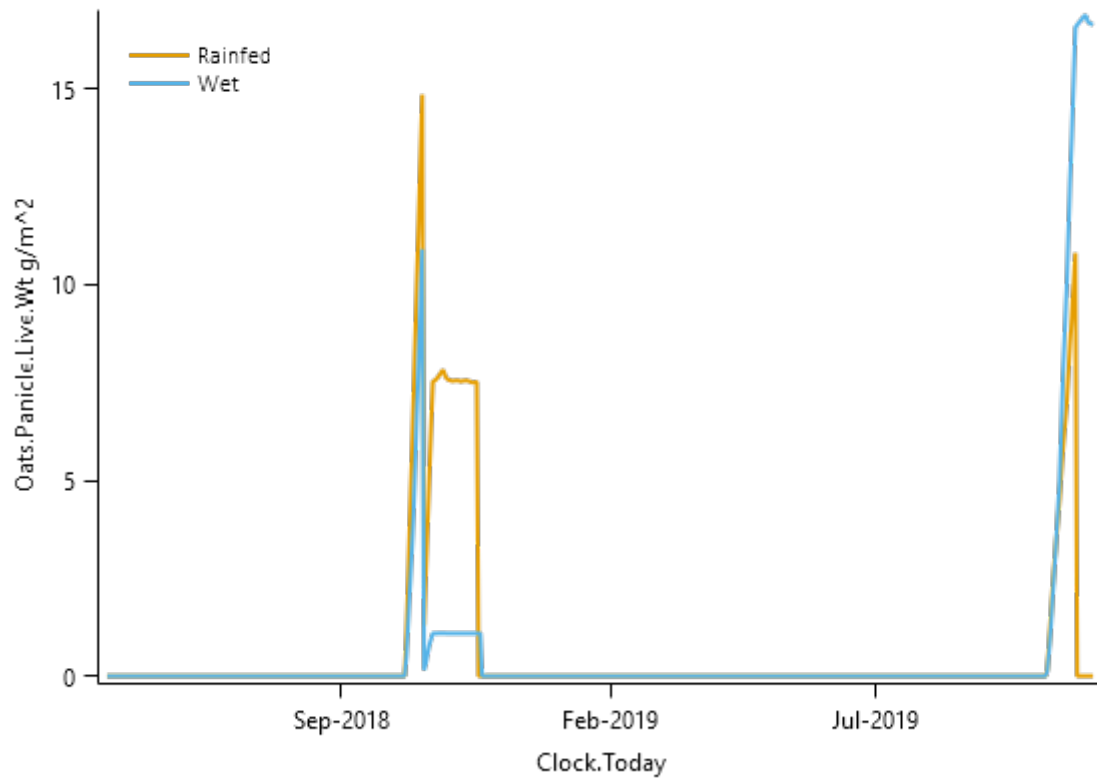
StemLive



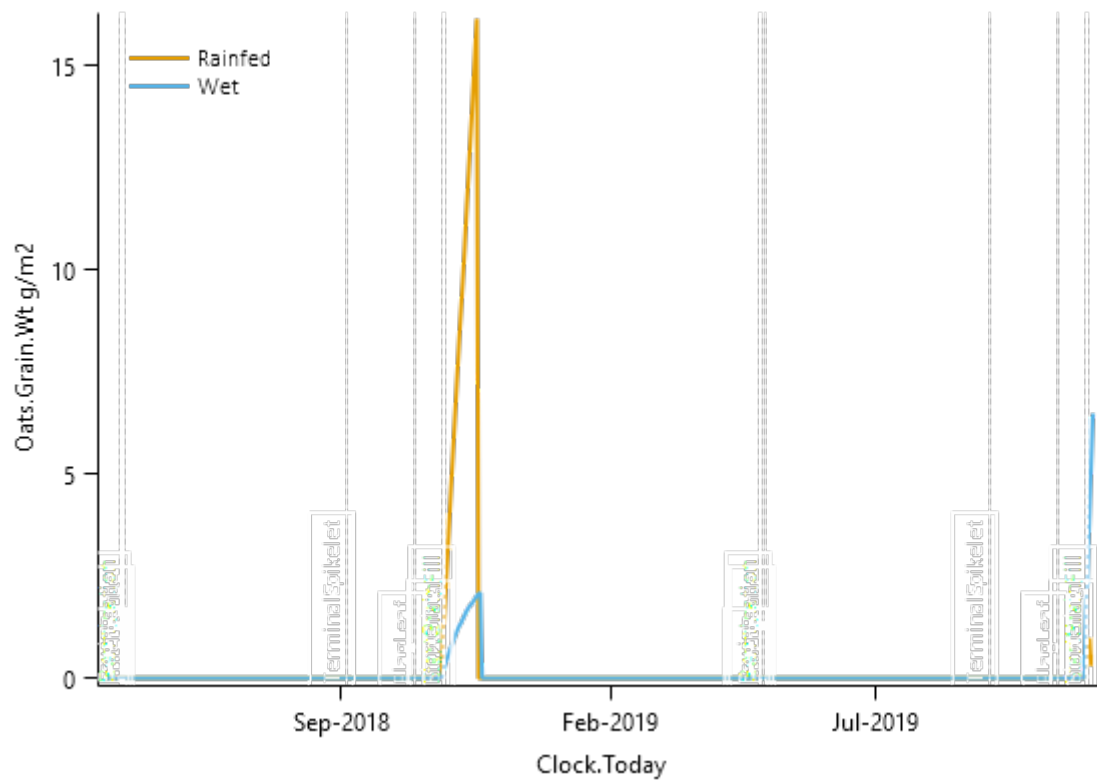
Ear



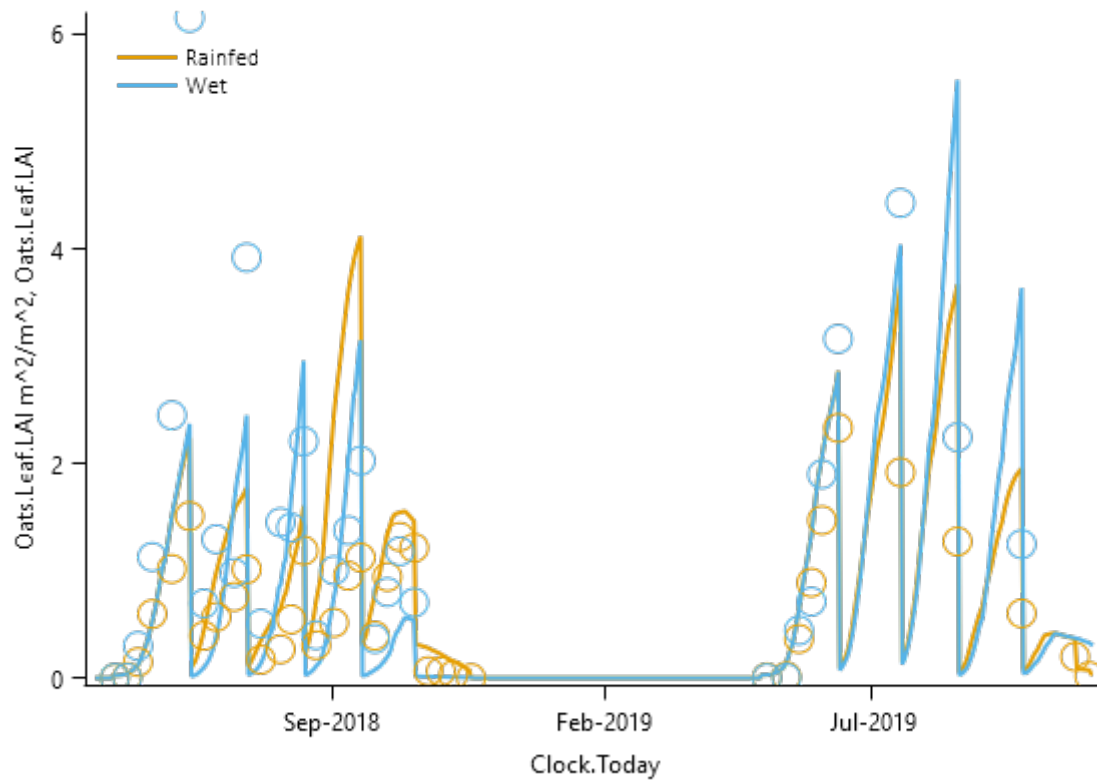
PanicleLive



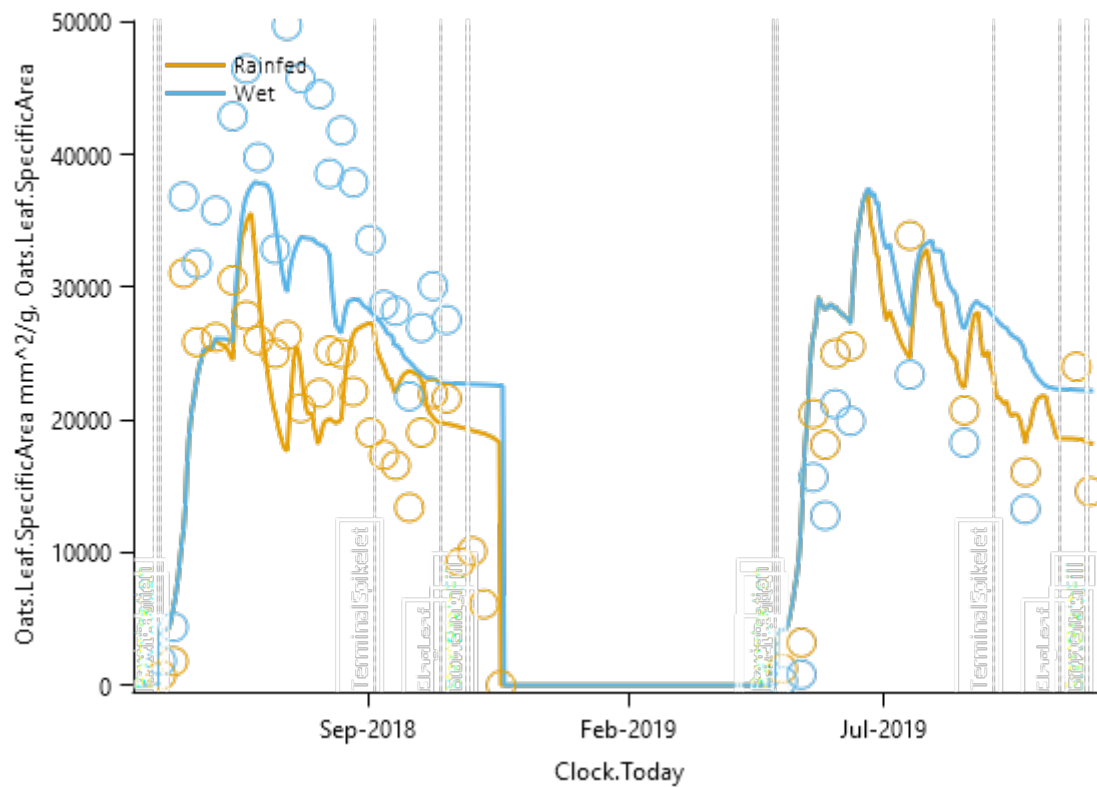
Grain



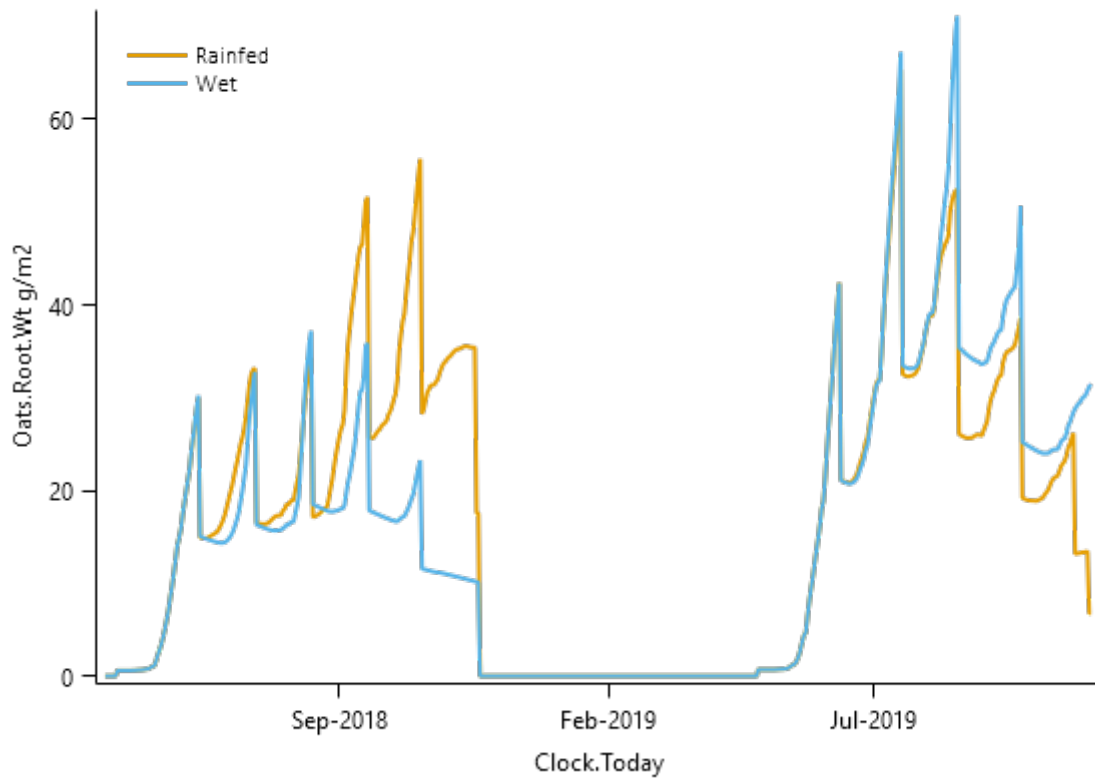
LeafLAI



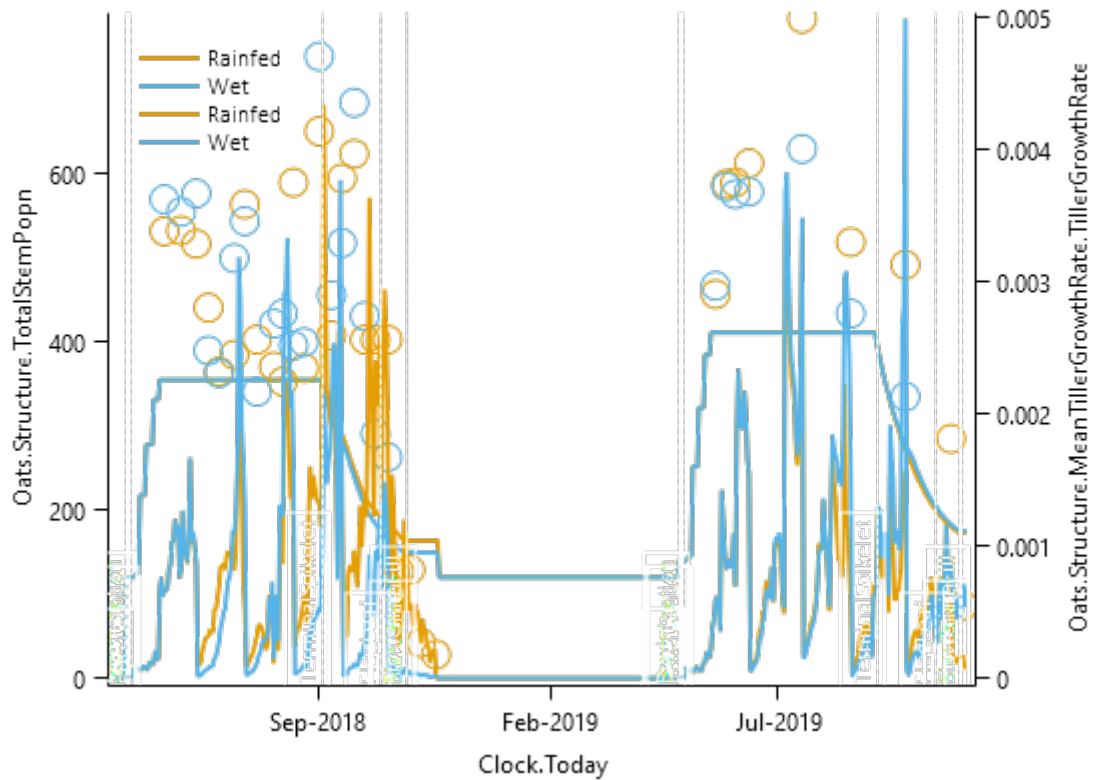
LeafSLA



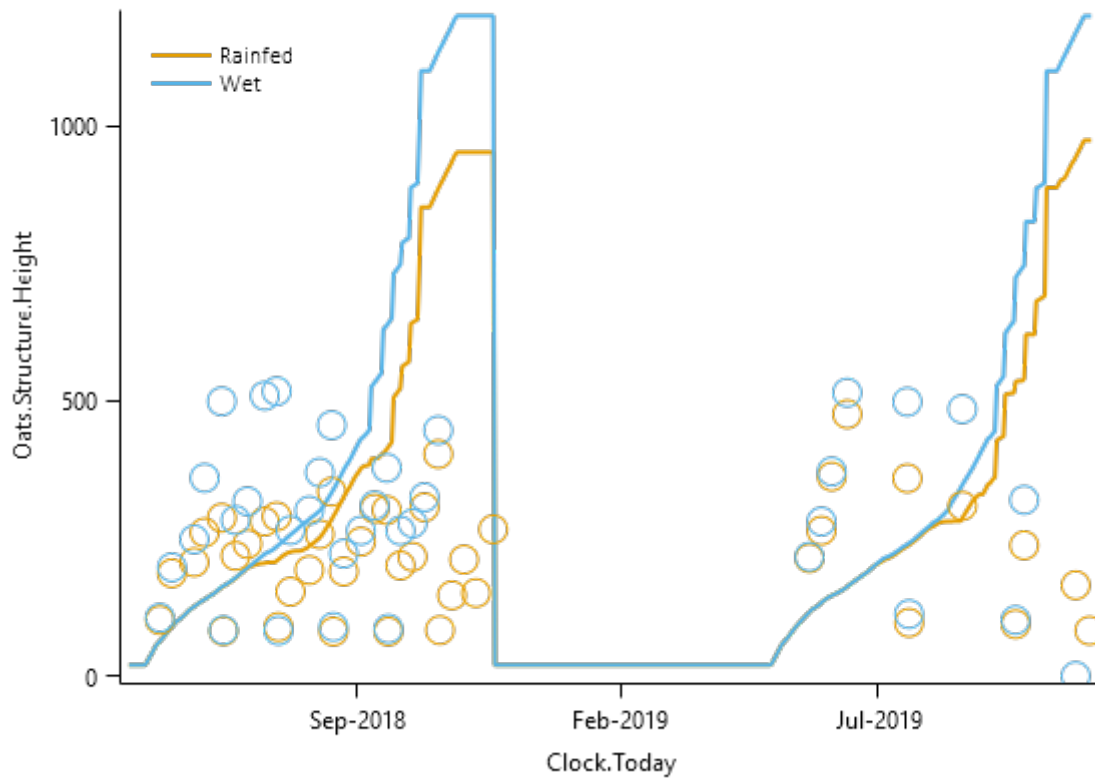
RootLive



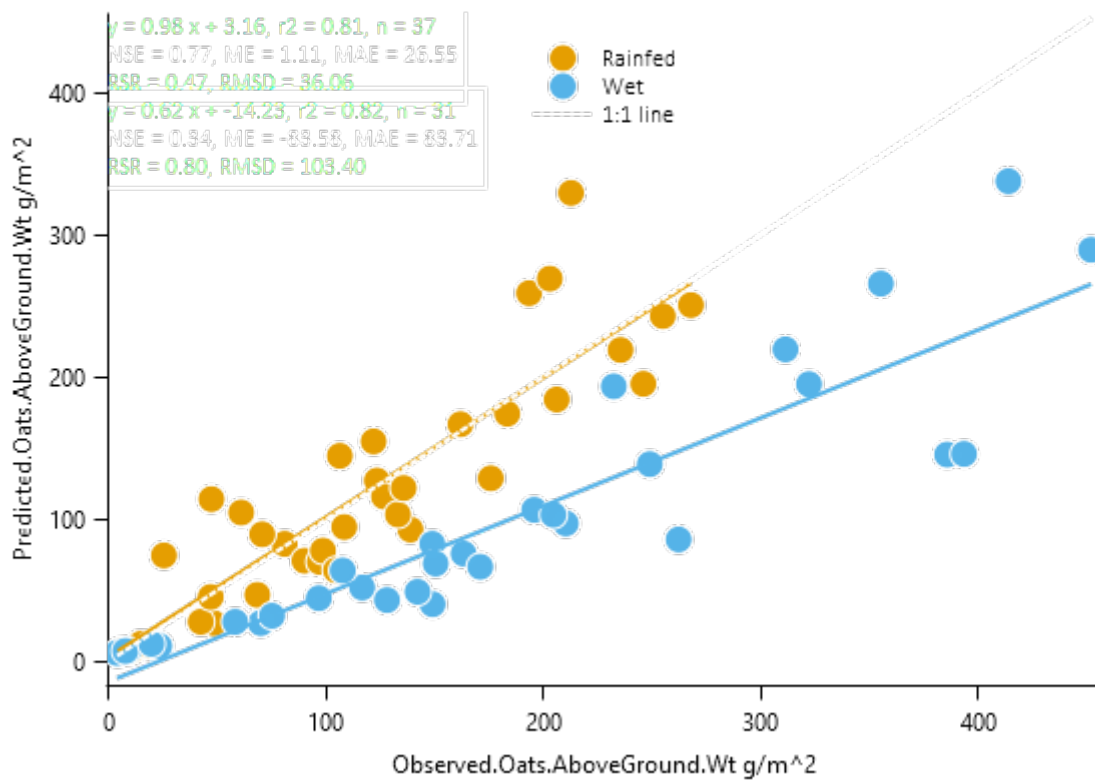
StemPopn



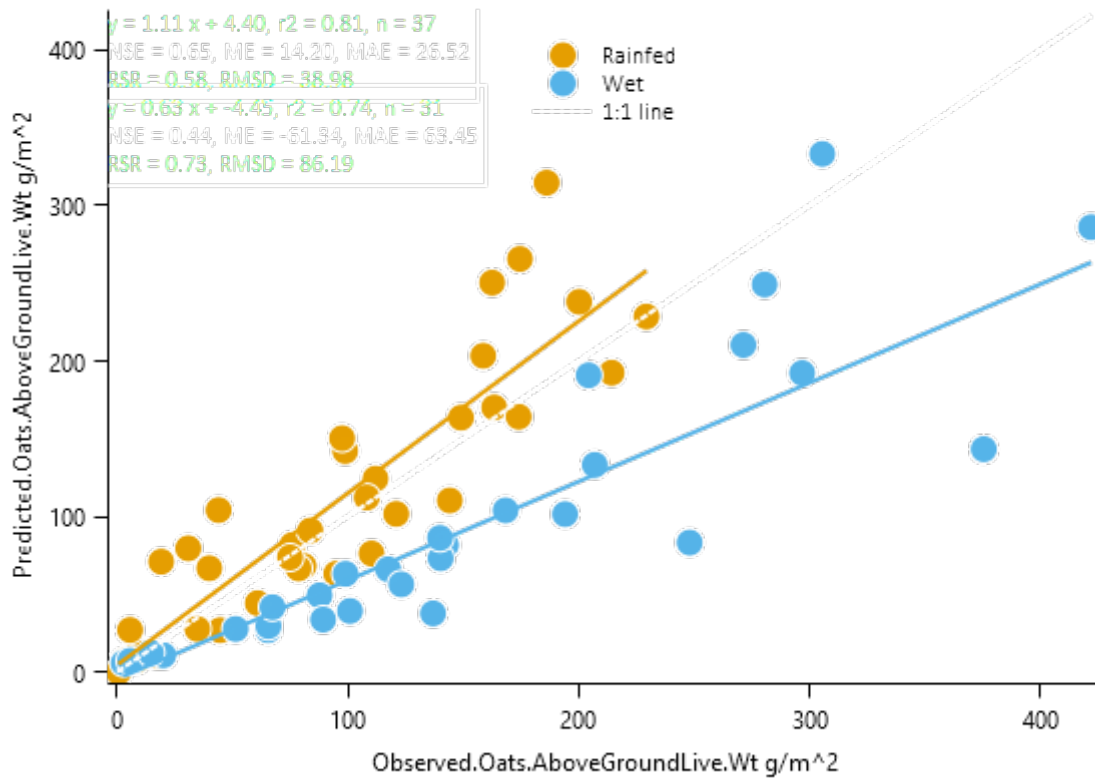
Height



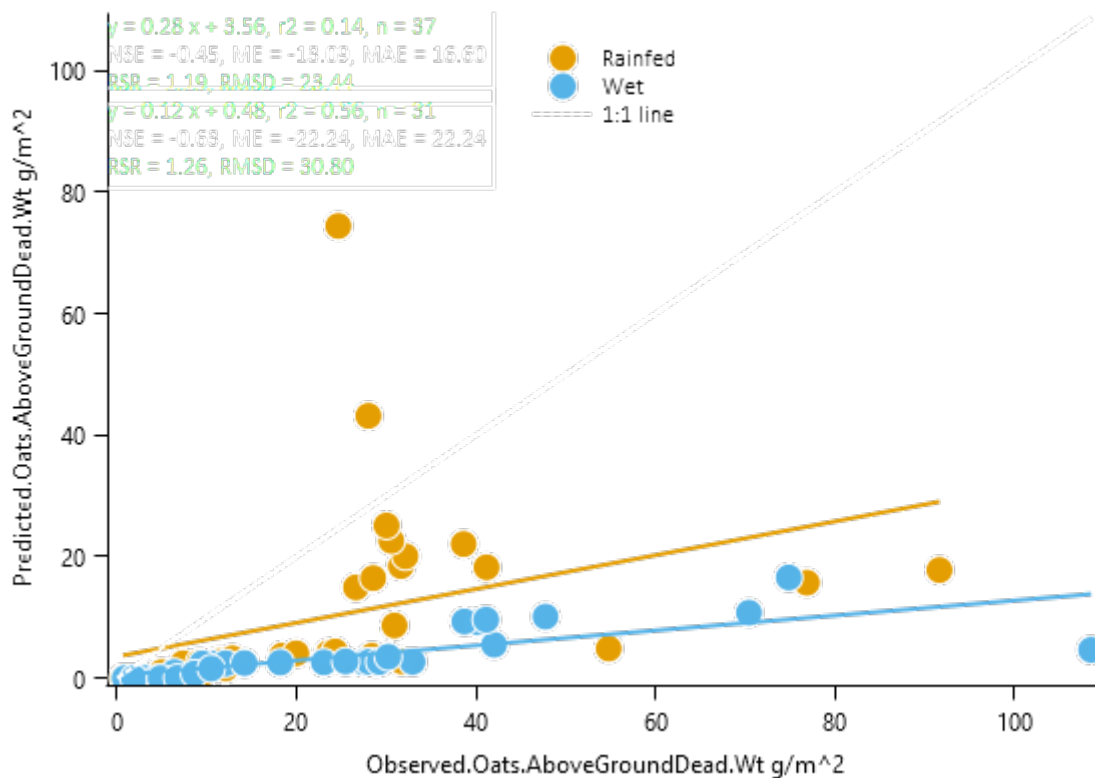
RegressionAboveGround



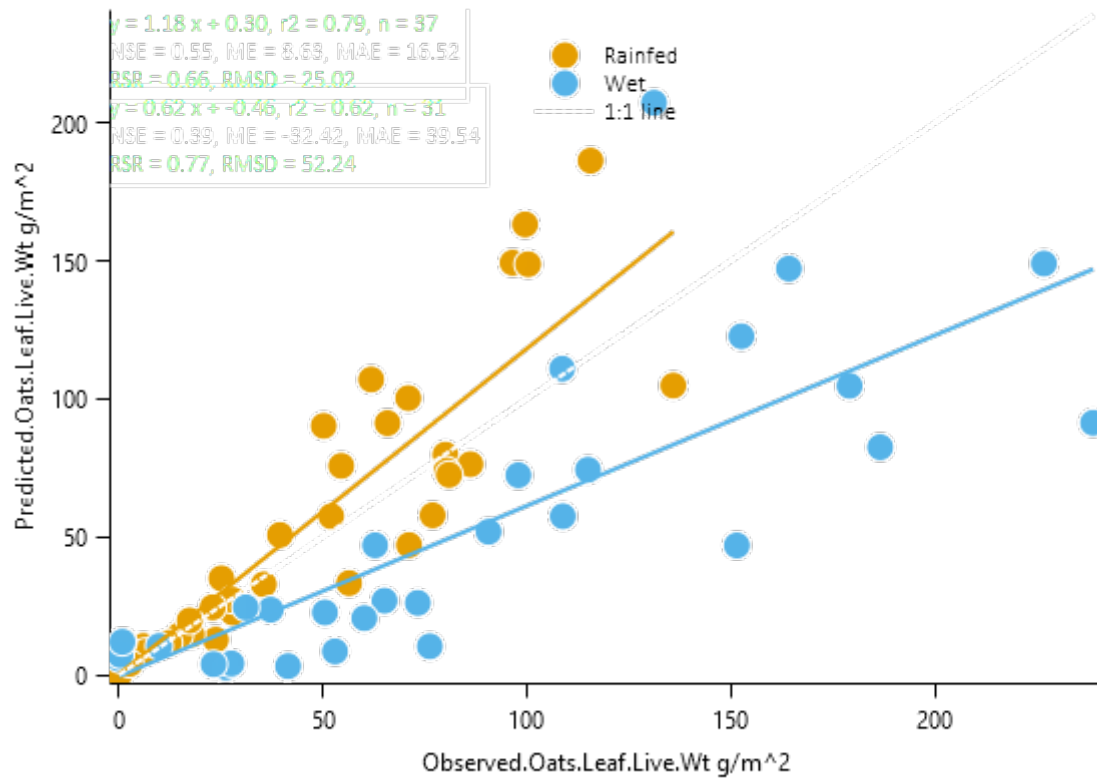
RegressionAboveGroundLive



