Interaktion von erhöhten CO_2 Konzentrationen und Stickstoff Metabolismus in Pflanzen

Interaction of elevated CO_2 and nitrogen metabolism in plants

Von der Fakultät Energie-, Verfahrens- und Biotechnik der Universität Stuttgart zur Erlangung der Würde eines Doktor der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von Konrad Krämer
aus Pforzheim

Hauptberichter: Prof. Dr. Arnd G. Heyer
Mitberichter: PD Dr. Michael Schweikert
Vorsitzende: Prof. Dr. Ingrid Weiss
Tag der mündlichen Prüfung: 08.09.2022

Institut für Biomaterialien und biomolekulare Systeme
Abteilung für Pflanzen-Biotechnologie
Universität Stuttgart
Deutschland
2023

Contents

Abbreviations 1

Zusammenfassung 2

Abstract 4

Objective and motivation 6

Theoretical background 8

Method development 17

Results and Discussion 19

Outlook 30

Results in form of publications 31

Acclimation to elevated CO_2 affects the C/N balance by reducing de novo N-assimilation 32

Interaction of Nitrate Assimilation and Photorespiration at Elevated CO_2 54

R Package 'paropt' 86

R Package 'ast2ast' 105

Appendix A 125

References 140

Lebenslauf 149

Eigenständigkeitserklärung 150

Acknowledgement 151

Abbreviations

ambient CO_2	aCO_2
elevated CO_2	eCO_2
nitrogen	N
carbon	C
glycine	Gly
serine	Ser
glutamate	Glu
glutamine	Gln
citrate	Cit
malate	Mal
fumarate	Fum
malate & fumarate	MF
glucose	Glc
fructose	Frc
sucrose	Suc
photosynthesis	PS
photorespiration	PR
Hydroxypyruvate-reductase	HPR
Glutamine-Synthetase	GS
Nitrate-reductase	NR
Glutamin-oxoglutarat-amino-Transferase	GOGAT
sucrose-phosphate-synthase	SPS
glutamate glyoxylate amino-transferase	GGT
serine glyoxylate amino-transferase	SGT

Zusammenfassung

Im Zuge dieser Arbeit wurde die Interaktion von erhöhter CO_2 (eCO_2) Konzentration und Stickstoff Metabolismus untersucht. Für die Modellpflanze *Arabidopsis thaliana* wurde zunächst nachgewiesen, dass eine mindestens sechswöchige Exposition gegenüber eCO_2 zu einem Rückgang der Photosyntheseleistung führt, der für andere Spezies beschrieben war und als Akklimatisierung der Photosynthese bezeichnet wird. Deren Ursache ist unbekannt; es wird aber vermutet, dass eine Beziehung zum Stickstoffmetabolismus besteht. Der erste Teil der vorliegenden Arbeit stellt die Hypothese auf, dass dieser Zusammenhang über den Stoffwechselweg der Photorespiration gebildet wird. Ausgang der Photorespiration ist die Oxigenierung von Ribulosebisphosphat im Calvin-Benson Zyklus, die unter eCO_2 vermindert abläuft. Um der Hypothese nachzugehen, wurden Metabolitprofile für akklimatisierte und nicht akklimatisierte Pflanzen des Wildtyps Col-0 und der *hpr1-1* Mutante aufgenommen. Die *hpr1-1* Mutante weist einen Defekt im Gen der peroxisomalen Hydroxy-Pyruvat-Reduktase auf und ist daher nur bedingt in der Lage, im Rahmen der Photorespiration anfallendes Hydroxy-Pyruvat zu metabolisieren. Die tageszeitlichen Verläufe der Metabolitprofile und Aktivitäten ausgewählter Enzyme, die im Primärstoffwechsel zentrale Positionen einnehmen, wurden einer simulationsgestützten Datenanalyse unterzogen, um Parameter relevanter Stoffwechselwege zu identifizieren. Dadurch konnte nachgewiesen werden, dass die verringerte Photorespiration unter eCO_2 verantwortlich für die Akklimatisierung ist. PR fungiert als Speicher für assimilierten Stickstoff und liefert Kohlenstoff-Gerüste für dessen *de novo* Assimilation. In der *hpr1-1* Mutante ist der Rückfluss von Kohlenstoffverbindungen aus dem PR in den Calvin-Benson Zyklus gestört, wodurch es zu einer starken Akkumulation von Aminosäuren kommt.

Aufbauend auf diese Ergebnisse wurde die Interaktion zwischen PR und N Assimilation weiter untersucht. Ausgangspunkt dieser Untersuchungen war die Beobachtung, dass im Rahmen der PR frei gesetztes Ammonium mit Nitrat um die Kohlenstoffquelle für die Aminosäure-Synthese konkurriert und so möglicherweise die Aktivität der hierfür relevanten Enzyme Glutaminsynthetase (GS) und Glutamin-Oxoglutarat-Aminotransferase (GOGAT) stimuliert. Für Pflanzen, die mit Nitrat als einziger N-Quelle versorgt wurden, konnte nachgewiesen werden, dass PR den Umsatz im GS/GOGAT Zyklus tatsächlich erhöht. Darüber hinaus zeigte das Model, dass die Biomassebildung aus Hexosephosephosphaten im Vergleich zur Nitrat Assimilation unter eCO_2 bei reduzierter PR erhöht ist. Dies

weist nach, dass unter eCO_2 in der pflanzlichen Biomasse Stickstoffverbindungen durch vermehrt assimilierten Kohlenstoff verdünnt werden. Weiterhin zeichnete sich ab, dass ein gesteigerter Bedarf an Zellenergie für die Verarbeitung der PR-Zwischenstufen das Wachstum der *hpr1-1* Mutante beeinträchtigen könnte.
Um die systembiologischen Modellierungen durchführen zu können, wurde das Software-Paket *paropt* entwickelt, das es ermöglicht, Optimierungen von Parametern von ODE-Systemen schnell und in hoher Qualität durchzuführen. Die Software wurde in der Programmiersprache C++ geschrieben. Als package für die Programmierumgebung R ist paropt öffentlich verfügbar, um die Software einem breiten Publikum in der Biologie zugänglich zu machen und eine verbesserte Nutzerfreundlichkeit zu erreichen. Besonders hervor zu heben ist, dass der ODE-Solver und der implementierte Optimierer auf dem neusten Stand sind. Um dem Anwender zu ermöglichen, das ODE-System in R zu schreiben ohne auf die Geschwindigkeit von C++ verzichten zu müssen, wurde ein weiteres R package mit dem Namen *ast2ast* entwickelt. Diese Software übersetzt einen Teil von R in C++. Dadurch ist es möglich, ohne Kenntnisse in der Programmiersprache C++ die gesamte Simulation in R durchzuführen, was sehr benutzerfreundlich ist.

Abstract

This work investigates the interaction of elevated CO_2 (eCO_2) concentration and nitrogen metabolism in plants. For the model plant *Arabidopsis thaliana*, it was demonstrated that exposure to eCO_2 for at least six weeks leads to a decrease in photosynthetic activity which has been described for other species and is referred to as acclimation of photosynthesis. The underlying reason for acclimation is unknown. However, it is supposed that a relationship exists to nitrogen metabolism. In this study, the hypothesis was investigated that photorespiration (PR) builds the link between nitrogen metabolism and acclimation to eCO_2. PR starts with the oxygenation of ribulose bisphosphate in the Calvin-Benson cycle, and this is decreased at eCO_2. To address the hypothesis, metabolite profiles were recorded for acclimated and non-acclimated plants of the wild-type Col-0 and the *hpr1-1* mutant. This mutant has a defect in the gene for the peroxisomal hydroxy-pyruvate reductase and is therefore restricted in the turnover of hydroxy-pyruvate that is produced during PR. The diurnal courses of the metabolite-profiles and activities of selected enzymes that occupy central positions in the primary metabolism were used in simulation-based data analysis to identify parameters of relevant metabolic pathways. This allowed proving that the reduced PR under eCO_2 is responsible for acclimation. PR acts as a store for assimilated nitrogen (N) and provides carbon scaffolds for its *de novo* assimilation. In the *hpr1-1* mutant, the reflux of carbon compounds from the PR into the Calvin-Benson cycle is disrupted. This leads to a massive accumulation of amino acids.

Based on these results the interaction between PR and N assimilation was further investigated. It was found that ammonium, released during PR, competes with nitrate for the carbon scaffolds in amino acid synthesis, and this appears to stimulate the activity of the enzymes glutamine synthetase (GS) and glutamine-oxoglutarate aminotransferase (GOGAT), which are most relevant in N fixation. For plants supplied with nitrate as the sole N source, it was shown that PR increases turnover in the GS/GOGAT cycle. Furthermore, the model showed that the biomass formation from hexose phosphates was increased compared to nitrate assimilation under eCO_2 at reduced PR. This substantiates that under eCO_2 in plant biomass nitrogen compounds are diluted by additionally assimilated carbon. Furthermore, it was shown that increased demand for cellular energy in the processing of the PR-intermediates could negatively affect growth of the *hpr1-1* mutant.

To enable systems biology modelling, the software package, *paropt*, was developed, which enables fast and reliable optimization of parameters of ODE systems. The software was written in the C++ programming language. As a package for the programming environment R, *paropt* is available to the scientific community. The ODE solver and the implemented optimizer are state-of-the-art technology. To enable writing of the ODE-system in R without having to sacrifice the speed of C++, another R package was developed, called *ast2ast*. This software translates a part of R into C++. Thus it is possible to conduct the entire simulation in R without knowledge of the programming language C++.

4 Objective and motivation

This study is devoted to investigating several aspects of elevated atmospheric CO_2 (eCO_2) and its interaction with plant central metabolism.

One focus is the acclimation of photosynthesis to eCO_2. Currently many theories exist, attempting to explain this phenomenon (Wong, 1990; Griffin and Seemann, 1996; Ainsworth and Rogers, 2007). However, a definite proof of one of the proposed mechanisms is missing. In this study the role of photorespiration (PR) was considered by comparing photosynthetic acclimation of wild type plants with that of the mutant *hpr1-1*, which is defective in a late step of the photorespiratory pathway (Timm *et al.*, 2008).

To examine the acclimation of plants to eCO_2, the first step was to establish conditions that lead to diminished photosynthetic activity after a certain time at eCO_2. To this end, gas exchange was measured over full diurnal cycles following different times of exposure to eCO_2 and net photosynthesis (PS) was calculated.

After having established a 46 day interval of exposure to eCO_2 as sufficient for acclimation, acclimated plants and plants exposed to eCO2 for only 48 h were used for metabolic profiling over a full diurnal cycle with sampling intervals of 2 h. In addition, enzymatic activities of sucrose-phosphate-synthase (SPS), hexokinases (glucokinase and fructokinase) and hydroxypyruvate-reductase (HPR) were quantified that were later used for mathematical modeling of primary metabolism (Krämer *et al.*, 2022a).

GC-MS/MS development

For the analysis of the physiological role of photorespiration a method was developed that allowed quantification of photorespiratory intermediates, glycine (Gly) and serine (Ser), on a GC-MS/MS device. The method comprised extraction, subsequent drying of the samples and their derivatisation. Parameters of gas chromatography, e.g., the heating rate of the column, had to be identified. In addition, a multiple reaction mode (MRM) method was developed allowing detection of the metabolites within multicomponent peaks with increased sensitivity.

Software development

After accomplishment of the analytical work, the data set was used in mathematical simulations. To this end, an ODE-system was set up, describing sucrose-cycling, a

simplified Tri-Carbonic-Acid (TCA) cycle, N assimilation, and the photorespiratory pathway. Using this model and the start values of the metabolites, the parameters for the reactions represented in the model, were optimized in order to obtain *in silico* state trajectories matching the measured time courses.
For executing these simulations, a software was developed that allows fast and high-quality optimization of parameters of ODE-systems (Krämer *et al.*, 2021).

Cross-talk between PR and N-assimilation

Ample evidence has been presented for a link of PR and N-assimilation (Rachmilevitch, Cousins and Bloom, 2004; Bloom, 2015). However, there is an ongoing discussion on how eCO_2, which reduces the rate of PR, affects N-assimilation (Andrews *et al.*, 2020; Bloom, Kasemsap and Rubio-Asensio, 2020). Thus, the consequences of reduced or constrained flux through the PR pathway on N-metabolism cannot easily be inferred. For a systematic study of the interaction of eCO_2 and nitrogen (N) assimilation, wildtype and *hpr1-1* mutant plants were analyzed that grew at either ambient CO_2 (aCO_2) or eCO_2 with nitrate as sole N source. Samples were taken over a full diurnal cycle, and metabolic data was obtained including the compounds glutamate (Glu) and glutamine (Gln). Moreover, the v_{max} values of the three enzymes HPR, nitrate reductase (NR) and glutamine-synthetase (GS) were recorded. Using this data in simulations, fluxes for PR and N assimilation could be calculated and analyzed (Krämer *et al.*, 2022b).
Beyond the cross-talk between PR and N-assimilation the question was addressed why *hpr1-1* plants suffer at aCO_2. The mutant *hpr1-1* is one of the few mutants affected in the PR pathway that is viable at aCO_2 concentrations. However, results of metabolite analysis from the literature (Timm *et al.*, 2008) and own data contrasted with the observed dwarfed and chlorotic phenotype of the mutant (J. Li *et al.*, 2019) at aCO_2. In this study several mechanism were proposed, such as high cell energy costs for NH_4^+ re-assimilation, which could explain the mutant phenotype (Krämer *et al.*, 2022b).

5 Theoretical background

Climate change

In the course of climate change the CO_2 concentration in the atmosphere drastically increases (Fig. 1) (Siegenthaler and Oeschger, 1978; Szulejko *et al.*, 2017). This alteration in CO_2 levels will cause a rise in sea levels (Titus and Narayanan, 1996) and a drastic increase in temperature (Syukuro and T. Wetherald, 1974; Syukuro and Richard T., 1980) (https://github.com/Konrad1991/climatechange). Rising temperatures affect plants in several ways. For instance, the number of collars and cumulative leaf area in corn is increased by higher temperatures (Hatfield and Prueger, 2015). Moreover, an enhanced biomass production was observed (Tungate *et al.*, 2007), and Slot and Kitajima (2015), showed that higher temperatures are accompanied by increased respiration rates. Increasing CO_2 concentrations will affect ribulose-1,5-bisphosphat-carboxylase/-oxygenase (RUBISCO). This key enzyme of the Calvin-Benson-cycle catalyzes a carboxylation reaction if CO_2 is the substrate. This results in two molecules of 3-phosphoglycerate. In contrast, if oxygen is used, RUBISCO catalyzes oxygenation of ribulose-1,5-bisphosphat, which leads to one molecule of 3-phosphoglycerate and one molecule of 2-phospho-glycolate (Leegood *et al.*, 1995). The altered ratio of CO_2 to O_2 will decrease the oxygenation- to-carboxylation ratio (Fig. 1) (Sharkey, 1988). Plants have to eliminate 2-phospho-glycolate (2PG) due to its strong toxicity, and this is accomplished in a metabolic pathway called photorespiration (PR). Considering the decreasing oxygenation rate, the input into PR will decrease in the future.