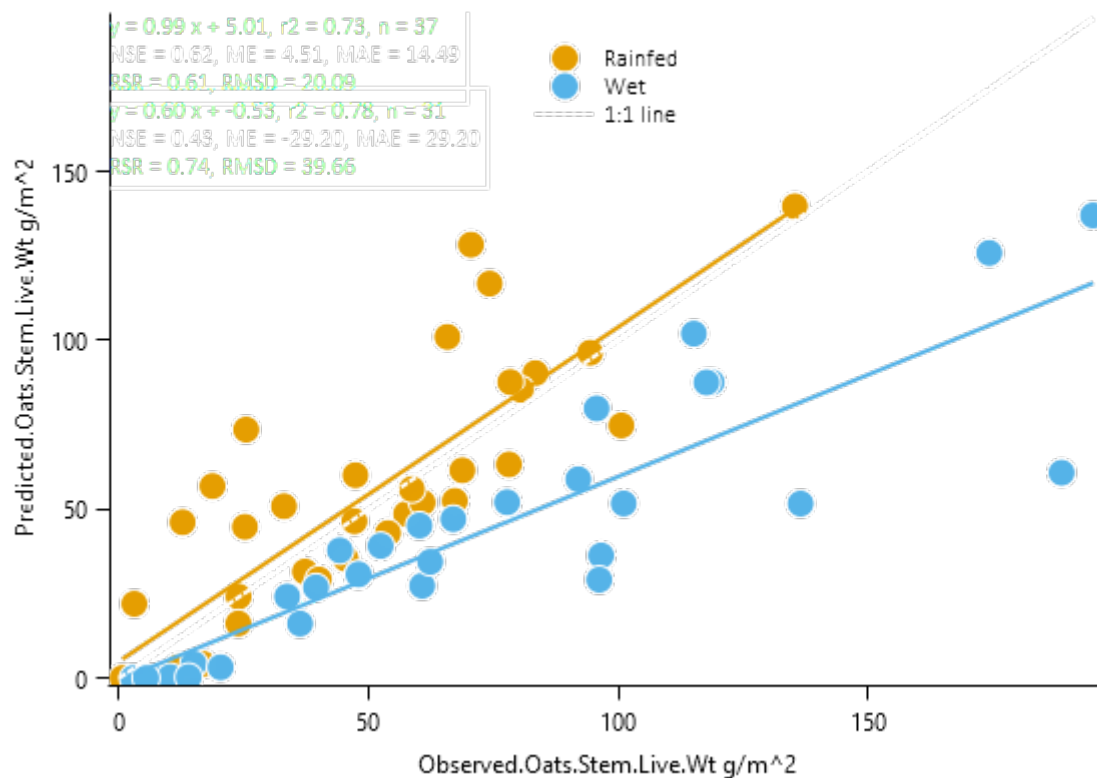
RegressionAboveGroundDead



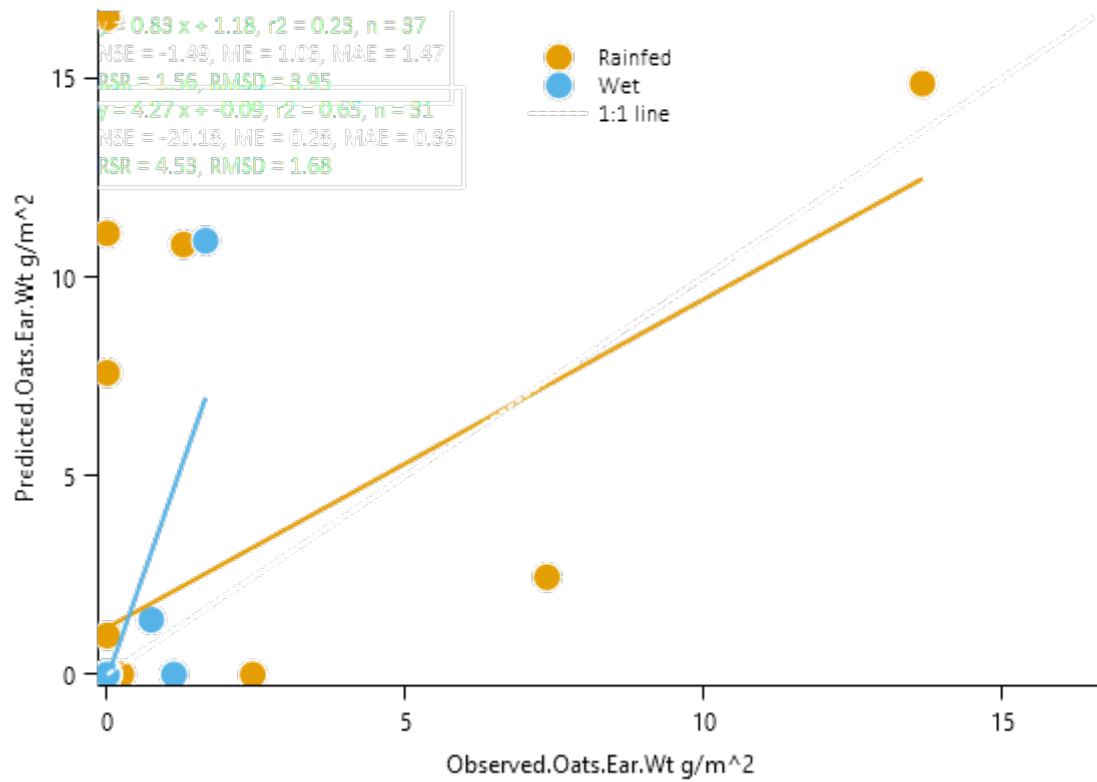
RegressionLeafLive



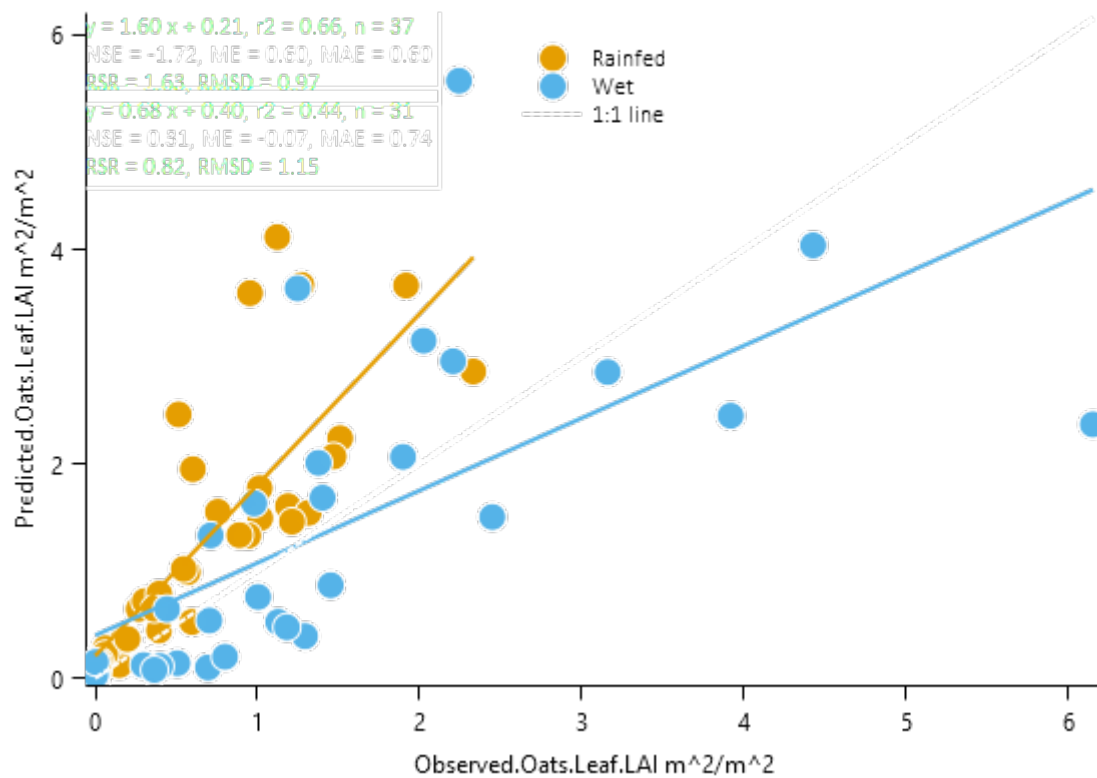
RegressionStemLive



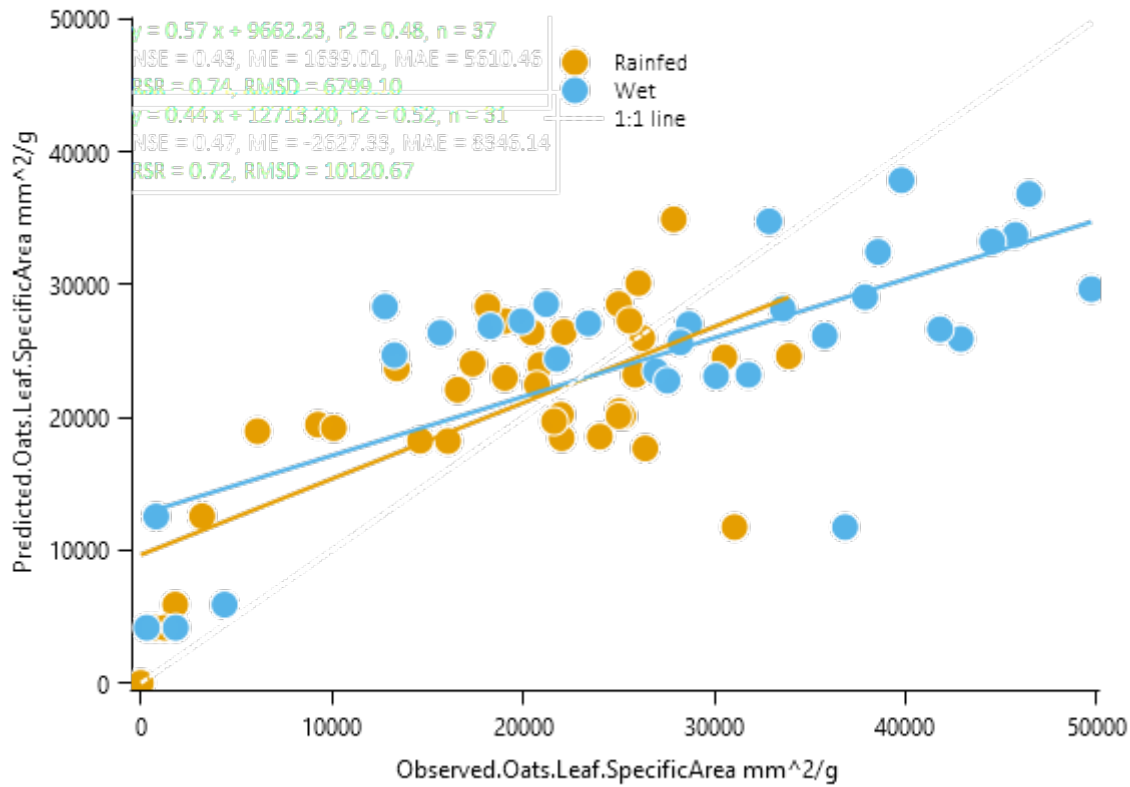
RegressionEar



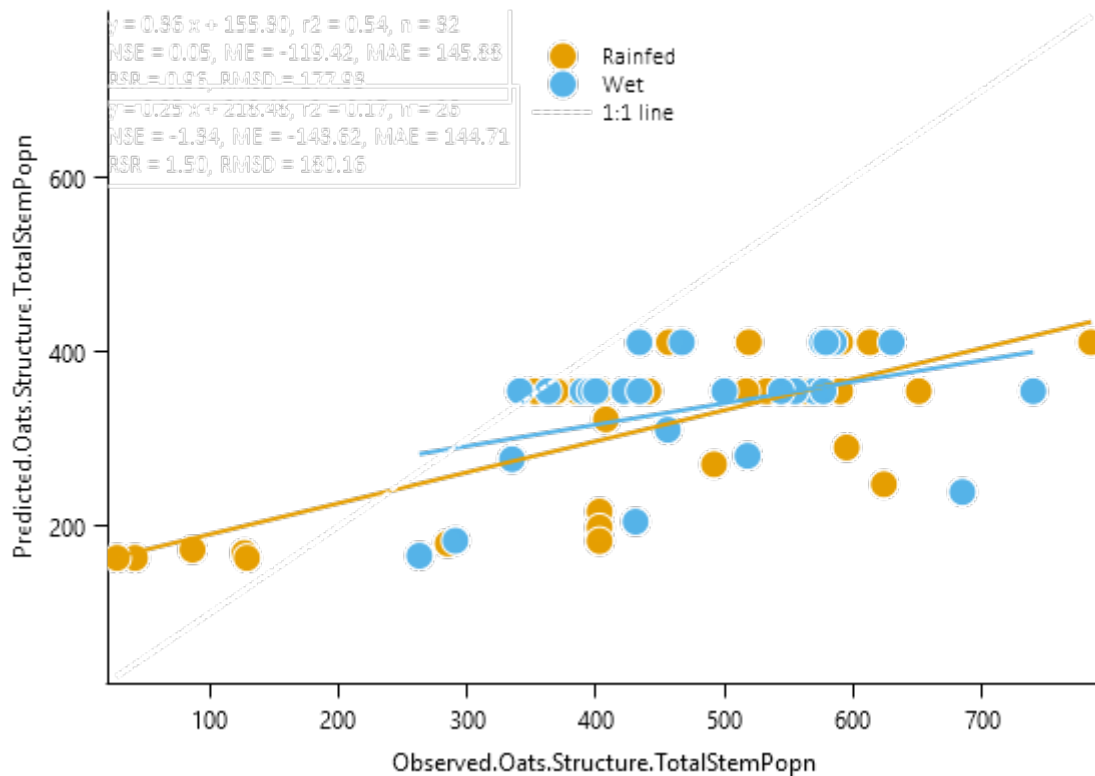
RegressionLeafLAI



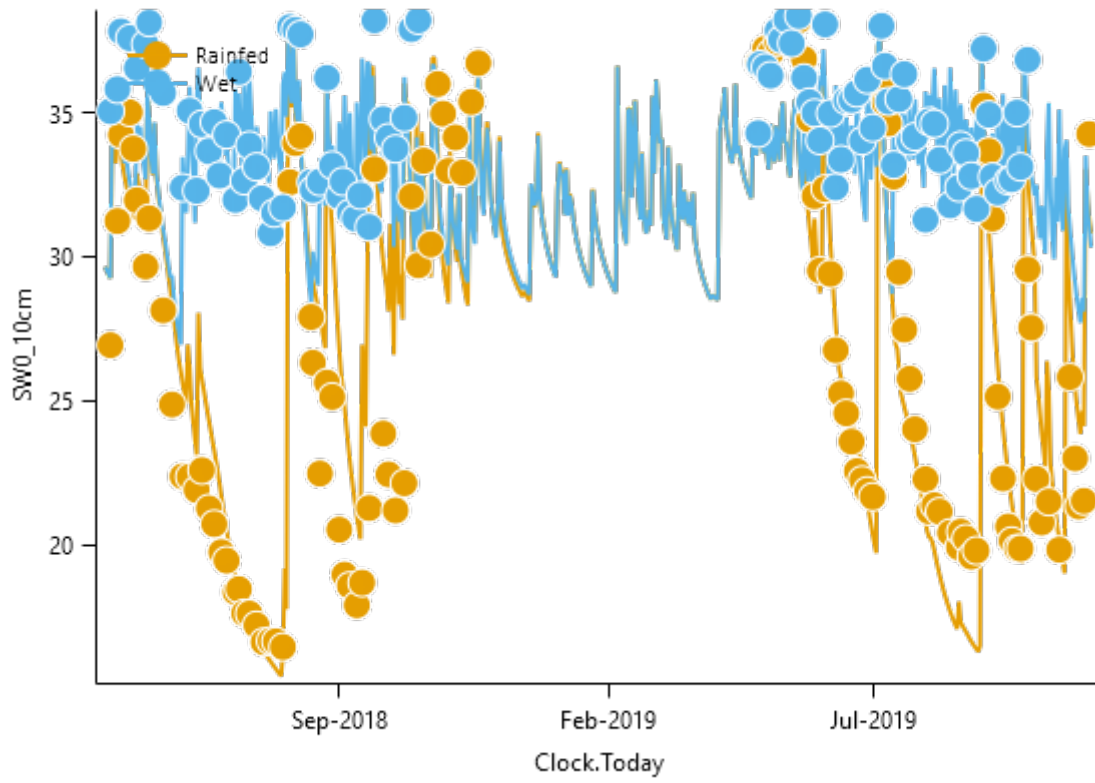
RegressionLeafSLA



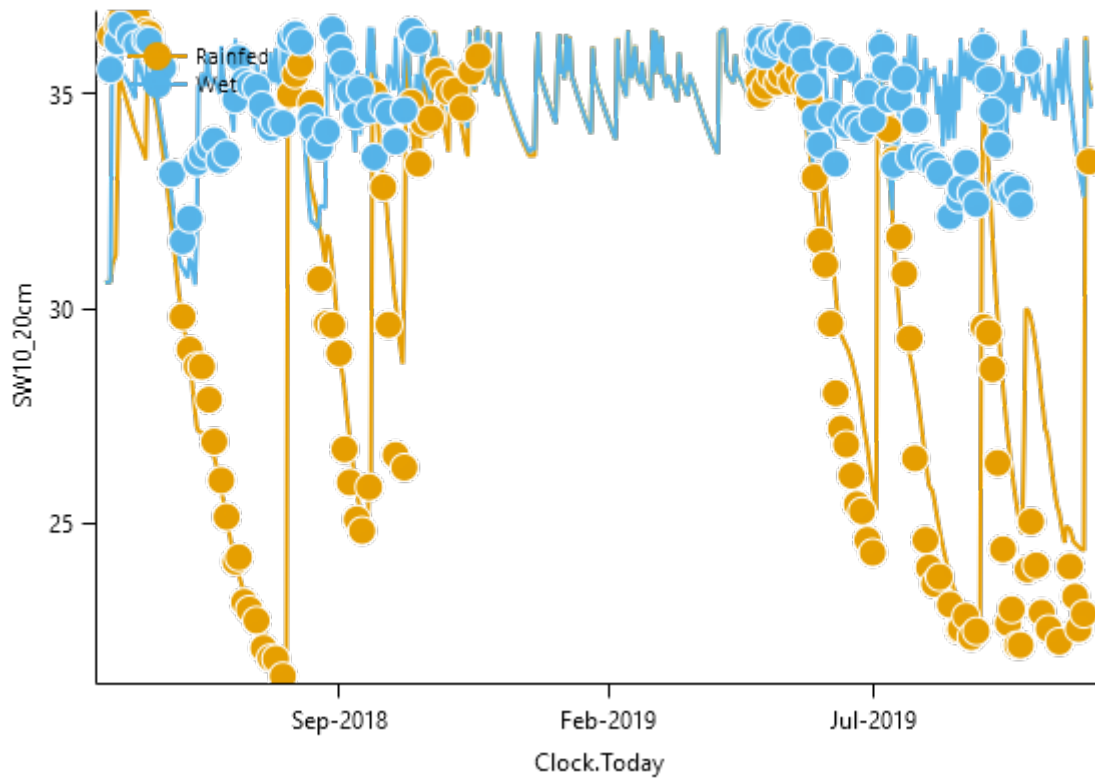
RegressionStempopulation



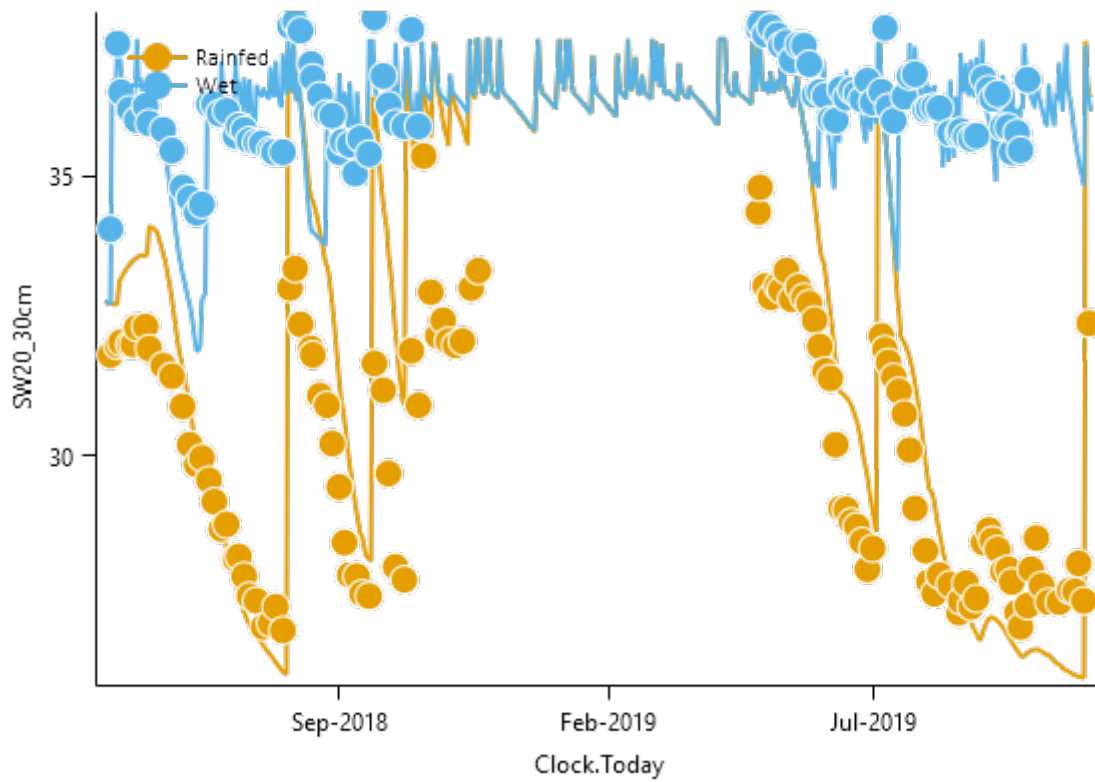
SW 10cm



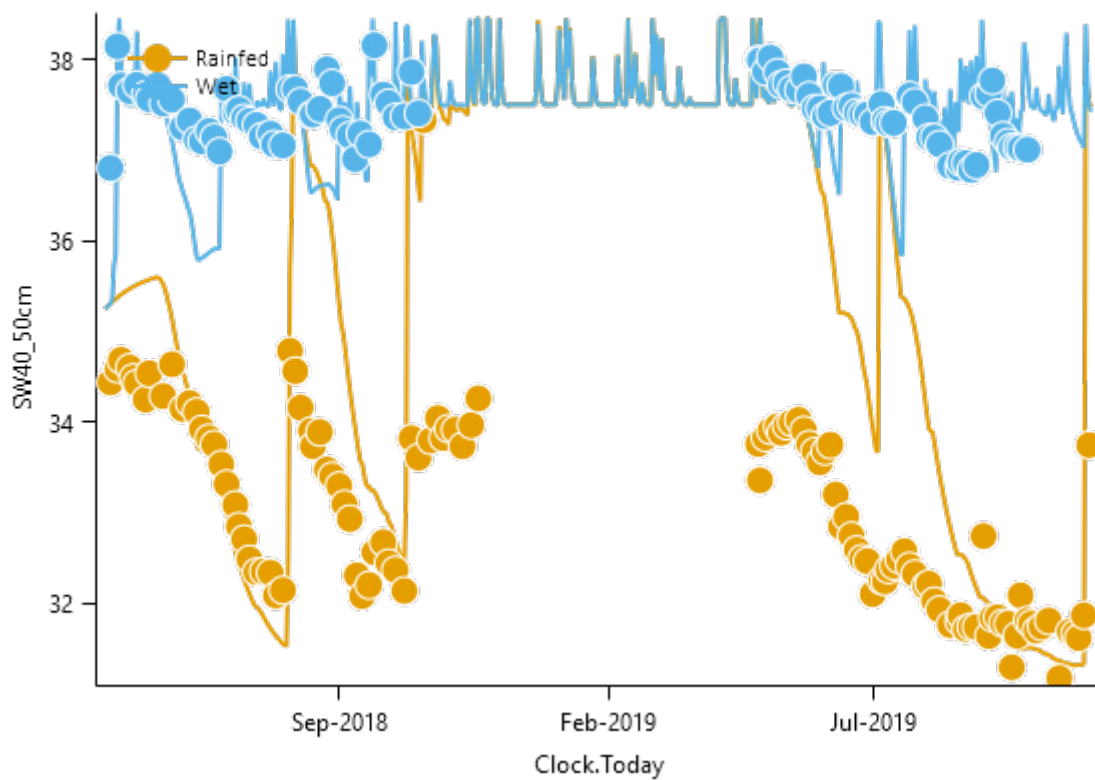
SW 20cm



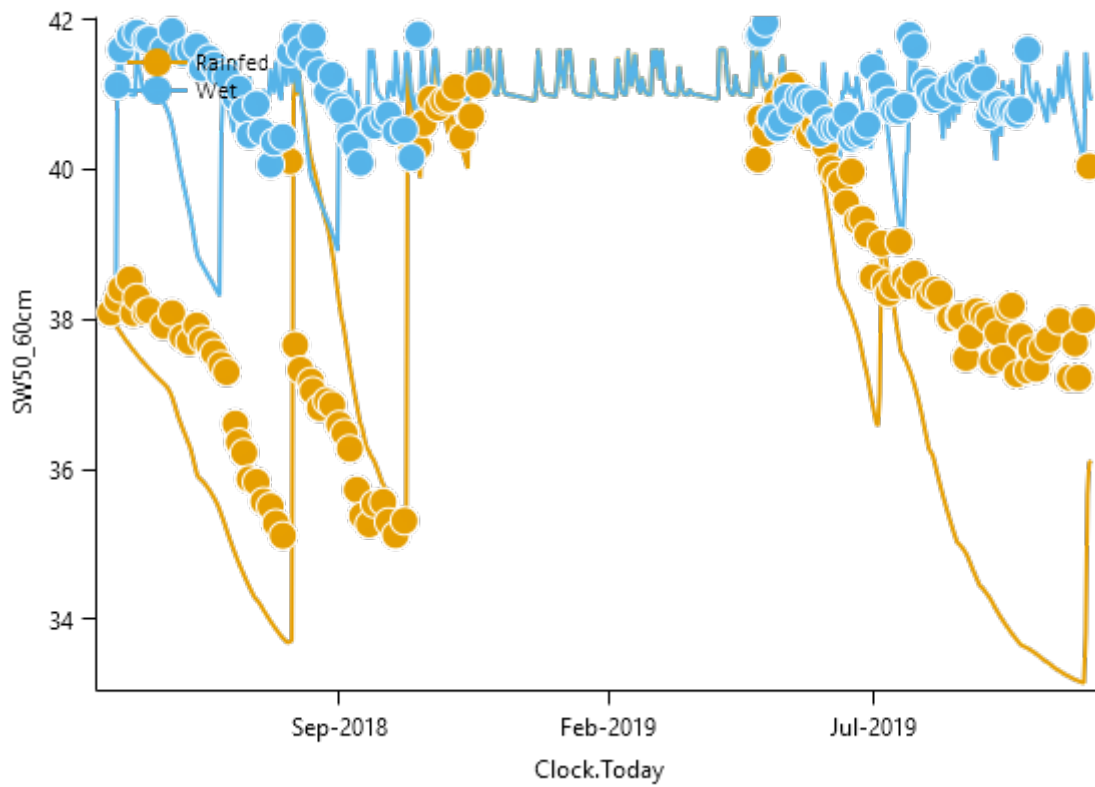