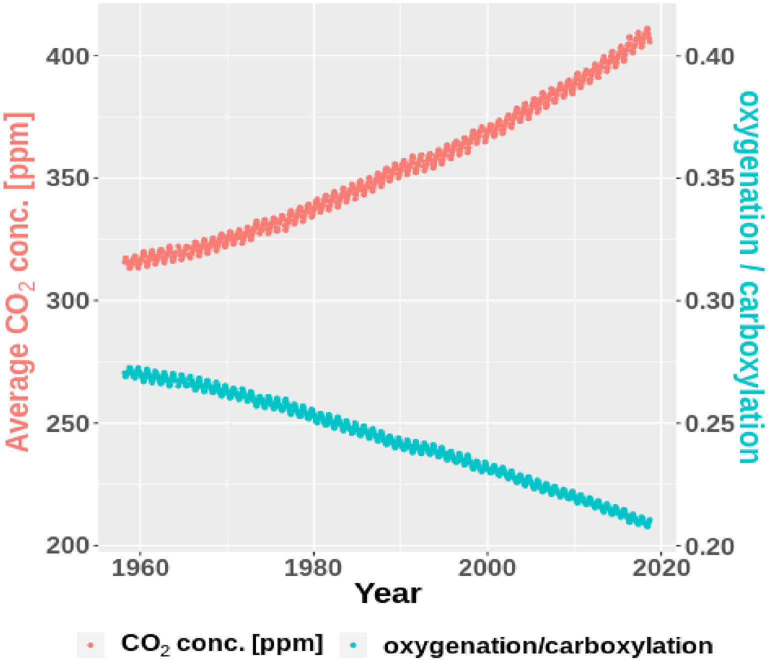


Figure 1: Red: CO_2 concentration across the last 60 years (https://datahub.io/core/co2-ppm#resource-co2-ppm_zip, Download 01.12.2021 11:03). Blue: ratio of oxygenation and carboxylation over the last 60 years. Based on the CO_2 values and the equation of Sharkey, 1988.

Photorespiration

As can be seen in Fig. 1, the ratio of oxygenation vs. carboxylation is currently ca. 0.2, meaning that about 20% of the reactions of RUBISCO are using O_2 instead of CO_2. Therefore, the flux through PR is the second largest following PS. Unsurprisingly, almost all mutations in the PR pathway are lethal at aCO_2.

Reactions in the chloroplast

In case RUBISCO uses oxygen instead of CO_2 as substrate, one molecule of 2PG is produced, which has to be processed by the PR pathway to avoid accumulation of a toxic compound (Leegood *et al.*, 1995). The first step is the conversion of 2PG to glycolate by the enzyme phosphoglycolate phosphatase which removes the phosphate group (Fig. 2).

Reactions in the peroxisomes

Subsequently, glycolate is transported from the chloroplast to the peroxisomes, where it is oxidized by the glycolate oxidase to glyoxylate. The electrons are transferred to O_2 forming H_2O_2. The H_2O_2 is disproportionated by the enzyme catalase. Next, glyoxylate is transaminated forming Gly. The N donor is either Ser or Glu resulting in hydroxy-pyruvate (HyPy) or α-KG, respectively (Fig. 2).

Reactions in the mitochondria

Gly is transported into mitochondria, where two molecules of Gly are converted to Ser. The first reaction is catalyzed by the glycine-decarboxylase (GDC) which release NH_4^+ and CO_2 as byproducts. Thus, already assimilated C and N are lost during PR. During the reactions NAD^+ is reduced forming NADH, and the remaining C1-body is transferred to 5,6,7,8-tetrahydropteroylpolyglutamate (THF), producing N5,N10-methylene-5,6,7,8- tetrahydropteroylpolyglutamate (CH_2-THF) (Douce *et al.*, 2001). CH_2-THF and another molecule of Gly are used by the serine-hydroxymethyltransferase (SHMT) to produce Ser and recover THF (Fig. 2).

Back to peroxisomes and chloroplast

The next step is the transport of Ser into the peroxisomes. As already described, Ser can serve as N donor for glyoxylate amination. This reaction is catalyzed by the enzyme serine/glyoxylate aminotransferase (SGT). The resulting C-skeleton, hydroxypyruvate (HyPy), is reduced by HPR forming glycerate. Besides that, HyPy can also be reduced in the cytosol. Therefore, mutants of the peroxisomal HPR are rare examples of PR pathway mutations which are not lethal at aCO_2 (Timm *et al.*, 2008). The last step of PR is the transport of glycerate into the chloroplast, where it is phosphorylated by the glycerate-kinase forming 3-PGA, an intermediate of the Calvin-Benson-cycle (Fig. 2).

Alternative reactions and cell-energy balance

While H_2O_2, which originates from glycolate oxidation, can be disproportionated by catalase, it can also react with glyoxylate yielding formiate and CO_2. In addition, it can react with HyPy forming glycolate and CO_2. Both reactions run non-enzymatically (Walton and Butt, 1981; Cousins *et al.*, 2008). It is known that ca. 10% of HyPy or glyoxylate are decarboxylated (Walton and Butt, 1981) Thus, an additional loss of C is the consequence. The formiate can be used to produce CH_2-THF and is therefore not lost (Wingler *et al.*, 2000; Cousins *et al.*, 2008). In conclusion, the PR pathway results in the loss of at least one unit of reduced carbon for two oxygenation reactions. Furthermore, one ATP is required as substrate for glycerate kinase and another one for the re-fixation of NH_4^+ by GS. In addition to that, the Glutamin-oxoglutarat-amino-transferase (GOGAT) requires two molecules of reduced ferredoxin for the conversion of Gln and α-KG to two molecules of Glu. The GOGAT reaction is required in order to maintain the flux through the GS/GOGAT cycle. This clearly demonstrates the high energetic and metabolic costs of PR.

Peterhansel and Maurino, 2011, proposed two possible shortcuts of the PR pathway taken from bacterial pathways. The first converts glycolate to glycerate via glyoxylate and tartronate semialdehyde. Within this pathway only one CO_2 is released and no cell-energy is consumed.

The second pathway is the conversion of glycolate to pyruvate (Pyr) with glyoxylate and Mal as intermediates. This would result in the release of two molecules of CO_2, and again, no cell-energy would be consumed. Thus, the question arises, why plants use the more costly and more complex PR pathway.

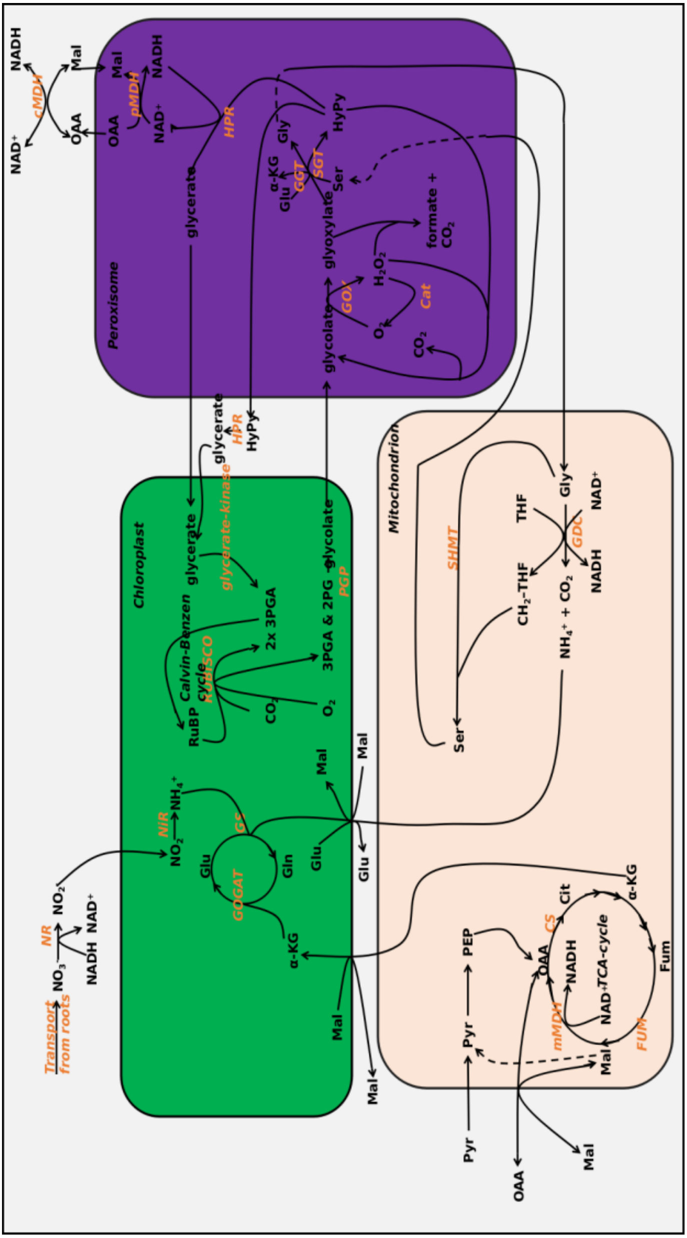


Figure 2: Metabolic network describing photorespiration, nitrogen assimilation and the TCA cycle. RUBISCO: ribulose-1,5-bisphosphat-carboxylase/-oxygenase, PGP: phospho-glycolate phosphatase, GOX: glycolate oxidase, Cat: catalase, GGT: glutamate/glyoxylate transaminase, SGT: Serine/Glyoxylate transaminase, HPR: Hydroxypyruvate reductase, pMDH: peroxisomal malate dehydrogenase, cMDH: cytosolic malate dehydrogenase, mMDH, mitochondrial malate dehydrogenase, GDC: glycine-decarboxylase, SHMT: serine hydroxymethyl-transferase, CS: cytrate synthase, FUM: fumarase, GS: glutamine synthetase, GOGAT: glutamine-oxoglutarate-aminotransferase, NR: nitrate reductase.

Nitrogen assimilation

Nitrogen (N) is an important element in plant metabolism, being part of essential compounds such as proteins and DNA (Heldt and Piechulla, 2015). Therefore, the assimilation of N is a pivotal metabolic process.

Uptake of N

The first step in N acquisition is the uptake of either nitrate or ammonium from the soil. Despite ammonium assimilation requires less cellular energy, plants rarely use ammonium (Noguero and Lacombe, 2016) although transporters exist enabling import into root cells. Within the cytosol ammonium can be assimilated or transported across the tonoplast into the vacuole (Ludewig, Neuhäuser and Dynowski, 2007). The assimilation of NH_4^+ is conducted by the isoenzymes glutamine synthetase 1 and 2 (GS1, GS2). In the root GS1 is the dominating activity, whereas in the shoots GS2 is the main isoform. Both isoforms transfer the nitrogen to Glu forming Gln in a coupled reaction of ATP hydrolysis (Masclaux-Daubresse *et al.*, 2010).

Plants can import NO_3^- using two different transport systems. A low affinity system, which includes, among others, the nitrate transporter (NRT1.2) and a high affinity system which includes NRT2.1 and NRT2.2 (Tsay *et al.*, 1993, 2007; Noguero and Lacombe, 2016). These transporters enable symport of NO_3^- and H^+ ions into the root. Thus, nitrate acquisition depends on a pH gradient across the membrane, established by a H^+-ATPase that exports H^+ while dephosphorylating ATP (Seyoshi, Ishikawa and Abdel-latif, 2010).

N assimilation

Subsequently, NO_3^- can be transported into the vacuole, assimilated, or transported via the xylem to the source tissues (P. Li *et al.*, 2008; Lin *et al.*, 2008; He, Ma and Zhang, 2016). For NO_3^- assimilation, it is reduced to nitrite by the enzyme nitrate reductase. The enzyme requires NAD(P)H as electron donor (Chamizo-Ampudia *et al.*, 2017). Nitrite is transported into the plastids where it is reduced by nitrite reductase (NiR) using 3 molecules of NADPH in the roots and 6 molecules of reduced ferredoxin in the shoot, respectively (Miller and Cramer, 2005). The product of the reaction is ammonium, which acts as an uncoupling agent of ATP synthesis and must be processed by the GS/GOGAT cycle (Bittsánszky *et al.*, 2015). Ammonium is incorporated into Glu by GS forming Gln. This reaction is energetically unfavorable and, therefore, it is coupled to ATP dephosphorylation. In the shoots Gln and α-KG are used by GOGAT in the chloroplasts for production of two Glu molecules with reduced ferredoxin acting as electron donor (Fig. 2) (Sahay *et al.*, 2021).

Cross-talk between PR and N assimilation

As the photorespiratory pathway is coupled to PS, strong diurnal dynamics can be observed in the compounds related to PR, such as Gly, Ser and glycerate (Timm *et al.*, 2021). Because the flux through PR is the second largest observed in plant metabolism, the NH_4^+ released by PR by far exceeds that originating from primary N assimilation (Bauwe, Hagemann and Fernie, 2010). Additional compounds involved in the PR pathway as well as primary N assimilation are α-KG and Glu. This illustrates the interlink between the two pathways. For safely transporting NH_4^+ from mitochondria to plastids, Linka and Weber, 2005, proposed two alternative shuttles. These would transport NH_4^+ in the form of either Gln or citrulline.

Due to the high activity of GDC involved in PR, a large amount of ammonium accumulates in mitochondria, which has to be re-fixed by the GS/GOGAT cycle (Fig. 2). In addition to that, Glu is consumed in the peroxisomes for Gly production. The resulting α-KG must be transported to the plastids, where it is used in the GOGAT reaction. Interestingly, the import of α-KG takes place as an antiport exporting Mal into the cytosol, while the export of Glu is connected to an import of Mal into the plastids (Renné *et al.*, 2003). Thus, the ratio of GGT to SGT activity is connected to the GS/GOGAT cycle.

In this study the *hpr1-1* mutant was used to study the effect altered PR flux. The mutant shows accumulation of Gly and Ser (Timm *et al.*, 2008) due to the impairment of the HPR reaction. This in turn affects the ratio of GGT to SGT. The results of the simulations presented here show that the mutant requires higher GGT activities as the wildtype Col-0. The byproduct of GGT is α-KG which is transported into the plastids in exchange of Mal. In the cytosol Mal is oxidized by the cytosolic malate dehydrogenase (cMDH) forming NADH and oxaloacetate. NADH can be used by NR for N assimilation (Rachmilevitch, Cousins and Bloom, 2004). The theory of Rachmilevitch, Cousins and Bloom, 2004, indicates that PR activity stimulates NADH production in the cytosol as it leads to increased export of Mal from the plastids into the cytosol in exchange with α-KG.

N assimilation at eCO_2

Bloom *et al.*, 2014, showed that eCO_2 leads to decreased nitrate assimilation. In contrast Andrews *et al.*, 2019, 2020, found evidence that eCO_2 does not affect NO_3^- assimilation.

Other effects of eCO_2 linked to N assimilation are inhibition of succinate dehydrogenase and cytochrome C oxidase and an impairment of nitrite import into the plastids (Asensio, Rachmilevitch and Bloom, 2015).

The overall flux through PR is decreased at eCO_2 and this has potential consequences for N assimilation. First of all the amount of α-KG originating from PR is decreased. In addition, the transport of Mal into the cytosol is coupled to the import of α-KG into plastids. Thus, less Mal is available for the cMDH enzyme which results in a decreased $NADH/NAD^+$ ratio thereby limiting NR (Rachmilevitch, Cousins and Bloom, 2004; Bloom *et al.*, 2010). Moreover, the substrate availability for GS and GOGAT could be disturbed because reduced PR alters the α-KG, Glu and NH_4^+ availability.

NH_3 plays an important role in the stimulation of anapleurotic reactions as it increases the activity of PEP carboxylase and Pyr kinase (Leegood *et al.*, 1995). Elevated CO_2 concentrations reduce the flux into PR. Thus, the turnover of GDC is reduced and less NH_4^+ is released. This results in a lower flux into the TCA cycle at eCO_2.

Acclimation to eCO_2

Plants which were transferred from aCO_2 to eCO_2 initially benefit from the altered environment. This results in higher net PS rates. However, after a certain time a decline is observed by up to 25%. This process is called acclimation to eCO_2 (Griffin and Seemann, 1996; Smith and Dukes, 2013) and is known for a long time. However, a proven explanation is still lacking.

Proposed theories for the acclimation to eCO_2

One theory that could explain the acclimation is the decreased stomatal conductance at eCO_2 (Shi *et al.*, 2021). A higher internal CO_2 concentration in the mesophyll would result in closure of stomata, and as a consequence the transpiration rate would decrease and thus the source tissues are sub-optimally supplied with minerals.
Another explanation is the down regulation of photosynthesis by an accumulation of carbohydrates (Dabu *et al.*, 2019; Gámez *et al.*, 2020; Kizildeniz *et al.*, 2021). Krapp *et al.*, 1993, showed that expression of genes involved in PS are down regulated by the addition of carbohydrates. This implies that the capacity for utilization of carbohydrates is too low at eCO_2 and thus, the increase in sugar levels would repress genes encoding components of the PS machinery (Drake, Gonzàlez-Meler and Long, 1997).
In addition, it is discussed that plant PS might be limited by N depletion. One explanation would be that C uptake is faster then N assimilation which would result in increasing C/N ratios (Chen *et al.*, 2005; Bloom, 2010). Additional factors, such as disturbed nitrate assimilation (Bloom *et al.*, 2014), the inhibition of succinate dehydrogenase, cytochrome C oxidase or the impaired import of nitrite into plastids at eCO_2 (Asensio Rachmilevitch and Bloom, 2015) could cause the increased C/N ratio.
Another possibility to explain the acclimation to eCO_2 would be diminished PR. As already described, this could decrease nitrate assimilation by reducing the NADH amount (Bloom, 2010). Alternatively, the potentially lower NH_3 concentration at eCO_2 disturbs the anapleurotic reactions into the TCA-cycle and/or the GS/GOGAT cycle.

6 Method development

The aim of inferring metabolic processes leading to photosynthetic acclimation at eCO_2 by means of mathematical modeling required a thorough quantification of metabolite dynamics over diurnal cycles, integrating data from different platforms such as gas exchange and solute quantification.

GC-MS/MS method development

For quantifying photorespiratory intermediates (Gly and Ser) and α-KG a gas chromatography device was used, which was coupled to two mass selective detectors (GC-MS/MS). This method is often employed for detection of compounds involved in plant metabolism (Lisec *et al.*, 2006; Alseekh *et al.*, 2021). But exact quantification of metabolites is still a challenge.

To improve reliability of metabolite quantification, the first step was improving the extraction of the metabolites from the plant tissue. As a starting point the method described by Lisec *et al.*, 2006, was used. Further steps were needed in order to enhance the extraction of carbonic acids. An additional extraction with H_2O was implemented similar to the carbonic acid extraction already described by Küstner, Nägele and Heyer, 2019. Afterwards, the samples were dried and the resulting pellets were derivatized as described in the literature (Lisec *et al.*, 2006). Subsequently, the metabolites were measured as described in Krämer *et al.*, 2022a.

In conclusion, the procedure developed within the course of this study has several improvements compared to previous methods. Regarding the carbonic acids the extraction was drastically improved. Moreover, the identified parameters for the GC-MS/MS device allowed sensitive, fast quantification of the compounds Gly, Ser and α-KG. Beyond that the metabolites could be quantified in an absolute manner, which is a demanding task using GC-MS/MS.

Software method development

In order to conduct the simulations of the metabolic networks the software *paropt* was developed and published as a package for the R programming language (Krämer *et al.*, 2021). The R package *paropt* has several notable properties. First it uses cutting edge ODE-solvers (Hindmarsh *et al.*, 2005) which are fast and yield highly accurate results. Second, a state of the art particle swarm optimizer is implemented, which is particular suitable for parameter optimization of ODE-systems and can call the ODE-solver using parallel instances. Beyond *paropt* the R package *ast2ast* was developed (Krämer *et al.*, 2022c). This software translates R code into C++ code. Thus, it is possible to include more safety tests for the ODE-system, as the input is a R function and not a already compiled C++ function, thereby preventing more user errors. Moreover, the very error prone process of writing C++ code is not necessary when using *ast2ast*. In Appendix A the theoretical background for the software is explained. In conclusion, *paropt* and *ast2ast* are open-source software which enables fast and accurate simulations.

7 Results and Discussion

Within this study several hypotheses regarding the interaction of PR and N assimilation were investigated in the context of elevated CO_2 concentrations.

Acclimation to eCO_2

The first attempt was to figure out mechanisms causing photosynthetic acclimation to eCO_2. Therefore, conditions had to be established where plants show diminished net PS. After 46 days at eCO_2 acclimation could be demonstrated as a significant decline in net PS compared to plants which were kept at eCO_2 for 48 h. As can be seen in Tab. 1 (Krämer *et al.*, 2022a) and Fig. 3, net PS shows a strong decrease for the wildtype Col-0 and the mutants *aox1a* and *hpr1-1*. The mutant *aox1a* (SALK 084897C) showed almost the same behavior for the metabolic and enzymatic data as the wildtype, therefore it was not considered in Krämer *et al.*, 2022a.

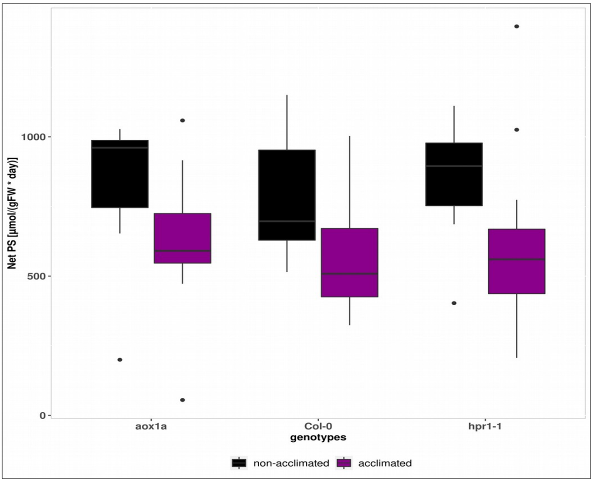


Figure 3: Boxplots showing acclimation of net PS to eCO_2. Net PS calculated for a full diurnal cycle. Col-0 non-acclimated n = 11, Col-0 acclimated n = 8, hpr1-1 non-acclimated n = 7, hpr1-1 acclimated n = 12, aox1a non-acclimated n = 10, aox1a acclimated n = 11. Data for Col-0 and hpr1-1 from Krämer et al., 2022a.

To figure out which mechanism underlies acclimation to eCO_2, metabolic and enzymatic data across a full diurnal cycle were gathered. Using this data, simulations were conducted employing the model presented in Krämer *et al.*, 2022a. The resulting optimized parameters allowed flux analysis of the reactions represented in Fig. 3 of Krämer *et al.*, 2022a. In addition, the obtained data could exclude some of the proposed hypotheses in the literature.

The theory of Shi *et al.*, 2021, stating that a reduced stomatal conductance at eCO_2 causes the acclimation, was not supported when analyzing the mineral content of acclimated plants. As can be seen in Fig. S1 of Krämer *et al.*, 2022a, there is no decrease in mineral concentration for acclimated plants. In contrast the minerals are differently distributed between the conditions. For instance, there is more sulfate but less phosphate found in acclimated plants. Thus, the supply with minerals is not generally reduced, which should be the case for a total reduction in xylem flux caused by reduced stomatal conductance.

Furthermore, our data contradict the low carbohydrate capacity theory put forward by Dabu *et al.*, 2019, Gámez *et al.*, 2020 and Kizildeniz *et al.*, 2021. The theory is based on a comparison of plants at aCO_2 and eCO_2, respectively, where the latter had higher sugar levels. For the experimental design used in the current study, where plants are exposed either long-term or short-term to eCO_2, no difference in the levels of carbohydrates Glc, Frc and Suc between acclimated and non-acclimated wildtype plants were found (Fig. S2). Thus, it is unlikely that carbohydrates are the reason for acclimation, especially when considering that elevated levels of carbohydrates over a long time interval are needed to repress expression of gens involved in PS (Krapp *et al.*, 1993).

In the *hpr1-1* mutant the non-acclimated plants showed lower Glc and Frc concentrations (Fig. S2 Krämer *et al.*, 2022a) compared to acclimated mutant plants. The treatment effect was significant when only the *hpr1-1* data is considered (Glc: P = 2.56e-08; Frc: P = 1.42e-08). Notably, the difference in carbohydrate level is too low in order to explain the strong decrease in net PS. It should be mentioned that *hpr1-1* plants shifted from aCO_2 to eCO_2 for only a short time have strongly elevated metabolic activity, which is evident from higher levels of respiration (see Table. S1 Krämer *et al.*, 2022a). Thus, the unusual low sugar levels in these plants may simply indicate high turnover rates for primary metabolites.

Furthermore, using only the non-acclimated Col-0 and *hpr1-1* plants, significant differences were found between the genotypes (Glc: P = 6.84e-05; Frc: P = 5.47e-05). However, using only the acclimated plants no significant differences between Col-0 and *hpr1-1* were detected. Thus, the reduced capacity for carbohydrate use at eCO_2

in the *hpr1-1* mutant is either a short-term effect or is a remnant of the condition at aCO_2.

Li *et al.*, 2019, proposed that Glc could be used in the oxidative part of the pentose-phosphate pathway to generate CO_2 which would decrease the flux into PR and thus alleviate the photorespiratory phenotype. Thus, the reduced Glc and Frc level could be due to the situation at aCO_2 and the mutant requires more then 48 h to refill the carbohydrate pools.

An additional theory explaining the acclimation to eCO_2 is the depletion of N. Several mechanism were proposed, which could lead to decreased N levels. The first one claims that N assimilation is not fast enough to keep up with the increased C uptake at eCO_2 (Chen *et al.*, 2005). This theory cannot be falsified using the quantified data or the analyzed fluxes. However, the acclimated plants have large amounts of NO_3^-. Thus, at least the uptake of N is not limiting. Bloom *et al.*, 2014 and Asensio, Rachmilevitch and Bloom, 2015, showed that eCO_2 affects the N assimilation in various ways, e.g., inhibition of succinate dehdyrogenase or the impaired import of nitrite into plastids. Notably, also the cytochrome C oxidase is inhibited at eCO_2. However, the data obtained for the wildtype plants do not show differences in dark respiration, which would be expected, if succinate dehdyrogenase and the cytochrome C oxidase are inhibited (Tab. S1 Krämer *et al.*, 2022a).

Nevertheless, a strong decrease in dark respiration was observed for the mutant following acclimation to eCO_2. A possible explanation could be reduced PS at eCO_2 in acclimated plants that would also reduce the input into PR, thus resulting in a smaller GDC flux. As one of the byproducts of GDC is NH_4^+, which is known to function as a stimulator for anapleurotic reactions (Leegood *et al.*, 1995) the flux into the TCA cycle may be reduced in the acclimated plants, thus lowering respiration rates. Therefore, the reduced respiration in acclimated *hpr1-1* plants may not be the cause of acclimation, but a consequence of the reduced PS.

Another explanation for the difference in dark respiration would be that the *hpr1-1* mutant has to metabolize PR intermediates, which accumulated during the light-phase. Thus, the nocturnal activity of GDC would increase dark respiration by producing CO_2.

Furthermore, Bloom, 2010, suggested that the diminished PR at eCO_2 could reduce the NADH amount in the cytosol thus limiting NR activity. The transport of Mal into the cytosol is connected to the import of α-KG into plastids. In the cytosol, Mal is oxidized in order to produce NADH. Additionally, Mal is imported into the plastids in exchange with Glu. The data obtained for Mal and Fum in (Krämer *et al.*, 2022a) is depicted in Fig. 4. For Mal a significant genotype effect was observed (P = 0.0103). The *hpr1-1* plants show slightly increased values compared to Col-0 (Fig. 4 A). As the information about the distribution within the cell is lacking, the location of Mal accumulation is not clear. It cannot be excluded that the export of Glu out of plastids was increased in *hpr1-1*, and this might have alleviated the PR phenotype as more Glu is available for GGT. However, this would have decreased the Mal pool available for oxidation in the cytosol to supply NR with NADH. Interestingly, the Fum pool (Fig. 4 B) is increased for acclimated plants. As Fum is a C storage in *Arabidopsis thaliana (Chia et al., 2000)* it is an indication that acclimated plants cannot use assimilated carbon because N is limiting.

Considering that the theory that PR increases NADH in cytosol suggested by Bloom, 2010, is in conflict with the fact that Mal is not only exported into the cytosol, but also imported into the plastids due to photorespiratory activity, Krämer *et al.*, 2022a, proposed a different theory where PR acts as N sink for newly assimilated nitrogen. A flux analysis (Krämer *et al.*, 2022a) showed that the turnover of total amino acids (AA) and the Cit/ α-KG pool is elevated in acclimated plants compared to non-acclimated plants. This is even more pronounced in the *hpr1-1* mutant. Because the mutant requires more Glu in order to transaminate glyoxylate, the higher turnover makes sense. In addition, the non-acclimated plants showed higher *de novo* assimilation of N. This demonstrated that N assimilation benefits from PR activity. Gauthier *et al.*, 2010, showed that the C skeletons for N assimilation originate from C assimilated during the previous day which is stored as Cit. In conclusion, PR regenerates α-KG (especially in the *hpr1-1* mutant) by incorporating N into glyoxylate whereby the assimilated N is stored in Gly and Ser. Reduced PR leads to a diminished N sink because the C skeletons for Gly production are lacking and less α-KG is recycled, which is necessary for *de novo* N assimilation. If PR is reduced over a long time interval, the N assimilation is negatively affected. Eventually, the increasing C/N ratio leads to the acclimation to eCO_2, as organic N compounds become limiting.