SW 30cm



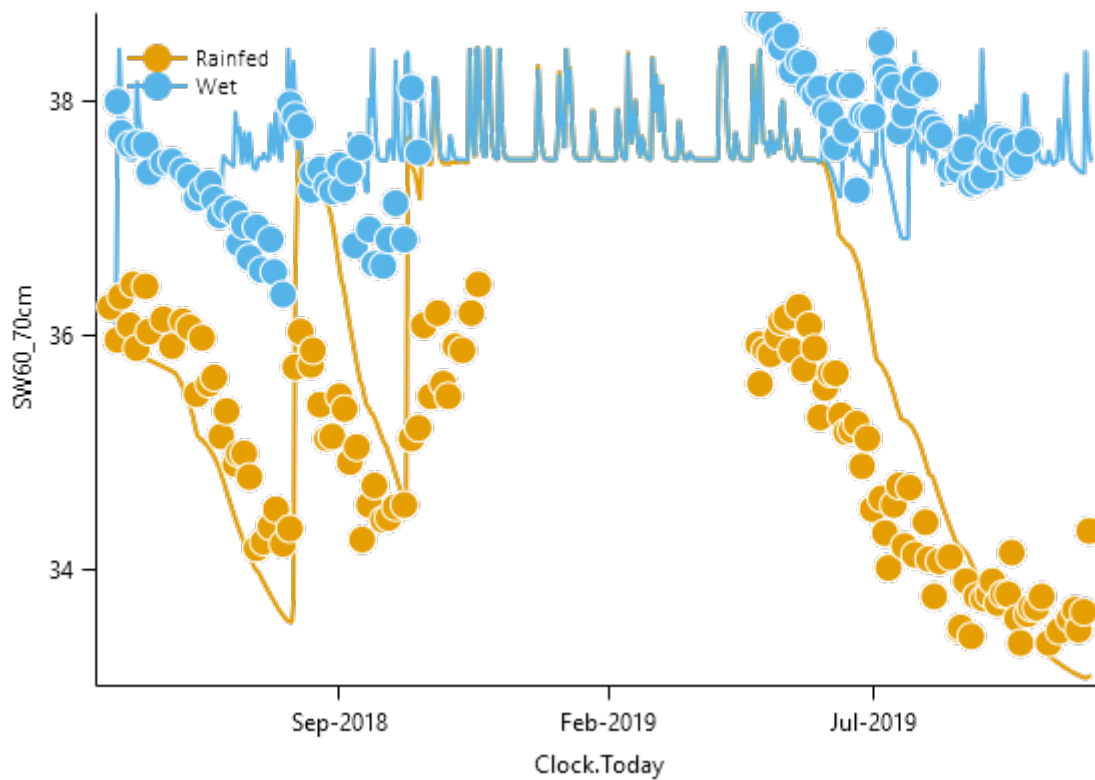
SW 50cm



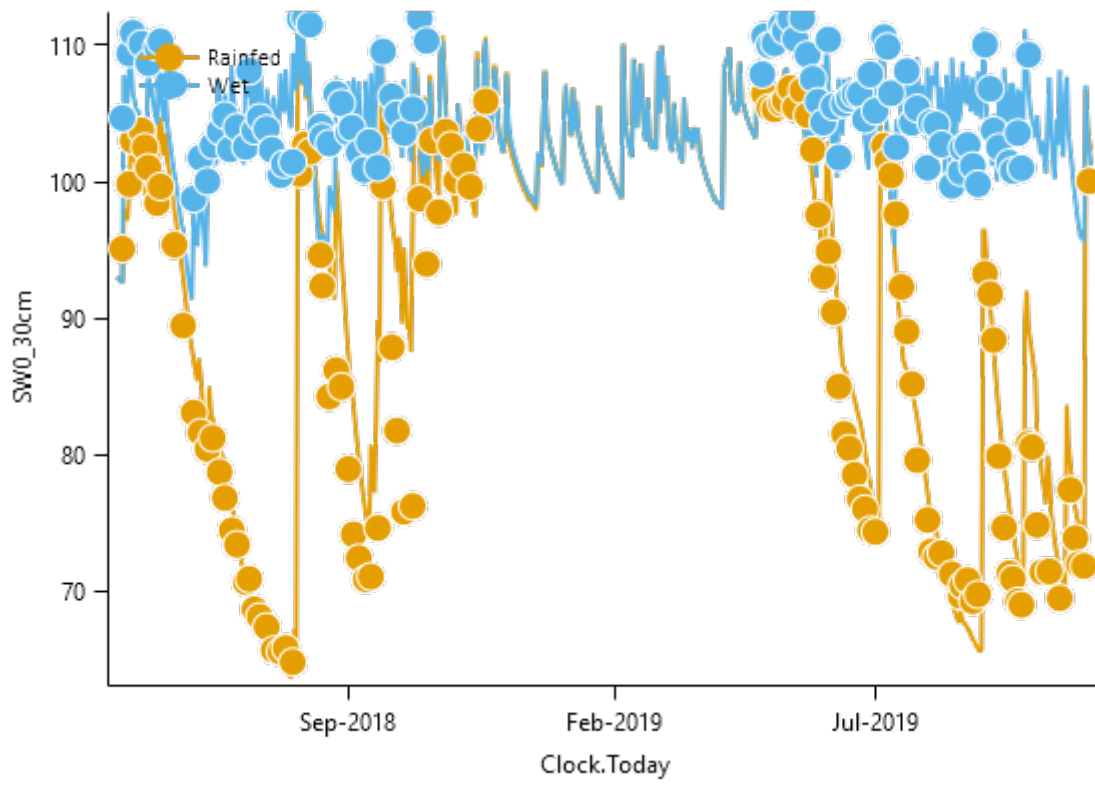
SW 60cm



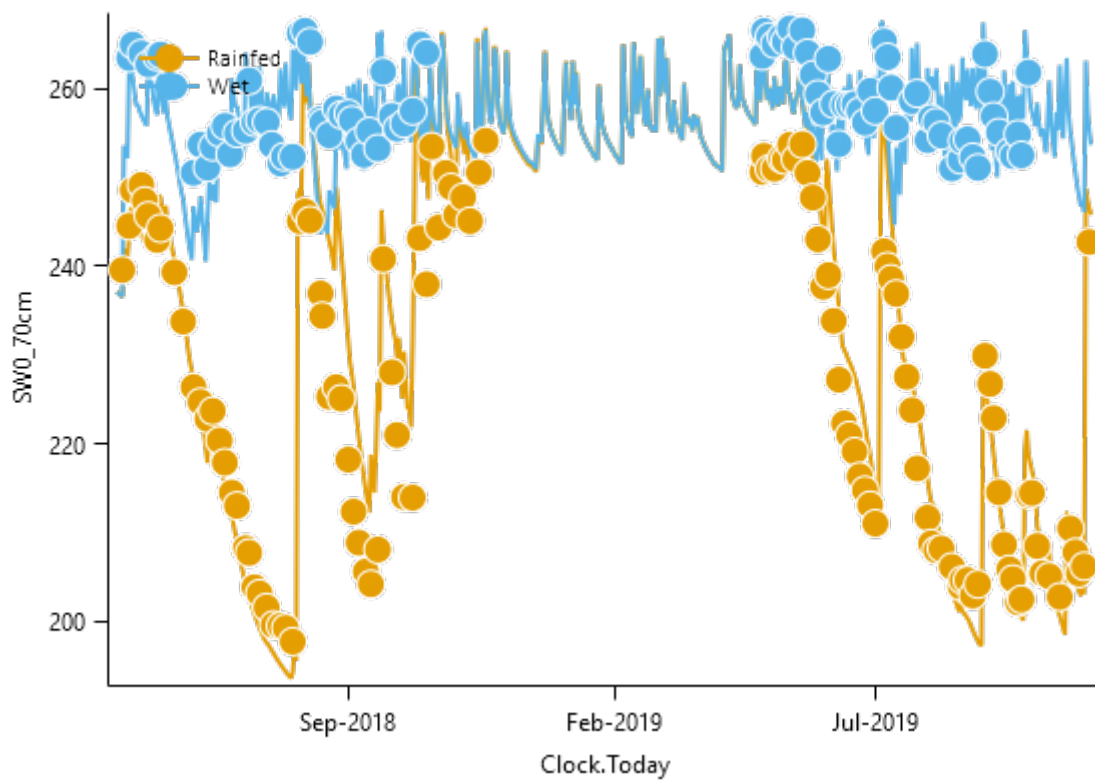
SW 70cm



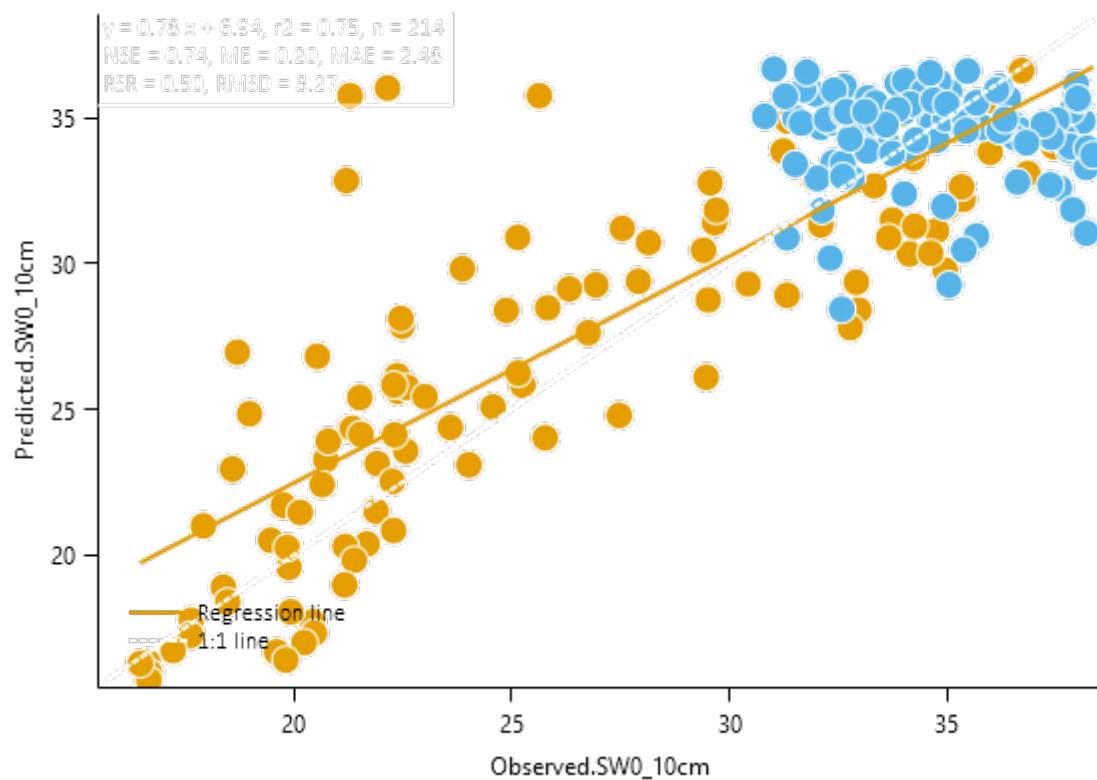
SW 0_30cm



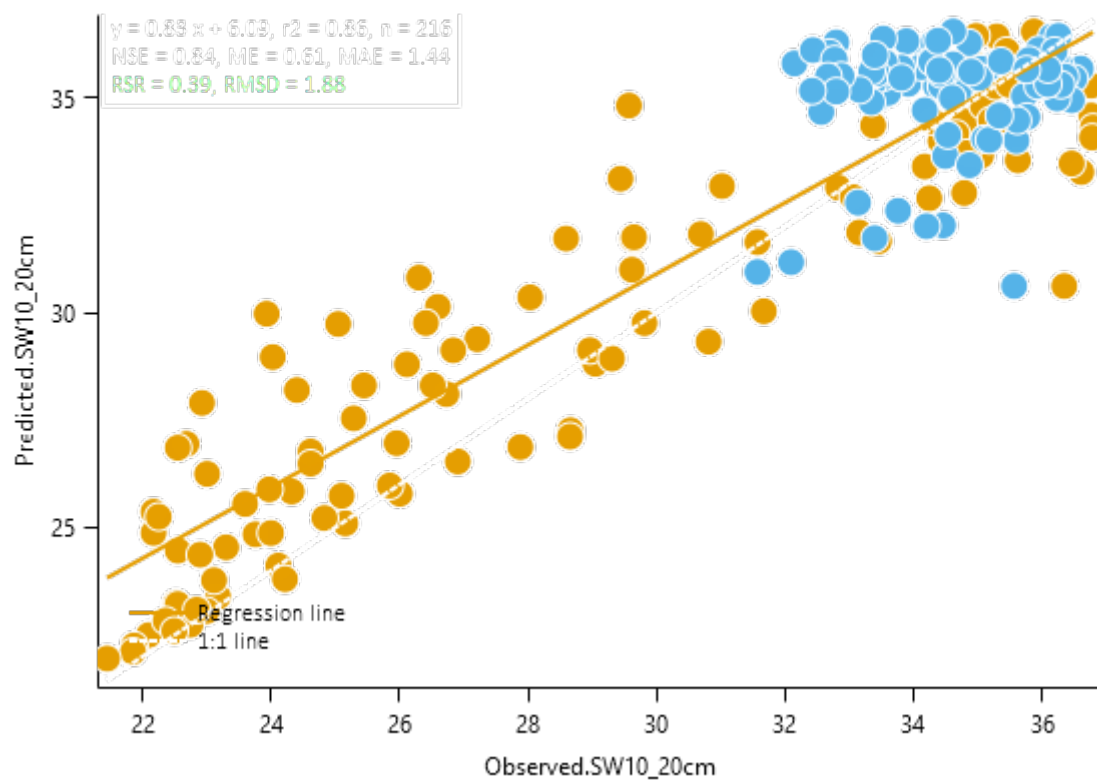
SW 0_70cm



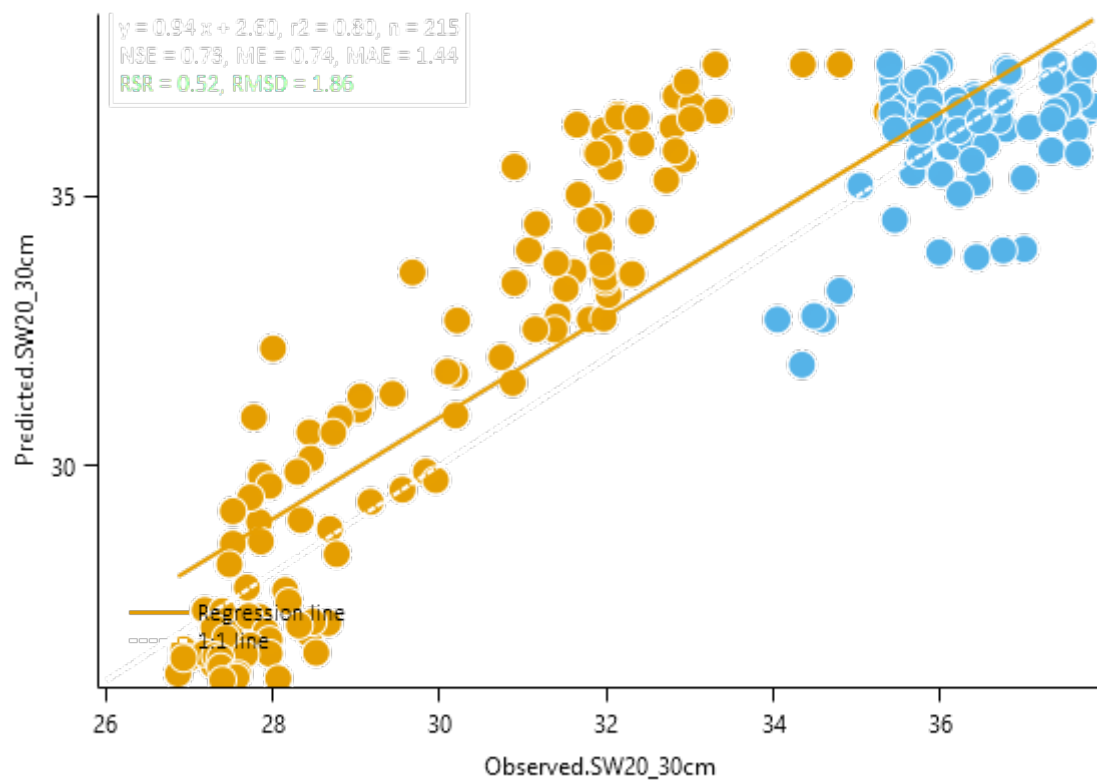
Regression0_10cm



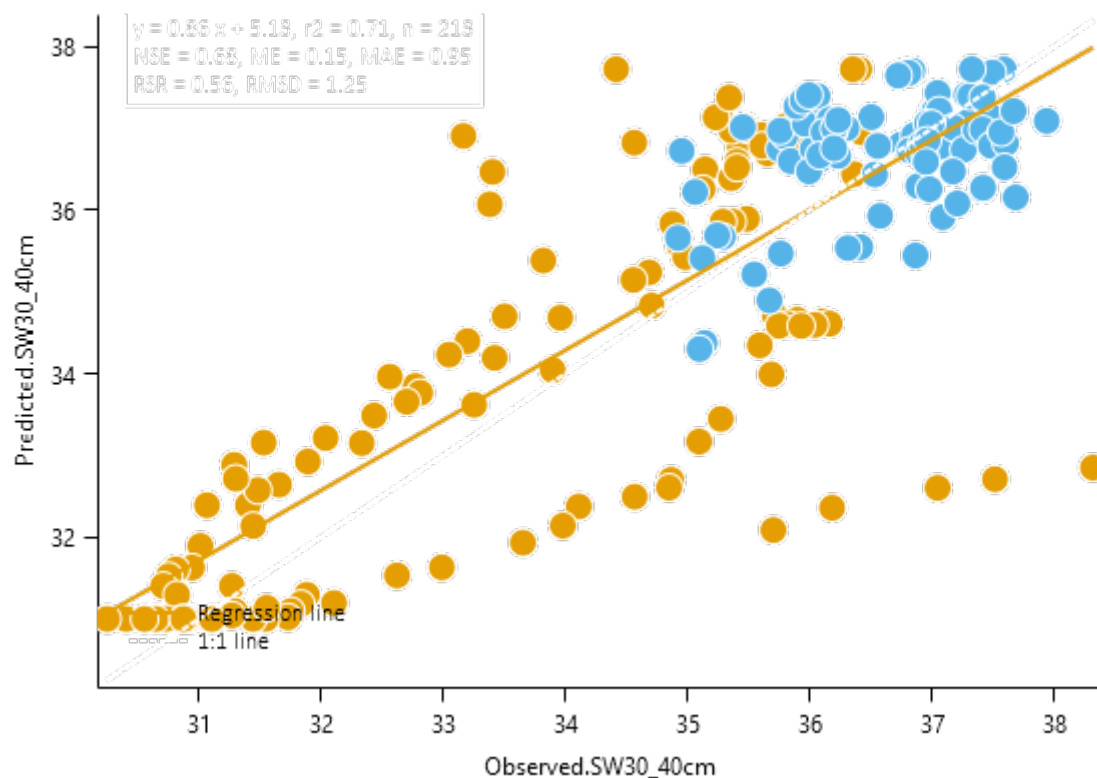
Regression10_20cm



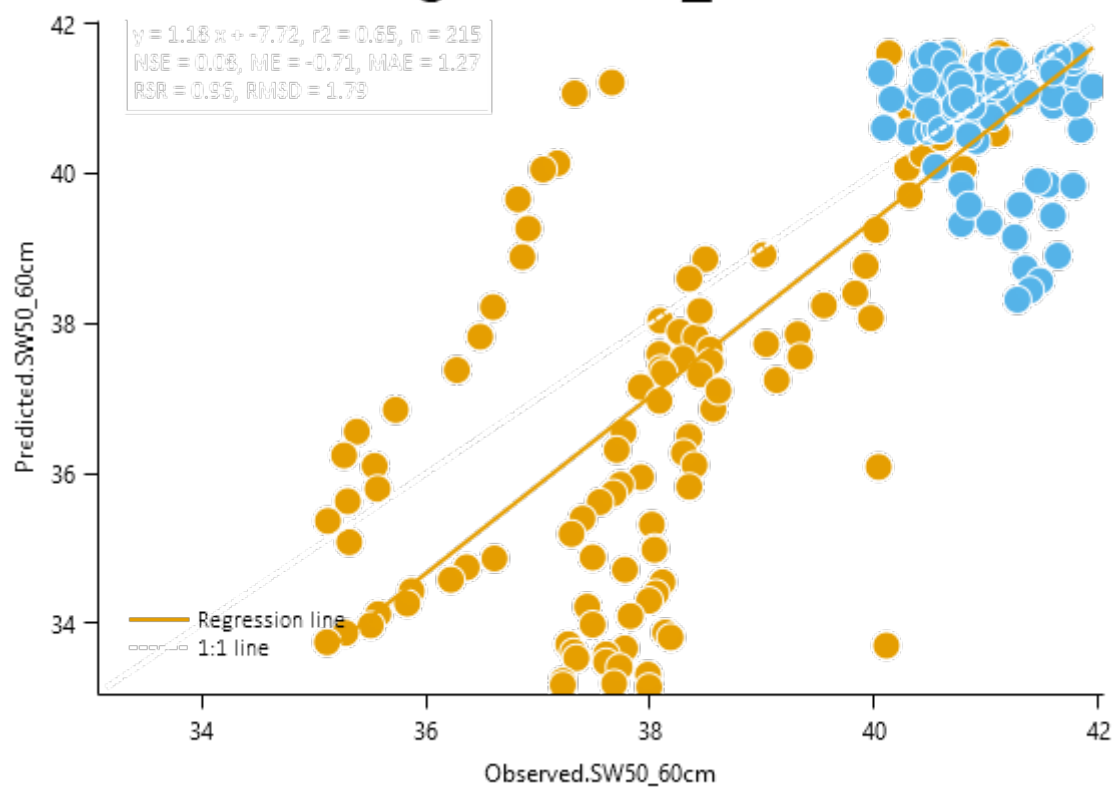
Regression20_30cm



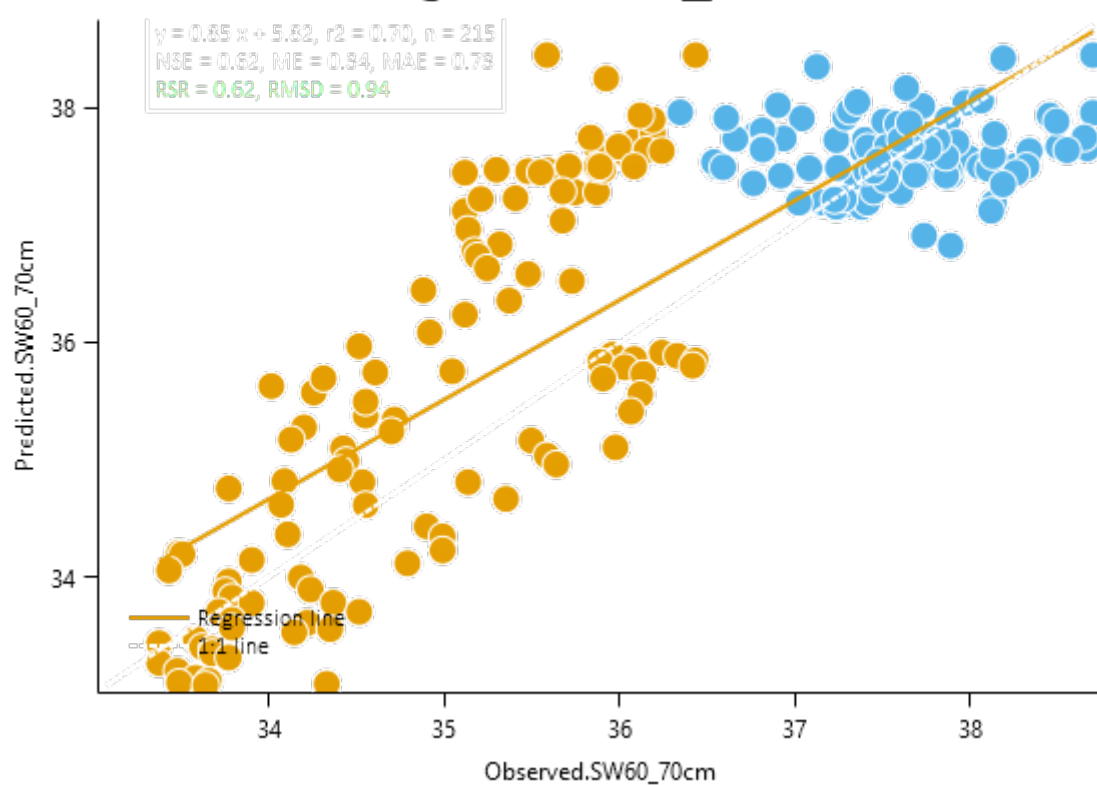
Regression30_40cm



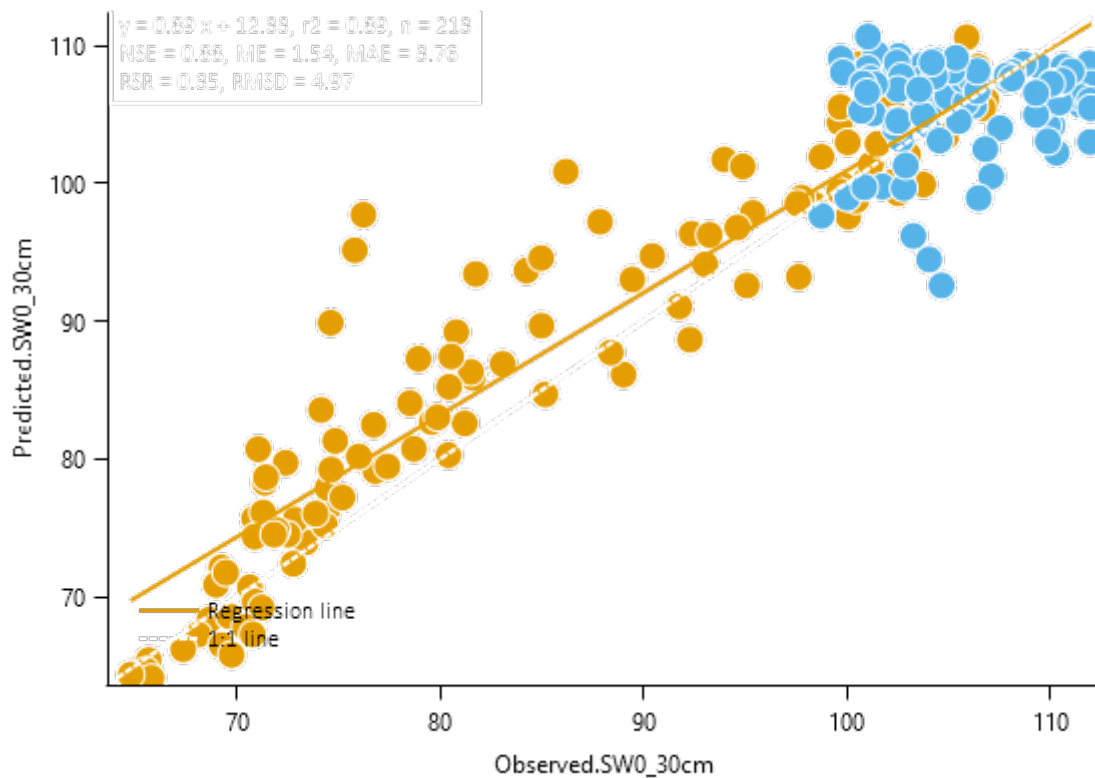
Regression50_60cm