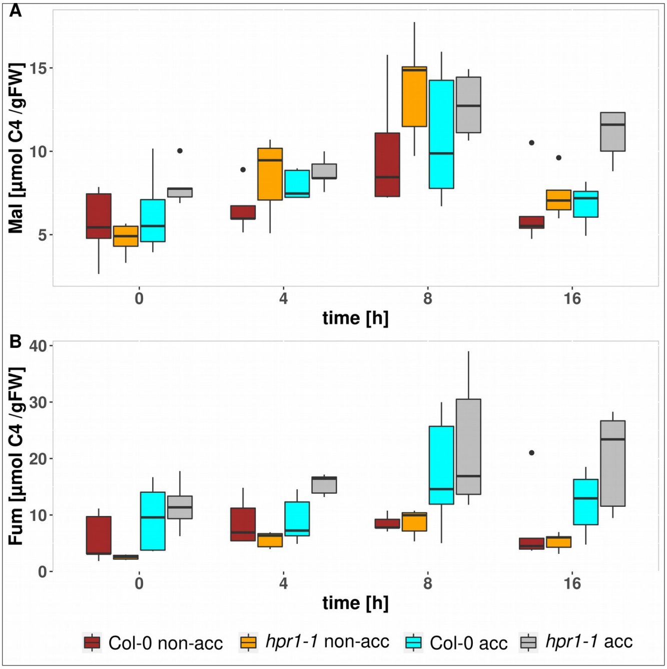


Figure 4: Boxplots of carbonic acids malate (A: Mal) and fumarate (B: Fum) (n = 5). Acclimated = acc. Non-acclimated = non-acc. Begin of light-phase = 0 h. End of light-phase is 8h. Data from Krämer *et al.*, 2022a.

The alternative oxidase at eCO$_2$

Plants can transfer the electrons stored in NADH to O$_2$ either by the cytochrome pathway or using the alternative oxidase (AOX). During electron transfer in the cytochrome pathway a pH gradient is created, which is used to produce ATP. The AOX pathway bypasses three of the four instances contributing to build-up of the pH gradient and, thus, substantially reduces the efficiency of ATP synthesis (Strodtkötter *et al.*, 2009). As the flux through GDC is high, it was proposed that AOX helps oxidizing NADH produced during the decarboxylation of Gly thereby preventing over-reduction of plastides as PR consumes reducing equivalents produced during PS (Zhang *et al.*, 2017; Sunil *et al.*, 2019). Inhibition of AOX leads to reduced PR as reported by (Zhang *et al.*, 2017). Initially the *aox1a* mutant was considered in Krämer *et al.*, 2022a, as this mutant is defective in AOX and therefore PR could be impaired. However, under the conditions used in the study *aox1a* plants showed no differences compared to Col-0 regarding metabolite and enzymatic data. For the PR intermedi-

ates Gly and Ser (Fig. 5 A) the mutant *aox1a* showed an only slightly significant increase for Gly (P = 0.0318). The ratio between Ser and Gly (Fig. 5 B) decreased during the day as the Gly concentration was increased in the plants. The lowest ratio was observed for non-acclimated Col-0 at the end of the light-phase, followed by *aox1a* acclimated and non-acclimated at the end of the light-phase. Therefore, the contribution of AOX in degrading NADH produced by GDC is marginal at 1000 ppm CO_2. In conclusion, AOX seems to be important only at photorespiratory conditions, e.g. high light and low CO_2 as reported by (Zhang *et al.*, 2017).
Sunil *et al.*, 2019, showed that the H_2O_2 generated by PR is a signal to up-regulate cyclic electron flow or AOX. Thus, the flux through PR partly regulates PS and therefore PR becomes an indicator for the overall PS activity. As H_2O_2 belongs to the reactive oxygen species (ROS), it is involved in abiotic stress responses (G. Miller *et al.*, 2010; Voss *et al.*, 2013). Results from the literature suggest that PR is involved in the adjustment of the redox status in plants coping with stress such as low temperatures or water stress (Cheng *et al.*, 2007; Voss *et al.*, 2013). Beyond that it is known that PR can protect PS from photoinhibition as it consumes reducing equivalents at least via GS.

Future work is necessary to examine the stress response of plants at non-photorespiratory conditions especially at eCO_2. It would be interesting how the interaction of eCO_2 with abiotic stress factors such as drought would affect plant metabolism.

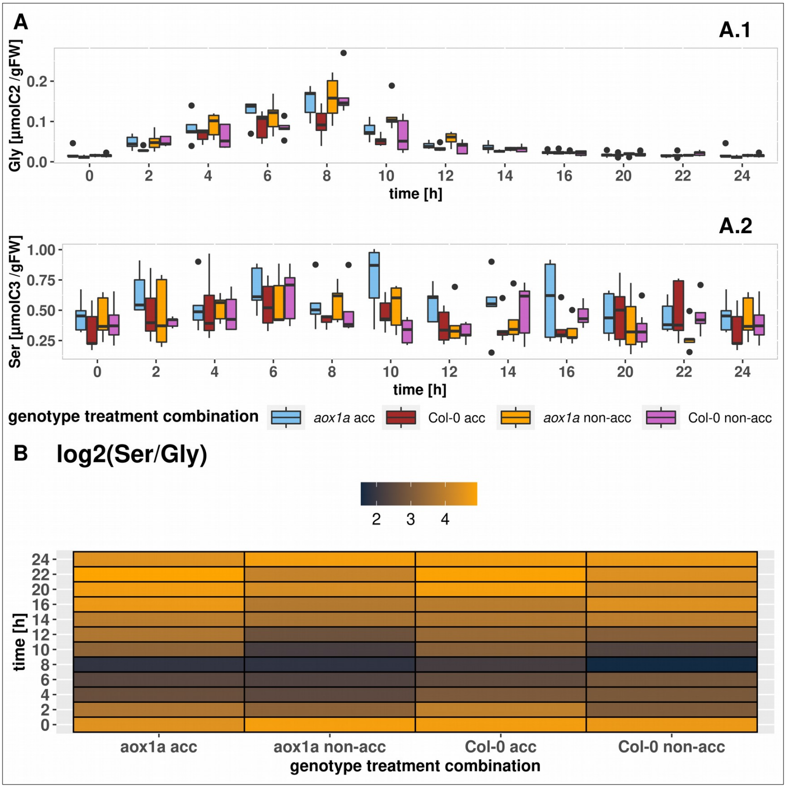


Figure 5: Comparison of acclimated (acc) and non-acclimated (non-acc) Col-0 and aox1a plants. Data for Col-0 from Krämer *et al.* 2022a. A: Photorespiratory intermediates (n = 5). A.1: Gly. A.2: Ser. B: log2 ratio of Ser to Gly using means of the data shown in A.

Cross-talk between PR and N assimilation

Non-acclimated *hpr1-1* plants (Krämer *et al.*, 2022a) showed a dwarfish and chlorotic phenotype at aCO_2 as already described in the literature by Timm *et al.*, 2008 and J. Li *et al.*, 2019. This phenotype rapidly changes, when plants are transferred to eCO_2. Hence, the plants suffer at aCO_2. However, it is not fully understood what limits growth of the mutant plants. For instance, metabolic analysis revealed that *hpr1-1* has more resources (e.g. amino-acids) when compared to Col-0 (Timm *et al.*, 2021). Considering the results in the literature, the question arose which mechanism causes the distinct phenotype at aCO_2.

Beyond that, Krämer *et al.*, 2022a, substantiated the link between PR and N assimilation. Therefore, the impact of PR at photorespiratory conditions (aCO_2) compared to non-photorespiratory conditions (eCO_2) was investigated in Krämer *et al.*, 2022b.

In this study, *Arabidopsis thaliana* plants (Col-0 and *hpr1-1*) were grown in hydroponic cultures with NO_3^- as sole N source. As a result, NH_4^+ originates either from PR or from the NR reaction. The obtained metabolic and enzymatic data-sets were used in simulations using the R package *paropt (Krämer et al., 2021).* The ODE-system of Krämer *et al.*, 2022a, was substantially extended. Until then only C-compounds were considered in dynamic models of primary metabolism of *Arabidopsis* thaliana (Nägele *et al.*, 2010; Küstner, Nägele and Heyer, 2019); albeit the total AA pool contains N, only the C content was of interest. Even though the ODE-system of Krämer *et al.*, 2022a, did include the PR pathway only C was balanced.

The extended ODE-system of Krämer *et al.*, 2022b, contains the PR pathway; yet the GDC/SHMT reaction is differently implemented. Instead of using one mass action reaction representing GDC and SHMT together, two different reactions were implemented. Previously, the assumption was made that due to the spatial proximity of the two reaction centers of GDC and SHMT only one reaction in the ODE-system was necessary (Douce *et al.*, 2001). However, Rebeille, Neuburger and Douce, 1994 showed that the GDC rate is higher compared to the SHMT rate in pea mitochondria. In addition, the model of Krämer *et al.*, 2022a, used the GDC/SHMT reaction to convert Gly to Ser, whereas the extended model of Krämer *et al.*, 2022b, includes NH_4^+ as state variable. Thus, the GDC and SHMT reactions have different tasks to accomplish in the model. In respect of the different structures of the ODE-systems and the results of Rebeille, Neuburger and Douce, 1994 two different reactions were implemented, both as Michaelis-Menten (MM) kinetics. GDC consumes one Gly, forming NH_4^+, and the SHMT reaction converts Gly to Ser. Notably, it was excluded that the SHMT flux could exceed the GDC flux, as the required CH_2-THF is produced by

GDC.

The boundaries for the export of Ser to BM were roughly estimated in the old model. In contrast, the new model links this reaction with the flux from AA to BM (AA2BM). The flux Ser to BM was implemented as AA2BM multiplied with the proportion of Ser in the proteom of *Arabidopsis thaliana*.

The main change in the new model is the inclusion of the GS/GOGAT cycle. For this reason the state variables Glu, Gln and α-KG were added. The *refixation* rate from the model described in Krämer *et al.*, 2022a, was replaced by the GS reaction. Therefore, the v_{max} values of the enzymes were measured. Moreover, a flux from Glu to α-KG was included given as the PR input into the system minus the HPR flux (PR – HPR); this was possible because both fluxes have the same unit '[µmol N/ (g*h)]' regarding N. The difference PR - HPR depicts the GGT reaction. For both enzymes, GS and GOGAT, MM kinetics were implemented according to Meek and Villafranca, 1980.

The enzymes involved in N assimilation show different activities over a full diurnal cycle due to regulation of the enzymes involved (Lillo, Meyer and Ruoff, 2001). Most importantly the GS reaction is activated by a reducing milieu (Choi, Kim and Kwon, 1999). During the assay the activity is measured in a reducing environment. Thus the values are potentially overrated particularly during the dark-phase (Berteli *et al.*, 1995; Gomes Silveira *et al.*, 2003). As the redox state in the plastids is unknown the impairment of GS during the dark-phase was estimated using computational experiments as described in Krämer *et al.*, 2022b. Thus, generating information about GS activity during the dark-phase, which cannot be obtained by enzyme assays.

Furthermore, a second supply of NH_4^+ was implemented, namely the NR reaction. The substrate NO_3^- was measured and the assumption was made that only a part of the NO_3^- can be used by NR, because it is located in the cytosol, while most of the NO_3^- resides in the vacuole. The cytosolic portion was calculated based on cell volumes determined by Koffler *et al.*, 2013.

The results of the flux analysis revealed that PR stimulated the turnover in the GS/GOGAT cycle. The plants at aCO_2 showed higher turnover compared to those at eCO_2 and *hpr1-1* at aCO_2 had the largest fluxes. Notably, a strong diurnal dynamic could be observed. In order to examine whether NH_4^+ originates from NR or PR, the ratio NR/PR was calculated for the light-phase. The results revealed that *hpr1-1* had to recycle more N primarily bound in Gly as compared to the wildtype. This was also found at eCO_2. Thus, even at 1000 ppm CO_2 PR operates; otherwise no difference for the ratio should be expected. Unsurprisingly, the ratio was elevated at eCO_2 compared to aCO_2 for Col-0 and the mutant. This means that the GS/GOGAT cycle has to turn over more NH_4^+ from PR at aCO_2.

Unexpectedly, no differences for NR could be observed between treatments, even though Bloom *et al.*, 2014, showed a decreased NO_3^- assimilation at eCO_2. Thus, the results support the findings of Andrews *et al.*, 2019 and Andrews *et al.*, 2020, who did not find reduced NO_3^- assimilation at eCO_2. Admittedly, the assumption that the NO_3^- concentration is linked to the volume of the cytosol could have led to the result of unaltered NR fluxes, because a constant input flux of NO_3^- into the cytosol is assumed.

Why does the *hpr1-1* mutant suffer at aCO_2

The metabolic pools measured for *hpr1-1* at aCO_2 (Krämer *et al.*, 2022b) prove that the plants have ample resources such as Cit, Mal, Fum and the AA pool. This is in contrast to the stunted growth and the chlorotic phenotype. Several hypotheses were put forward, which could explain the phenotype of the mutant.

The common opinion is that the mutant *hpr1-1* suffers from an intoxication by PR intermediates. Even though the GDC and SHMT reaction are increased in *hpr1-1* (Fig. 5 Krämer *et al.*, 2022b) the capacity is still too low to entirely remove toxic intermediates as known from the literature (ca. twice the amount of the wildtype).

(Timm *et al.*, 2008, 2011; Florian, Araújo and Fernie, 2013). It is noteworthy that genetically modified *Nicotiana tabacum* cv. Petite Havana plants, capable of converting glycolate to glycerate in the chloroplasts, showed increased biomass concentrations and a strong increase in glycolate, Gly and Ser levels (South *et al.*, 2019). Thus, high glycolate level do not necessarily explain the mutant phenotype.

Moreover, Li *et al.*, 2019, found in the *hpr1-1* mutant increased levels of 2-PG, which is an inhibitor for triose-phosphate isomerase (TPI) and causes an increase in glyceraldehyde 3-phosphate concentrations.

In this study, several other factors could be identified, which could contribute to the limitation of growth in *hpr1-1*.

Potentially, the mutant could suffer from elevated NH_4^+ concentrations. As can be seen in Fig. 6, NH_4^+ is increased in plants grown at aCO_2 compared to plants at eCO_2 ($P = 0.000315$ and $P = 1.15e-11$ for the genotype and treatment respectively). Remarkably, *hpr1-1* at aCO_2 showed the highest concentration. Whether this increase could harm the plants by deteriorating pH gradients is unclear. Another possibility by which NH_4^+ could harm plants would be the high amount of cell energy required for the re-fixation of NH_4^+. Thereby ATP and reduced ferredoxin could become limiting. It is known that the *hpr1-1* mutant has a diminished ATP/ADP ratio (Timm *et al.*, 2021), still, the decrease in the ratio is not very large.

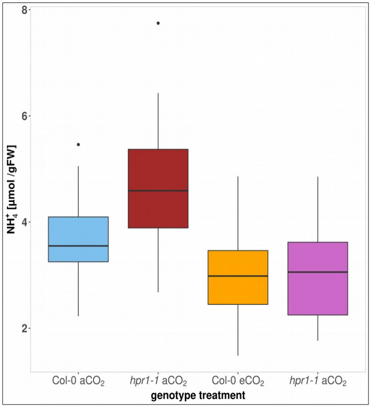


Figure 6: Boxplots of NH_4^+ concentration for Col-0 and hpr1-1 at aCO_2 and eCO_2 (Data from Krämer *et al.*, 2022b) (n = 40).

Additionally, the distribution of N and C between different metabolites is disturbed in the mutant. For instance, large amounts of Ser are found (Fig. 4 Krämer *et al.*, 2022b). Beyond that *hpr1-1* has high demands for Glu in order to convert HyPy. Thus, it is possible that Glu produced in the GS/GOGAT cycle is mainly used by GGT, which would result in a limitation of protein production because Glu is not available. Interestingly, it is hypothesized that the mitochondrial GS converts Glu and NH_4^+ from PR to Gln, which is then transported to plastids (Linka and Weber, 2005). Thus, given that the GDC shows elevated activity in *hpr1-1* at aCO_2 (Krämer *et al.*, 2022b), Glu could be highly demanded for the transport of NH_4^+, which would additionally limit the access to Glu for other processes. As *hpr1-1* has lower chlorophyll levels and Glu is one of the precursors for chlorophyll, the lack of Glu could explain the chlorotic phenotype (Li *et al.*, 2019).

Most likely, *hpr1-1* suffers from a combination of the proposed mechanisms, intoxication by PR intermediates, elevated NH_4^+ levels and a perturbed distribution of C and N due to the high level of Ser and the high demand of Glu.

8 Outlook

The results of Krämer *et al.*, 2022a and Krämer *et al.*, 2022b, clearly demonstrate the importance of PR and its role in N assimilation beyond the importance of PR in the protection of PS from photoinhibiton and its role in the response to abiotic stress (Voss *et al.*, 2013) The results also showed that PR acts as N sink during the day. In addition, PR stimulates the turnover in the GS/GOGAT cycle. As it is known that C assimilated during the previous day is used for N assimilation (Gauthier *et al.*, 2010) PR activity is crucial to regain α-KG for N assimilation.

However, the question by which mechanism eCO_2 affects N assimilation in a negative way is still open. In order to address this issue it would be necessary to establish a mechanistic model which includes different cell compartments. In such a model, the NO_3^- concentration could be assessed more realistically, leading to better results for the NR flux. Moreover, it would be interesting to know in which compartment the PR intermediates are stored, as this would change the substrate amount for the participating enzymes in the mechanistic model.

Results in form of publications

- Krämer et al. 2022a: Acclimation to elevated CO_2 affects the C/N balance by reducing de novo N-assimilation
- Krämer et al. 2022b: Interaction of Nitrate Assimilation and Photorespiration at Elevated CO_2
- Krämer et al. 2021: R package paropt
- Krämer et al. 2022c: R package ast2ast

Received: 31 August 2021 | Revised: 13 December 2021 | Accepted: 15 December 2021
DOI: 10.1111/ppl.13615

SPECIAL ISSUE ARTICLE

Physiologia Plantarum

Acclimation to elevated CO_2 affects the C/N balance by reducing de novo N-assimilation

Konrad Krämer | Gabi Kepp | Judith Brock | Simon Stutz | Arnd G. Heyer

Institute of Biomaterials and Biomolecular Systems, Department of Plant Biotechnology, University of Stuttgart, Stuttgart, Germany

Correspondence
Arnd G. Heyer, Institute of Biomaterials and Biomolecular Systems, Department of Plant Biotechnology, University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany.
Email: arnd.heyer@bio.uni-stuttgart.de

Funding information
Landesgraduiertenförderung Baden-Württemberg

Edited by Y. Gibon

Abstract

Plants exposed to elevated atmospheric CO_2 concentrations show an increased photosynthetic activity. However, after prolonged exposure, the activity declines. This acclimation to elevated CO_2 is accompanied by a rise in the carbon-to-nitrogen ratio of the biomass. Hence, increased sugar accumulation and sequential downregulation of photosynthetic genes, as well as nitrogen depletion and reduced protein content, have been hypothesized as the cause of low photosynthetic performance. However, the reason for reduced nitrogen content in plants at high CO_2 is unclear. Here, we show that reduced photorespiration at increased CO_2-to-O_2 ratio leads to reduced *de novo* assimilation of nitrate, thus shifting the C/N balance. Metabolic modeling of acclimated and non-acclimated plants revealed the photorespiratory pathway to function as a sink for already assimilated nitrogen during the light period, providing carbon skeletons for *de novo* assimilation. At high CO_2, low photorespiratory activity resulted in diminished nitrogen assimilation and eventually resulted in reduced carbon assimilation. For the *hpr1-1* mutant, defective in reduction of hydroxy-pyruvate, metabolic simulations show that turnover of photorespiratory metabolites is expanded into the night. Comparison of simulations for *hpr1-1* with those for the wild type allowed investigating the effect of a perturbed photorespiration on N-assimilation.

1 | INTRODUCTION

Rising emissions have led to a dramatic increase of atmospheric CO_2 concentration during the last decade (IPCC, 2014; Solomon et al., 2009), and despite intensive research, for example, on consequences for productivity at the ecosystem level (Liu et al., 2018; Long et al., 2004), many questions regarding the effects of elevated CO_2 (eCO_2) on plant metabolism are still unanswered (Misra & Chen, 2015). Plants benefit from eCO_2 concentrations by an increased net photosynthesis (NPS) rate. However, after a period of several days to weeks of growth in eCO_2 a decline is observed, which is called acclimation to eCO_2 (Griffin & Seemann, 1996; Smith & Dukes, 2013). A multitude of studies have shown that various plant species show this response, including important crops (Gutiérrez et al., 2013; Shimono et al., 2010), and even though many theories have been put forward, the physiological basis for acclimation to eCO_2 is still unclear.

One possible explanation is down regulation of photosynthesis (PS) due to an accumulation of carbohydrates (Dabu et al., 2019; Gámez et al., 2020; Kizildeniz et al., 2021). A clear increase of glucose, fructose, and starch was observed in *Arabidopsis thaliana* when grown for longer periods at eCO_2 (Cheng et al., 1998). High carbohydrate levels correspond to a dilution of nitrogen (N) containing compounds such as amino acids (AA) and protein, and the resulting low level of ribulose-1,5-bisphosphate-carboxylase/-oxygenase (RUBISCO) (Chen et al., 2005; Cheng et al., 1998; Paul &

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2021 The Authors. Physiologia Plantarum published by John Wiley & Sons Ltd on behalf of Scandinavian Plant Physiology Society.

Driscoll, 1997) could also explain reduced PS per weight unit as proposed by Gifford et al. (2000), Kuehny et al. (1991), and Wong (1990). Other concepts explaining acclimation to eCO_2 are directly related to N acquisition. Decreased stomatal conductance resulting from higher internal CO_2 concentrations in leaves may cause low transpiration rates, thus interfering with mineral uptake and transport (Ainsworth & Rogers, 2007; Shi et al., 2021). Nitrogen partitioning within the plant and its availability to photosynthetically active leaves was shown to contribute to acclimation in soybean (Kanemoto et al., 2009), and its allocation to either biomass or optimization of the photosynthetic apparatus is important for crop yield (Biernath et al., 2013).

Because assimilation of N in the form of AA requires carbon skeletons, the provision of α-ketoglutarate (α-KG) takes a central role, thus tightly linking the mitochondrial citrate cycle to *de novo* N fixation. Considering that glycolysis and day respiratory CO_2 release are, at least to some extent, inhibited by light, Gauthier et al. (2010) questioned the origin of α-KG from recently assimilated carbon and reported that remobilization of existing carbon pools is important for α-KG provision. However, an alternative route to C-skeletons for AA synthesis via malate produced in the chloroplast in the so-called pyruvate branch of the photorespiratory pathway has been proposed (Bloom et al., 2020).

A striking feature of eCO_2 is a reduction of the oxygenase activity of RUBISCO, which is the entry point into photorespiratory metabolism (Keys, 2006). This metabolic route links carbon (C) and N metabolism, most obviously by producing the AA glycine (Gly), but has an impact on various other metabolic pathways, including sulfur assimilation and secondary metabolism (Abadie et al., 2021; Timm et al., 2021). Recently, it was shown that in C3, but not in C4, plants eCO_2 improved the use of ammonium as a nitrogen source (Wang et al., 2020). However, interactions between nitrogen metabolism, PS and photorespiration (PR) are not yet well understood. Liang et al. (2021) proposed that in rice, which is generally believed to oversize its photosystem antenna, PR has a function of stabilizing the C/N balance by consuming photosynthetic products and providing AAs for nitrogen metabolism, thus preventing a continuous increase in the C/N ratio, which would induce premature senescence and yield reduction.

To investigate whether PR functions as a gatekeeper between carbon and nitrogen assimilation, we set up a mathematical model that allows simulation of fluxes through PR and PS, as well as calculation of *de novo* N fixation. We used this model to simulate fluxes in plants exposed to either short-term or prolonged exposure to eCO_2. Under the latter conditions, photosynthetic acclimation caused a marked reduction in NPS. This was more pronounced in the *hpr1-1* mutant of *A. thaliana*, which is defective in the peroxisomal hydroxypyruvate-reductase (HPR). This enzyme catalyzes the last step in the PR pathway providing glycerate for re-import into the chloroplast (Timm et al., 2008). Instead, the mutant accumulates high levels of serine (Ser) and Gly, thus altering the balance between C and N metabolism.

2 | MATERIALS AND METHODS

2.1 | Plant materials and growth conditions

Arabidopsis thaliana wild-type Col-0 and the mutant *hpr1-1* (SALK067724) were grown in soil (seedling substrate, Klasmann-Deilmann GmbH) in a growth chamber with 8/16 h light/dark regime (100 μmol m^{-2} s^{-1}; 22/16°C). The *hpr1-1* mutant was obtained from the group of Hermann Bauwe at the University of Rostock; for details, see Timm et al. (2008). All plants were grown at ambient CO_2 concentrations (450 ± 20 ppm) for the first week. Afterward, half of them were transferred to eCO_2 (1000 ± 20 ppm), whereas the other plants were transferred after 44 days. Plants were harvested after 46 days of cultivation. Plants that grew longer at eCO_2 concentrations are denominated acclimated, while those that spent only 48 h at eCO_2 are termed non-acclimated. To exclude the possibility that the plants differ in developmental states, leaves were counted of 10 plants per condition. A Fisher's exact test resulted in a *p*-value of 0.9889. Thus, there was no difference in the developmental stage of the plants. From both sets of plants, samples were taken every 2 h over a full diurnal cycle, with the time point at the end of the night harvested only once. This time point was used as the start and end of the simulations.

To examine metabolic changes caused by shifting plants from ambient CO_2 concentrations to eCO_2, Col-0, and the *hpr1-1* mutant were grown at ambient CO_2 levels and harvested at the beginning and end of light phase.

2.2 | Gas-exchange measurements and calculation of PR

The CO_2 uptake by the plants was measured as already described by Nägele et al. (2010). An entire rosette was measured over 24 h in a growth chamber with 8/16 h light/dark regime (100 μmol m^{-2} s^{-1}; 22/16°C, 65% RH, 1000 ppm CO_2). Dark respiration was calculated as a mean over the entire dark phase. While NPS was measured as CO_2 exchange of whole plant rosettes with the atmosphere, the input into PR was calculated according to Sharkey (1988) (Equations (1) and (2)). For the CO_2 concentration at the carboxylation site, 600 ppm were used.

$$v_0 = (A + R_d)/(\phi^{-1} - 0.5) \tag{1}$$

where v_0 is the oxygenation rate, A is the photosynthetic rate, R_d is the respiration during the night, and ϕ is the ratio between oxygenation and carboxylation of RUBISCO. R_d is the mean value of the entire night without the first 5 min of the dark phase.

$$\Phi = (2\tau*)/[CO_2] \tag{2}$$

where ϕ is the ratio between oxygenation and carboxylation, τ^* is the CO_2 compensation point in the absence of dark respiration and $[CO_2]$

is the CO_2 concentration at the carboxylation site. τ^* was measured according to the method described by Brooks and Farquhar (1985).

2.3 | Metabolic profiling

Gas-chromatography coupled to mass spectrometry (GC–MS)/MS analysis samples were extracted using 750 μl methanol with 25 nmol ribitol as internal standard. After adding methanol, samples were incubated for 15 min at 70°C and agitated for 10 min at RT followed by centrifugation (5 min 17,000*g*). The supernatant was transferred to a new vessel. Next, 400 μl H_2O was added to the pellet, incubated at 95°C for 10 min followed by shaking for 10 min at RT. Then, samples were centrifuged (5 min 17,000*g*) and the supernatant pooled with solution from the previous step. Afterward, 300 μl H_2O and 200 μl chloroform were added to the suspension. After centrifugation (2 min 17,000g), the two phases were separated. Only the polar phase was used for analysis and dried in a speedvac. The dried samples were derivatized using 20 μl of methoxamine dissolved in pyridine (40 mg ml^{-1}) by incubation for 90 min at 30°C. Afterward, 80 μl *N*-methyl-*N*-(trimethylsilyl)trifluoroacetamide were added and the mixture incubated for 30 min at 50°C. The metabolites were measured using GC–MS/MS. For injection, 1 μl of the derivatized sample was used. The GC–MS/MS device was a GCMS-TQ8040 (Shimadzu). Helium was used as carrier gas with a column flow of 1.12 ml min^{-1}, and for the stationary phase, a 30 m Optima 5MS-0.25 μm fused silica capillary column was used. The injection temperature was 230°C. The transfer line and ion source were set to 250 and 200°C, respectively. The initial temperature of the column oven was 80°C and this was increased by 15°C per min until the final temperature of 330°C was reached, which was held for 6 min. After a solvent delay of 4.6 min, spectra of the MS device were recorded in the multiple reaction mode with specific target-ions for each metabolite. External standards were used for quantification. Besides metabolic profiling via GC–MS/MS analysis, additional metabolites, i.e. starch, hexose phosphates (HP), carboxylic acids (fumaric acid, malic acid, and citric acid) as well as minerals (nitrate, phosphate, and sulfate) and the total AA pool, were quantified by HPLC as described previously (Küstner et al., 2019). Levels of malate and fumarate (MF) were summed up as MF pool. For the carboxylic acids and minerals, quantification was performed at the beginning, the middle and the end of the light phase—as well as the middle of the night. The HP were measured at every second time point. Data for the remaining time points were obtained by spline interpolation.