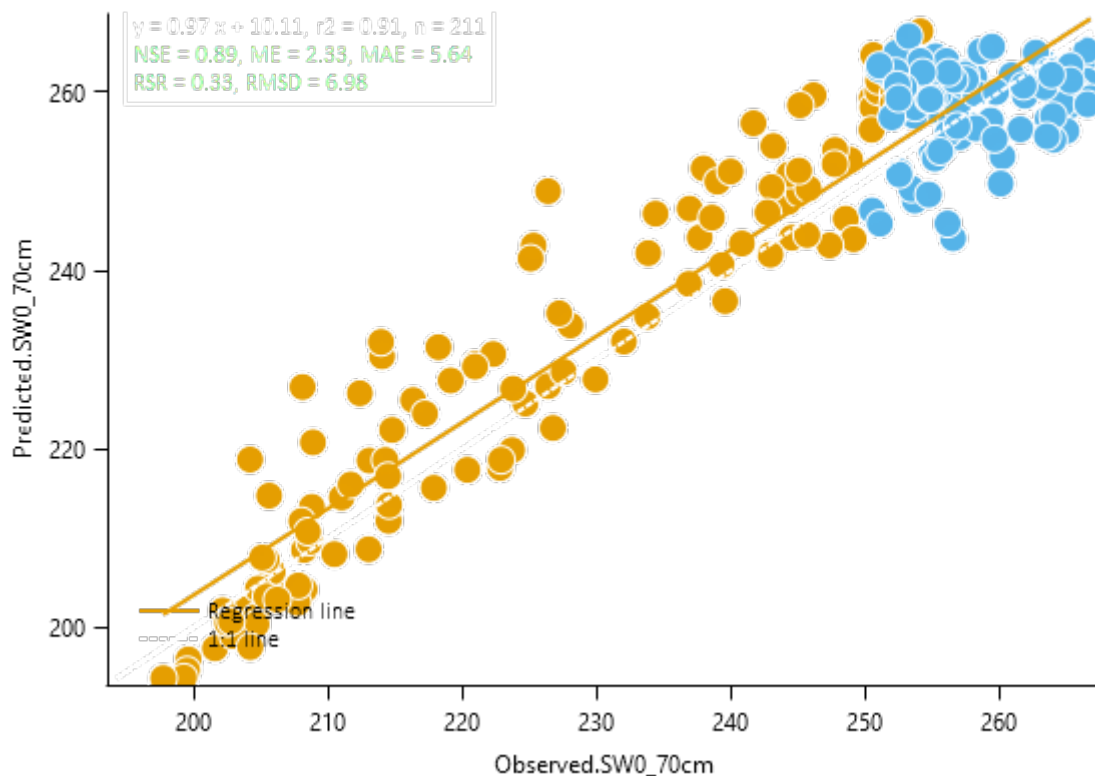
Regression60_70cm



Regression0_30cm



Regression0_70cm



23 References

- BA Keating, Commonwealth Scientific, H Meinke, ME Probert, NI Huth, IG Hills, 2001. NWheat: documentation and performance of a wheat module for APSIM. CSIRO Division of Agriculture Technical Memorandum, CSIRO Division of Tropical Agriculture, Brisbane.
- Brown, H. E., Jamieson, P. D., Brooking, I. R., Moot, D. J., Huth, N. I., 2013. Integration of molecular and physiological models to explain time of anthesis in wheat. *Annals of Botany* 112 (9), 1683-1703.