2.4 | Enzyme activities

The HPR activity was determined according to Bauwe (2017). The activities of the enzymes sucrose-phosphate-synthase (SPS), glucokinase, and fructokinase were measured according to Küstner et al. (2019). All enzyme activities were determined at the beginning, middle and end of the light phase, as well as in the middle of the night. Values for the remaining time points were calculated by spline interpolation.

2.5 | Simulations

For simulations, the Github-version of the R-package "paropt" was used (Krämer et al., 2020) and https://github.com/Konrad1991/paropt). During each simulation, the system of ordinary-differential-equations (ODE) is repeatedly solved with different parameter sets. Hence, for each parameter set, a unique state solutions is produced. This *in silico* solution is compared to the measured states to yield an error which is used to evaluate the parameters. Based on the errors of each set, the optimizer updates the parameter sets. For optimization, the particle swarm optimization algorithm was used; for details of implementation, see (Akman et al., 2018; Krämer et al., 2020; Sengupta et al., 2018). The package uses the SUNDIALS Software to solve the ODE-system (Hindmarsh et al., 2005). Because of its superiority in solving stiff ODE-systems, the backward-differential-equation solver was used (Hindmarsh et al., 2005) with relative tolerance set to 1e-6 and absolute tolerances set for each state to 1e-8. For each run, a different seed was set for random number generation. The code of the ODE-system can be found in the supplement.

The *on/off* mode of illumination in the plant growth chamber caused abrupt transitions between day and night leading to overshooting of changes in reaction rates (Küstner et al., 2019). To allow calculation of a smooth day-night-transition, an interval from 0.1 h before and 0.1 h after start of the dark period was defined. Within this interval, supporting data points for the ODE-solver were generated for PS and PR. To achieve this, the property of the tangens hyperbolicus to show increasing values of a sigmoid form from 0 to 1 for the interval [0,2] was used (Equation (3)). This calculation is based on the approach of Fenton and Karma (1998), who used it to produce a good fit of curves for restitution of action potentials. Application of this method largely prevented overshooting or undershooting at the day-to-night transition. However, simulations of metabolite dynamics using one complete parameter set for day and night still yielded insufficient fit of measured and calculated metabolite levels. This resulted from a characteristic of the Catmull–Rom spline used for interpolations. This spline is a cubic hermite spline that uses two points preceding and two points following a given time point to calculate the value for that time point. Thus, values at the end of the day have a strong impact on the night and vice versa. To eliminate this mathematical artifact, it was necessary to use separate parameter sets for day and night, for which parameters were identified independently.

$$\mathit{SupportingPoint} = \tanh\left(\left(2/\mathit{TimeIntervalLength}\right)\mathit{CurrentTime}\right)\ \mathit{DiffDayNight} \qquad (3)$$

where *SupportingPoint* is the PS or PR value at the current time point within the interval, *TimeIntervalLength* = 0.2 h, *CurrentTime* is the time point within the interval 0.1 h before and 0.1 h after start of the dark period minus the starting point of the interval, and *DiffDayNight* is the ΔPS or ΔPR between day and night.

2.6 | Data analysis and statistics

Data evaluation, normalization, visualization and statistics were performed in Microsoft Excel (Microsoft Office version 2010) and the R software (The R Project for Statistical Computing; http://www.r-project.org/).

3 | RESULTS

3.1 | Acclimation to eCO_2

As a first step toward understanding the metabolic processes underlying plant acclimation to eCO_2, conditions were established that allowed verification of eCO_2 acclimation by gas exchange measurements. Plants were grown at 1000 ppm CO_2 for 6 weeks, and during the seventh week NPS rates were compared to those of control plants grown at ambient CO_2 (450 ppm) for 6 weeks and shifted to eCO_2 only 48 h ago. The latter treatment ensured that plants were not in a transitional state between ambient and eCO_2, but did also not yet photosynthetically acclimate to eCO_2. As shown in Table 1, plants acclimated to eCO_2 had diminished NPS. The non-acclimated wild type, Col-0, had an average NPS rate of 132.04 $\mu molg^{-1}$ FW h^{-1}, while acclimated plants showed gas exchange rates of only 90.98 $\mu molg^{-1}$ FW h^{-1}. For the *hpr1-1* mutant, acclimation resulted in photosynthetical activity of 104.96 $\mu molg^{-1}$ FW h^{-1}, and non-acclimated plants showed an activity of 152.78 $\mu molg^{-1}$ FW h^{-1}. While two-way-ANOVA for genotype and treatment revealed a clear effect of acclimation ($p = 0.00491$), NPS of the mutant was not significantly different from wild type under eCO_2. Nevertheless, the *hpr1-1* mutant had a higher respiration than the wild type, indicating higher metabolic activity during the dark phase (see Table S1).

Acclimation to eCO_2 correlated with several changes in primary metabolites (Figure S1). Among soluble sugars (Figure S2), hexoses displayed the strongest effect of acclimation to eCO_2, while sucrose (Suc) remained fairly constant. Fructose (Frc) and glucose (Glc) levels during the light phase declined by trend in acclimated wild type but significantly rose in acclimated *hpr1-1* plants (ANOVA for treatment × genotype, $F_{3,76} = 11.82$, $p = 0.0011$). Thus, the higher Frc concentration of non-acclimated wild type as compared to mutant plants was lost during acclimation. Starch content at the end of the day was not altered in acclimated plants, but was higher in *hpr1-1* as compared to Col-0 (Figure S3). MF accumulated during the light phase and declined in the dark, creating a clear diurnal profile in both genotypes that was more distinct in acclimated plants (Figure 1). These two metabolites were pooled based on the similar behavior reported by Küstner et al. (2019). Although absolute levels sank during acclimation in Col-0 ($p = 0.0435$), diurnal dynamics persisted but declined in the mutant. Citrate (Cit) displayed a diurnal pattern opposite to MF in non-acclimated plants with strongly elevated levels in *hpr1-1* ($p = 9.19e-11$). The diurnal oscillation was completely lost in acclimated wild type but still visible in the mutant. The non-acclimated *hpr1-1* mutant formed a clearly separated cluster in principal component analysis (PCA) due to accumulation of compounds involved in N-assimilation, such as total AA, nitrate and α-KG (Figure S4). Total AA increased during the day in non-acclimated plants ($r = 0.376$, $p = 0.0002$) and reached very high levels in *hpr1-1* (Figure 2C). Acclimation caused a significant reduction in free AA only in the mutant ($F_{3,284} = 25.8$, $p = 8.4e-15$). As with Cit, diurnal dynamics decayed in acclimated wild type, but not in the mutant. As shown in Figure 2A,B, Gly and Ser, which were separated from the total AA pool, massively accumulated during the day in *hpr1-1*, but especially Ser levels remained constant after acclimation. Both genotypes had lower nitrate concentrations under eCO_2 ($p = 3.05e-7$, Figure 2D).

For modeling, enzyme activities of SPS, the hexokinases (fructokinase and glucokinase) and the HPR were considered key enzymes in the pathways for sucrose build-up, sucrose cycling and the photorespiratory salvage pathway, respectively. In the case of HPR, this was done, because this activity discriminates the *hpr1-1* mutant from the wild type. *In vitro* maximum turnover rates were determined to obtain parameter boundaries for the values used in modeling Michaelis–Menten kinetics (Figure S5). Fructokinase showed an increased activity in the non-acclimated *hpr1-1* mutant ($p = 3.77e-05$). Even though no significant effect could be demonstrated, there was also a tendency to an increased glucokinase activity. The wild type displayed a clearly increased SPS activity for acclimated plants ($p = 0.00977$). This was not the case for *hpr1-1*. Interestingly, no effect except the expected genotype effect was observed for HPR activity.

TABLE 1 Means and standard deviation (SD) of net photosynthesis in $\mu mol/gFW^{-1}\ h^{-1}$ during light phase for Col-0 and hpr1-1 in either non-acclimated or acclimated state. Measurements were carried out at elevated CO_2 ($n = 12$, 10, 11, and 12)

Genotype	Acclimation	Mean	SD
Col-0	Non-acclimated	132.04	30.94
hpr1-1	Non-acclimated	152.78	37.70
Col-0	Acclimated	90.98	35.22
hpr1-1	Acclimated	104.96	42.74

3.2 | Metabolic changes caused by transfer from ambient CO_2 to eCO_2

To distinguish between the effect of eCO_2 treatment and the acclimation to eCO_2, a set of plants grown at ambient CO_2 was analyzed and compared to plants exposed to eCO_2 for 48 h (Table S2). For the photorespiratory intermediates Gly and Ser a strong decrease was observed following the shift to eCO_2. Thus, photorespiratory intermediates that build up at ambient CO_2 did not persist in non-acclimated plants, indicating formation of a new homeostasis. In addition,

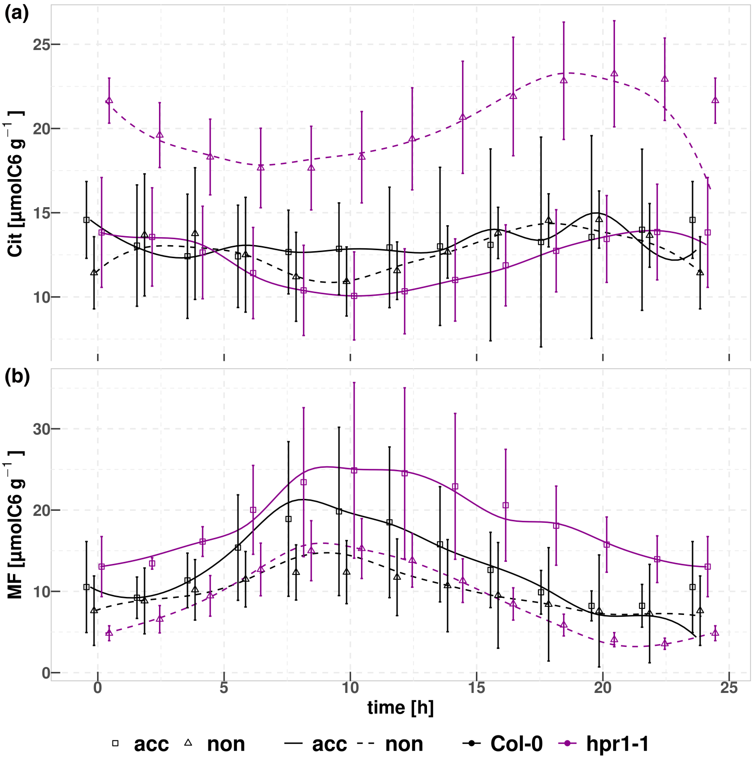


FIGURE 1 Measured values and results of simulations for citrate (Cit) (A) and the malate/fumarate (MF) pool (B). Lines represent the means of 20 simulations. Means of measured values ± standard deviation ($n = 5$) are shown as dots. Light period is from 0 to 8 h. To prevent overlapping, error bars were displaced by 0.25 h

carbonic acids were measured at ambient CO_2 concentrations and plants shifted to eCO_2. For the MF pool similar diurnal dynamics were detected at ambient and eCO_2, although, for the latter at lower absolute concentrations. In contrast, Cit was strongly affected by the eCO_2 treatment. As can also be seen in Figure 1, the Cit pool showed almost no diurnal dynamics under eCO_2 in wild type plants, but was highly dynamic at ambient CO_2 (Table S2). The *hpr1-1* mutant had higher levels than wild type of Cit under ambient as well as eCO_2, but in this case diurnal dynamics persisted in non-acclimated plants and declined only after acclimation (Figure 1).

3.3 | Model construction

For simulating the impact of eCO_2 on interaction of carbon and nitrogen metabolism, a model was set up as described in Figure 3. The exchange of CO_2 with the environment covers the sum of three processes: photosynthetic fixation, PR and mitochondrial respiration. Although mitochondrial respiration can only be measured during the dark phase, it was set as a constant rate over the entire diurnal cycle, because Küstner et al. (2019) have demonstrated that this improves modeling of diurnal metabolite dynamics. Thus, the amount of mitochondrial respiration was added to measured carbon uptake to calculate the flux of primary fixation. This is termed adjusted PS. Like mitochondrial respiration, PR is a process that releases CO_2, thus reducing net carbon uptake. During the light phase, it must therefore also be added to the measured carbon uptake in order to calculate the rate of primary fixation. The adjusted primary fixation was used as input term to form HP, which are connected to the carbohydrates (Suc, Glc, and Frc) in the sucrose-cycling pathway (Küstner et al., 2019; Nägele et al., 2010). In addition, HP were considered as substrate for the synthesis of MF which functioned as a precursor for the Cit/α-KG pool. Cit and α-KG were modeled as one pool, because α-KG is

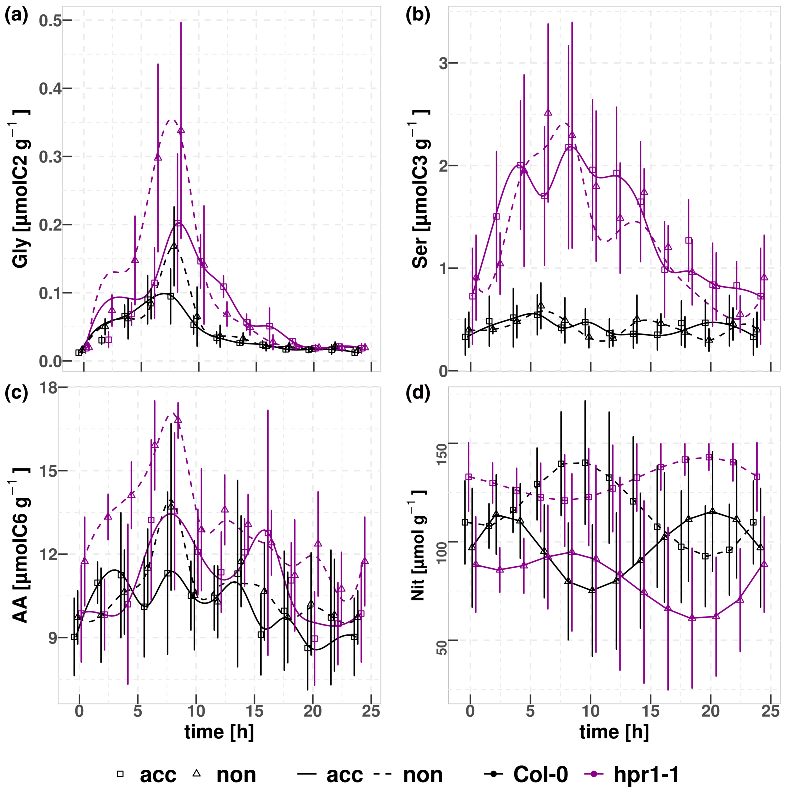


FIGURE 2 Measured values and results of simulations for glycine (Gly) (A), serine (Ser) (B), total amino-acids (AA) (C) and nitrate (Nit) (D). (A–C) Lines represent the means of 20 simulations. Means of measured values ± standard deviation ($n = 5$) are shown as dots. Light period is from 0 to 8 h. To prevent overlapping, error bars were displaced by 0.25 h

mostly generated via isocitrate dehydrogenases (Tcherkez et al., 2017), while transamination reactions, e.g. using glutamate and oxaloacetate for α-KG production, do not contribute significantly (Hodges, 2002). Abadie et al. (2017) have shown that citrate, isocitrate, and α-KG together form the source for Glu production, no matter whether they stem from "old" or recently fixed carbon.

It has been demonstrated that during the light phase α-KG production relies mostly on stored citrate (Tcherkez et al., 2009). However, various modes of the TCA cycle and anaplerotic reactions have been proposed (Hanning & Heldt, 1993; Sweetlove et al., 2010), and thus a connection of the MF and Cit/α-KG pool was allowed also in the light, when phosphoenolpyruvate is used to form oxaloacetate. Phosphoenolpyruvate in our model is part of the HP pool that contains all short-lived intermediates. The MF pool served as substrate for dark respiration. Although this is a simplification of the TCA cycle, it allowed for dissection of Cit/α-KG dynamics, connected to AA metabolism, and energy household.

3.3.1 | PR and C/N interaction at eCO_2

PR activity was calculated according to Sharkey (1988) (Equations (1) and (2)). Values of the CO_2 compensation point (τ^*) for wild type and *hpr1-1* were measured according to Brooks and Farquhar (1985), resulting in a value of 50 ± 10 ppm for Col-0 and 92.5 ± 17.6 ppm for the *hpr1-1* mutant. No significant treatment effect was obtained in our measurements. Because the CO_2 concentration at the carboxylation site is unknown, PR rates were calculated assuming in internal CO_2 concentration of 600 ppm, following the suggestion of Sharkey (1988). A parameter α was introduced into the rate equation

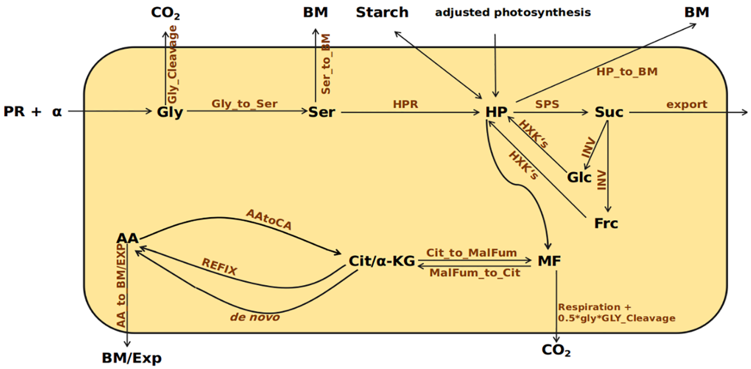


FIGURE 3 Schematic representation of the ODE-model used for simulations. States within the system-boundaries: glycine (Gly), serine (Ser), amino acids (AA), hexosephosphates (HP), sucrose (Suc), glucose (Glc), fructose (Frc), citrate (Cit), α-Ketoglutarate (α-KG), malate, and fumarate (MF). States outside of the system: biomass (BM), starch (Starch), and export (Exp). Constant fluxes across the system boundaries: net photosynthesis (adjusted photosynthesis), photorespiration (PR). A Michaelis–Menten kinetic was used for sucrose-phosphate-synthase (SPS), hexokinases (HXK's = fructokinase and glucokinase), invertase (INV), and the hydroxypyruvate-reductase (HPR). The remaining statements are modeled as mass action kinetics

for PR in order to allow flux through the salvage pathway in the absence of NPS, i.e. during the dark phase. Boundaries for α were set between 0 and 5 for all simulations. Inclusion of α substantially improved simulation of dynamics for Gly and Ser during the night, especially for the *hpr1-1* mutant, where metabolism of both AAs was distinctly extended into the dark phase (Figure 2A,B). A striking feature of the *hpr1-1* mutant was an altered ratio of PS to dark respiration; the latter being strongly elevated (Table S1). According to Sharkey (1988), this results in calculating a higher rate of PR, which, due to the mutant phenotype, causes accumulation of Gly and Ser, because PR flux is reduced in *hpr1-1*. This was confirmed by the metabolite data (Figure 2) and caused a higher value of α during the dark phase in this mutant.

It should be emphasized that the model shown in Figure 3 is a pure carbon metabolism model, which, however, allows for calculation of N fluxes at the interconversions of carboxylates and AA. While PR provides the carbon contained in the Gly pool, the nitrogen is donated by either Glu or Ser. Therefore, the HPR rate, which releases N, was subtracted from PR in order to take account of the N delivered by Ser. The remaining N comes from Glu, which in our model is contained in the AA pool. The corresponding rate from AA to Cit/α-KG is called AAtoCA (Figure 3 and Equation (4)). The stoichiometric factor 0.813 takes into account that the AA pool has 1.23 μmol N per μmol C6.

$$AAtoCA = [(PR + \alpha) - HPR]\, 0.813\, AA \tag{4}$$

where *AAtoCA* is the flux from AA pool into the Cit/α-KG pool, *AA* is the current AA concentration, *PR* is the PR, *α* is the factor described above, and *HPR* is the current value of the HPR activity.

During the reaction from Gly to Ser, one N is released in the form of ammonia. This has to be re-fixed. The carbon skeleton is α-KG which is represented by the Cit/α-KG pool. Therefore, a reaction called REFIX was introduced into the model (Figure 3 and Equation (5)). REFIX represents the rate of Cit/α-KG to AA conversion and is proportional to the rate from Gly to Ser. The factor 4.878 is the C/N ratio of the AA pool and is used to transform the N1 flux into a C6 flux. Note that the parameters gly_cleavage and gly_to_ser were set to the same value in order to account for the spatial proximity of glycine cleavage enzyme and the serine-hydroxymethyl-transferase (Douce et al., 2001).

$$REFIX = gly\; gly\_cleavage\; 4.878 \tag{5}$$

where *REFIX* is the flux from Cit/α-KG pool into the AA pool, *gly* is the current Gly concentration and *gly_cleavage* is the rate constant. Thus, only the parameters for de novo N-assimilation needed to be optimized according to the dynamics of AA and Cit/α-KG cycling.

As can be seen from Equations (4) and (5), the fluxes between AA and the Cit/α-KG pool that arise from PR are only depending on Ser and Gly concentrations. Besides these, a flux from the Cit/α-KG pool into the AA pool, which is independent of PR activity but relies on the α-KG concentration as substrate, results from *de novo* N assimilation. Thus, by identifying the parameters for Equations (4) and (5), it is possible to differentiate between *refixation* of ammonia and *de novo* N-fixation.

3.4 | Simulated fluxes at eCO$_2$

Simulations of metabolite dynamics matched measured values with a maximal deviation of 8.2% per time point and state and remained

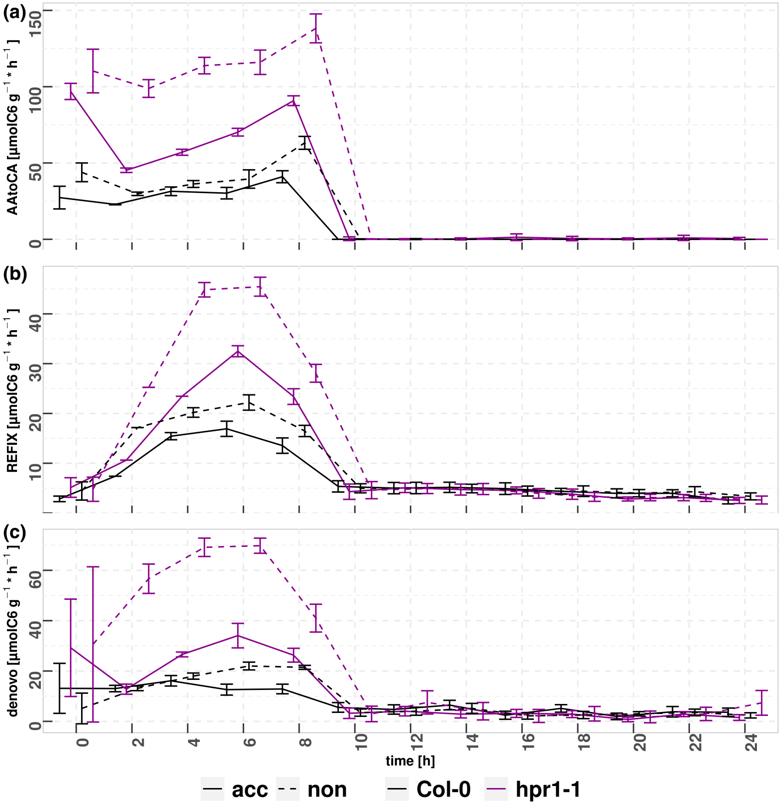


FIGURE 4 Flux calculations for interconversion of total amino acids and the Cit/α-KG pool. AAtoCA is the amino acid consumption for Gly synthesis (A). REFIX is the refixation of N produced by glycine-cleavage enzyme (B). Denovo is the de novo fixation of N (C). Error bars depict SD simulations ($n = 20$). Light period is from 0 to 8 h. All values are evaluated every 2 h. To prevent overlapping of error bars calculated for identical time points, bars were displaced by 0.25 h

within the standard deviation with few exceptions (Figures 1, 2, and S2). Although the default error calculated by the *paropt* algorithm is the sum of absolute differences between measured and simulated states, a relative error was calculated by the cost function in order to improve results especially for low concentrated metabolites Hence, it was possible to optimize very small (e.g. Gly) as well as large states (e.g. AA) at the same time with similar accuracy. Based on substrate concentrations and the identified parameters, flux rates at specific time-points could be calculated. Figure 4 and Table 2 show the calculated fluxes. As expected, fluxes in the AA-to-Cit/α-KG cycle, which are in part dependent on PR, are significantly increased for the non-acclimated plants (Table 2).

The rate of Gly accumulation in the light was about 1.5-fold higher in non-acclimated versus acclimated Col-0 and about doubled

TABLE 2 Means of 20 simulations for the fluxes, 6 h after light-on, in µmol C6 gFW^{-1} h^{-1}. Col-0 acclimated (C.acc), Col-0 non-acclimated (C.non), *hpr1-1* acclimated (h.acc), and *hpr1-1* non-acclimated (h.non). *AAtoCA* is the amino-acid consumption for synthesis of Gly. *REFIX* is the refixation of N produced by glycine-cleavage enzyme. *denovo* is the de novo fixation of N. Letters indicate the results of Tukey's HSD following two-way ANOVA for genotype × treatment. Groups sharing the same letters are not significantly different ($p \leq 0.05$; $n = 20$)

Fluxes	C.acc	C.non	h.acc	h.non
AAtoCA	30.22^{d}	39.49^{c}	70.19^{b}	116.00^{a}
REFIX	16.92^{d}	22.19^{c}	32.49^{b}	45.45^{a}
denovo	12.60^{d}	21.98^{c}	34.06^{b}	69.77^{a}

in non-acclimated *hpr1-1*. These higher rates, in turn, stimulated higher turnover of the AA-to-Cit/α-KG cycle. Simulations revealed that this supported a significantly increased *de novo* N assimilation using α-KG as substrate, while already fixed N was deposited as Gly and, at least in the mutant, also as Ser (Figure 2).

As a consequence of the mutation, HPR activity was strongly reduced in *hpr1-1* plants. Thus, the flux from Ser to HP was limited, which restricted the N flux from Ser to Gly. This in turn caused more AA to be used for Gly formation with the effect of elevated *de novo* N assimilation in the mutant (Figure 4 and Table 2).

4 | DISCUSSION

4.1 | Metabolic changes during acclimation to eCO$_2$

Three sets of plants were analyzed in this study—plants grown at ambient CO$_2$ concentration as well as plants either continuously grown at eCO$_2$ or shifted from ambient to eCO$_2$ for 48 h. Comparison of plants grown under ambient versus eCO$_2$ revealed that the most significant changes in the PR intermediates Gly and Ser completed within the first 2 days at eCO$_2$, while changes in Cit metabolism took longer to reach a new homeostasis. Continuous growth at eCO$_2$ resulted in a reduction of photosynthetic activity by about 20% in wild type and 30% in the *hpr1-1* mutant, thus clearly demonstrating acclimation of PS to eCO$_2$ within 44 days of exposure. Similar effects, expressed as reduced rate of RUBISCO carboxylation (V_{cmax}), have been reported by Jauregui et al. (2018) after 28 days at eCO$_2$ under long day conditions. In this study, the extent of acclimation coincided with the amount of foliar starch that appeared to negatively affect RUBISCO performance. Under the short day conditions applied in the present study, no difference in starch content was observed for acclimated and non-acclimated plants, arguing against a direct effect of the starch content. Interestingly, Jauregui et al. found dependence of acclimation on the source of N fertilization: while plants fertilized with nitrate did show acclimation, plants fertilized with a mixture of ammonium and nitrate retained higher photosynthetic rates under eCO$_2$ over extended periods of time (Jauregui et al., 2015), thus indicating an interaction of acclimation with N fixation. Because the most important source of ammonium during the light period is PR, we recorded diurnal profiles of PR intermediates under ambient and eCO$_2$ and found the typical diurnal profiles, i.e. a light-dependent accumulation of Gly and Ser in Col-0 and *hpr1-1* even at eCO$_2$. As already reported by Timm et al. (2008), amplitudes were increased in the *hpr1-1* mutant. This proved that PR still operates under the condition of a doubled atmospheric CO$_2$ concentration and is in agreement with data reported by Sage (1994). In various studies involving PR mutants, CO$_2$ concentrations of 3000 ppm and even higher were used in order to avoid oxygenation of RUBISCO (e.g. Timm et al., 2010). In the present study, 1000 ppm were applied to substantially reduce, but not eliminate, PR in both genotypes (Table S2 and Figure 2), because otherwise the impact of perturbed PR on acclimation to eCO$_2$ could not be investigated. With PR operating in both genotypes at a reduced rate in eCO$_2$ the *hpr1-1* mutant still showed increased Gly and Ser levels when compared to the wild type, and thus the effect of different PR fluxes could be examined. This revealed that diurnal dynamics of Cit persisted in non-acclimated *hpr1-1* plants, but dynamics completely declined in wild type and acclimated *hpr1-1*. The decreased diurnal dynamics of Cit reflect reduced consumption of α-KG for ammonia re-fixation under eCO$_2$, and this was strongly delayed in the *hpr1-1* mutant. Gauthier et al. (2010) showed that α-KG is produced from carbon that was stored during the previous night. The fact that the MF pool retained diurnal dynamics in both genotypes under eCO$_2$, although at reduced absolute levels (Table S2), while dynamics declined for Cit, is strong evidence that the MF and Cit pool are independent of each other. As demonstrated by Tcherkez et al. (2009), α-KG predominantly derives from "old" Cit, while malate production depends on phosphoenolpyruvate. However, the simulations presented here do not finally disclose at which mode the TCA cycle operates during the light phase (Sweetlove et al., 2010).