- Brown, Hamish E., Huth, Neil I., Holzworth, Dean P., Teixeira, Edmar I., Zyskowski, Rob F., Hargreaves, John N. G., Moot, Derrick J., 2014. Plant Modelling Framework: Software for building and running crop models on the APSIM platform. *Environmental Modelling and Software* 62, 385-398.
- Evers, Jochem B, Vos, Jan, Andrieu, Bruno, Struik, Paul C, 2006. Cessation of tillering in spring wheat in relation to light interception and red: far-red ratio. *Annals of Botany* 97 (4), 649-658.
- Jamieson, P. D., Brooking, I. R., Porter, J. R., Wilson, D. R., 1995. Prediction of leaf appearance in wheat: a question of temperature. *Field Crops Research* 41 (1), 35-44.
- Jamieson, P. D., Semenov, M. A., Brooking, I. R., Francis, G. S., 1998. Sirius: a mechanistic model of wheat response to environmental variation. *European Journal of Agronomy* 8, 161-179.
- Lawless, Conor, Semenov, MA, Jamieson, PD, 2005. A wheat canopy model linking leaf area and phenology. *European Journal of Agronomy* 22 (1), 19-32.
- Monteith, J. L., Moss, C. J., 1977. Climate and the Efficiency of Crop Production in Britain [and Discussion]. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 281 (980), 277-294.
- Peake, A., Whitbread, A., Davoren, B., Braun, J., Limpus, S., 2008. The development of a model in APSIM for the simulation of grazing oats and oaten hay. In: *Global Issues, Paddock Action. Proceedings of the 14th ASA Conference, 21-25 September 2008, Adelaide, South Australia.*, Eds: Unkovich, M.
- Reyenga, P.J., Howden, S. M., Meinke, H., McKeon, G.M., 1999. Modelling global change impacts on wheat cropping in south-east Queensland, Australia. *Environmental Modelling & Software* 14, 297-306.
- S Asseng, B.A Keating, I.R.P Fillery, P.J Gregory, J.W Bowden, N.C Turner, J.A Palta, D.G Abrecht, 1998. Performance of the APSIM-wheat model in Western Australia . *Field Crops Research* 57 (2), 163 - 179.
- S Asseng, A Bar-Tal, J.W Bowden, B.A Keating, A Van Herwaarden, J.A Palta, N.I Huth, M.E Probert, 2002. Simulation of grain protein content with APSIM-Nwheat . *European Journal of Agronomy* 16 (1), 25 - 42.
- Wang, E., Robertson, M. J., Hammer, G. L., Carberry, P. S., Holzworth, D., Meinke, H., Chapman, S. C., Hargreaves, J. N. G., Huth, N. I., McLean, G., 2002. Development of a generic crop model template in the cropping system model APSIM. *European Journal of Agronomy* 18 (1-2), 121-140.