Even though PR was not stalled at 1000 ppm of CO$_2$, its contribution to gas exchange was substantially reduced. With the rate of the Gly accumulation in the light taken as an indicator of RUBISCO oxygenation, PR was reduced by about 35% in Col-0 and nearly 50% in the *hpr1-1* mutant. A central question addressed in this study was whether reduced PR, and the concomitant reduction in ammonium availability, could have caused photosynthetic acclimation as suggested by the data of Jauregui et al. (2015). Because PR and PS are interlinked (Sharkey, 1988), it is difficult to separate cause and effect. However, comparison of the *hpr1-1* mutant with Col-0 wild type revealed important details. If we assume that the ratio of carboxylation and oxygenation of RUBISCO is not affected by the *hpr1-1* mutation, the much stronger reduction of daily Gly accumulation after long-term exposure to eCO$_2$ in *hpr1-1* must be attributed to an altered flux through the PR salvage pathway. With this flux and an independent recording of photosynthetic activity, a ratio of PR to NPS can be calculated, which is higher in non-acclimated plants (Col-0 ca. 17% and *hpr1-1* ca. 26%) even though both sets of plants were analyzed in an atmosphere with identical CO$_2$ concentration. This implies a decline of PR flux relative to NPS during acclimation to eCO$_2$, and this decline was supported by our flux simulations that identified reduced PR fluxes in acclimated plants. There are two possible explanations for the higher PR flux in non-acclimated plants. Either the higher PS in non-acclimated plants caused a local depletion of CO$_2$ at the carboxylation site that could not be compensated by CO$_2$ diffusion within the leaf tissue, or metabolic and/or enzymatic alterations in acclimated plants limited the capacity for PR.

Sage (1994) reported that for internal CO$_2$ concentrations between 200 and about 550 ppm, PS is limited by the regeneration of ribulose-1,5-bisphosphate and, thus, responsive to changes in the CO$_2$-to-O$_2$ ratio, while for higher internal CO$_2$ (C_i) concentrations, PS becomes limited by P_i regeneration and is than unaffected by changes in C_i. Given that C_i is about 600 ppm under the conditions of eCO$_2$ applied here, local depletion of CO$_2$ at the carboxylation site becomes unlikely. In contrast, however, some of the metabolite changes observed during acclimation favor a metabolic limitation of PR. Acclimated Col-0 plants had a reduced ratio of sugars versus carboxylic acids during the light phase, indicating a metabolic shift toward TCA cycle intermediates, which is indicative of a C-limited metabolic state.

An increase in carboxylic acids, especially fumarate and citrate, and a concomitant decline in the sugar-to-carboxylate ratio during the light phase at eCO_2 was also observed by Watanabe et al. (2014) for Arabidopsis seedlings grown on artificial medium for 20 days. Seedlings showed a higher CO_2 efflux rate at eCO_2, resulting from the use of carboxylates instead of sugars for respiration. The authors calculated a higher cost for ATP production at eCO_2, which they related to higher energy demand for carbohydrate export to the roots (Watanabe et al., 2014). However, they also found higher rates of AA production that could have stimulated ATP consumption. This was not observed in our study, where the total AA pool (not containing Gly and Ser) declined and nitrate levels were lowered after prolonged exposure to eCO_2.

In acclimated soil grown plants, transamination of glyoxylate by glutamate could have become limiting due to the decrease in the AA pool. This would cause a decline in PR flux and, thus, a depletion of Calvin–Benson cycle intermediates, because the regeneration of glycerate from Ser would slow down. The apparent stimulation of flux into the TCA cycle could thus be considered a compensatory measure to stimulate AA synthesis or may simply reflect reduced buildup of sugars because of detracted export of triose-phosphates from the plastids that resulted from reduced replenishment by the PR pathway. This was especially obvious in *hpr1-1*, which, due to the mutation, has a lower PR capacity than Col-0.

This implies that the capacity of the PR and the PS activity are balanced in order to prevent accumulation of the toxic intermediates 2-PG, glycolate, and glyoxylate. While this is physiologically reasonable, the question remains, why PR flux is reduced in acclimated plants.

4.2 | N assimilation

To determine whether acclimation of PS and reduced PR are causally linked, we developed a mechanistic model and identified kinetic parameters that were used for flux calculation of C and N fixation (Figures 3 and 4). For simulations, the upper boundaries for *de novo* N fixation were set to lower values during the night. The physiological background for this is high demand for reducing equivalents that are generated during PS in the light period (Farré & Weise, 2012; Gibon et al., 2006). Simulations of metabolite dynamics over a full diurnal cycle revealed an impact of PR on *de novo* N assimilation. Remarkably, non-acclimated plants shifted to high CO_2 for not more than 48 h showed higher *de novo* N fixation as compared to acclimated plants (Figure 4). Hence, long-term exposure to high CO_2 reduced N acquisition, resulting in an increased C/N ratio. This has also been observed by Jauregui et al. (2016) and coincided with halving of the foliar ammonium content, while nitrate content remained unchanged.

While the pathways of PS, PR, and N fixation involve various subcellular compartments, compartmentalization was not included in the model, because metabolite measurements were not subcellulary resolved. Thus, the reported concentrations of Gly and Ser in whole cell extracts very likely underestimate the local concentrations in mitochondria and peroxisomes. In case of the local concentration were higher than disclosed here, fluxes would be larger, resulting in higher turnover between Cit and AA. While this would change the absolute fluxes, relations reported for genotypes and treatments would not be affected. Our simulations indicated that at eCO_2 refixation of ammonium released during Gly cleavage was low due to low PR activity (Figure 4 and Table 2). Thus, under eCO_2, only a small amount of N is deposited in the form of Gly and Ser, two AAs with very low carbon content. An important consequence of reduced Gly turnover in the peroxisome is a low rate of glutamate transamination and, therefore, release of α-KG becomes restricted (Bloom et al., 2020). Because α-KG is the prime carbon skeleton for *de novo* AA production, this could negatively affect *de novo* N acquisition, thus leading to an elevated C/N ratio. Hodges (2002) reported that provision of α-KG by mitochondrial isocitrate-dehydrogenase should be inhibited in the light due to the production of NADH. Alternatively, α-KG could be produced by aspartate transaminase, but this would not create a net flux of N into the AA pool. Thus, deposition of already fixed N in AAs with low C content could be essential for effective *de novo* N assimilation. This concept puts PR into the role of a storage pathway for assimilated N during the light phase, when, because of a non-cyclic TCA (Sweetlove et al., 2010), provision of carbon for AA synthesis is limited. Gauthier et al. (2010) pointed out that C stored during the previous night has to be re-mobilized in order to supply α-KG for *de novo* N assimilation. Our data substantiates this finding by demonstrating that N present within the α-KG family of AA can temporarily be stored in the PR pathway in order to increase α-KG availability for *de novo* N assimilation. This is also supported by the finding that the non-acclimated *hpr1-1* mutant, which stores large amounts of Gly and Ser, showed the highest *de novo* fixation of N and, in accordance with that, had elevated total AA content.

It should also be mentioned that eCO_2 may negatively affect N assimilation by additional mechanisms, such as the inhibited transport of nitrite into the plastids and inhibition of succinate dehydrogenase, thus lowering energy provision for N assimilation (Asensio et al., 2015). These effects cannot be addressed in the mechanistic model presented here and may potentially add to the inhibition of *de novo* N assimilation. However, Bloom et al. (2014) showed that the impact of eCO_2 on nitrate assimilation is the most important effect, thus emphasizing the importance of the provision of carbon skeletons for N assimilation and the storage of N in the PR pathway.

Besides the N storage function, reduced PR turnover could also affect malate availability in the cytosol. As proposed by Bloom et al. (2010) and Rachmilevitch et al. (2004), the cutback in ammonium release due to lower PR rates might cause decreased export of malate from the plastids that are exchanged against α-KG as part of the malate shuttle. Bloom et al. (1989) pointed to the high demand for electrons produced by PS in the assimilation of nitrate. Malate in the cytosol could be required to produce NADH as substrate for the nitrate-reductase. Long-term exposure to eCO_2 might therefore cause a shift in the C/N based on a shortage of NADH in the cytosol. In the *hpr1-1* mutant, however, large amounts of NADH are produced by the glycine-cleavage enzyme (Leegood et al., 1995), and this could relate to the high AA levels in this mutant. An unsolved question regarding PR as promoter of malate transport into the cytosol and its use for NADH production remains with its function also in the exchange against glutamate produced in the plastids (Nunes-Nesi et al., 2010; Renné et al., 2003). This transport is essential for providing glutamate for the transamination that converts glyoxylate into Gly. Thus, the malate transport hypothesis may not be able to fully explain the benefits of PR for N assimilation.

4.3 | Metabolic dysfunctions in the *hpr1-1* mutant at ambient CO_2 concentrations

The design of the current study involved plants exposed to eCO_2 either long-term or short-term to uncover mechanisms of eCO_2 acclimation. Thus, the metabolic effects of the *hpr1-1* mutation under ambient CO_2 are not immediately visible. However, considering literature data and metabolic features of non-acclimated *hpr1-1* plants, some peculiarities of the mutant became obvious. The *hpr1-1* mutant had high AA as well as starch and citrate levels, but nevertheless produced less biomass. Timm et al. (2010) and Li et al. (2019) described a chlorotic phenotype, and this was attributed to a possible intoxication by glycolate, glyoxylate or 2-phospho-glycolate (Dellero et al., 2016). After a 48 h shift to eCO_2, we observed reduced hexose levels and higher dark respiration in *hpr1-1* as compared to wild type, as well as increased hexokinase activity, indicating enhanced catabolism. This is supported by low MF levels and an accumulation of α-KG especially during the night. While it is very likely that this relates to processing of the high Gly and Ser stock that piled up under ambient CO_2 (Table S2), it also demonstrates a metabolic shift from sugar to carboxylate metabolism. This is accompanied by strongly elevated phosphate levels, which are inhibitory to SPS and cytosolic fructose-bisphosphate phosphatase via stimulation of fructose-6-phosphate-2-kinase. It could thus be envisaged that *hpr1-1* is limited in carbohydrate supply and, hence, in provision of cell wall material for growth. However, our flux calculations also point to an additional problem. Turnover of Gly was strongly expanded into the night in *hpr1-1* (Figure 2) and may thus fall into a period of low abundance of glutamine synthetase 2 (GS2), which is responsible for re-fixation of ammonia. Seabra et al. (2013) have shown that GS2 abundance is low during the night, and although posttranslational regulation might compensate for low protein abundance under normal conditions, this may not suffice for the high Gly levels present in *hpr1-1*, which may then lead to accumulation of toxic levels of ammonia. However, it cannot be excluded that non-enzymatic decarboxylation of hydroxypyruvate occurs in *hpr1-1*, which, as suggested by Cousins et al. (2011), would lead to intoxication by glycolate.

5 | CONCLUSIONS

Our model is capable of describing the fluxes between the AA and Cit/α-KG pools, thus allowing the assessment of N re-fixation and *de novo* assimilation. Simulations for conditions of varied PR activity revealed that the storage capacity of the PR pathway for N in the form of Gly and Ser is correlated with the level of *de novo* N assimilation. Reduced PR activity after long-term eCO_2 exposure therefore appears to at least contribute to photosynthetic acclimation to eCO_2 by creating a C/N imbalance.

ACKNOWLEDGMENTS

The authors would like to thank Prof Hermann Bauwe for a generous gift of seeds of the *hpr1-1* mutant used in this study. Nadja Beuttenmüller and Annika Allinger are acknowledged for expert plant cultivation. K.K. received a scholarship “Landesgraduiertenförderung (LGF)” of the Federal State of Baden-Wuerttemberg (Germany). Konrad Krämer holds a stipend by the county of Baden Württemberg.

Open access funding enabled and organized by Projekt DEAL.

AUTHOR CONTRIBUTIONS

Konrad Krämer: Designed the study, conducted experiments, developed the model, and wrote the manuscript. Gabi Kepp, Simon Stutz, and Judith Brock: Conducted experiments. Arnd G. Heyer: Designed the study and wrote the manuscript.

DATA AVAILABILITY STATEMENT

Raw data can be made available by the authors upon request.

ORCID

Arnd G. Heyer https://orcid.org/0000-0003-2074-3234

REFERENCES

Abadie, C., Lalande, J., Limami, A.M. & Tcherkez, G. (2021) Non-targeted 13C metabolite analysis demonstrates broad re-orchestration of leaf metabolism when gas exchange conditions vary. *Plant, Cell & Environment*, 44(2), 445–457. https://doi.org/10.1111/pce.13940

Abadie, C., Lothier, J., Boex-Fontvieille, E., Carroll, A. & Tcherkez, G (2017) Direct assessment of the metabolic origin of carbon atoms in glutamate from illuminated leaves using 13C-NMR. *New Phytol*, 216, 1079–1089. https://doi.org/10.1111/nph.14719

Ainsworth, E. & Rogers, A. (2007) The response of photosynthesis and stomatal conductance to rising [CO2]: mechanisms and environmental interactions. *Plant, Cell & Environment*, 30(3), 258–270. https://doi.org/10.1111/j.1365-3040.2007.01641.x

Akman, D., Akman, O. & Schaefer, E. (2018) Parameter estimation in ordinary differential equations modeling via particle swarm optimization. *Journal of Applied Mathematics*, 2018, 1–9. https://doi.org/10.1155/2018/9160793

Asensio, J.S.R., Rachmilevitch, S. & Bloom, A.J. (2015) Responses of Arabidopsis and wheat to rising CO2 depend on nitrogen source and nighttime CO2 levels. *Plant Physiology*, 168(1), 156–163. https://doi.org/10.1104/pp.15.00110

Biernath, C., Bittner, S., Klein, C., Gayler, S., Hentschel, R., Hoffmann, P. et al. (2013) Modeling acclimation of leaf photosynthesis to atmospheric CO2 enrichment. *European Journal of Agronomy*, 48, 74–87. https://doi.org/10.1016/j.eja.2013.02.008

Bloom, A.J., Burger, M., Kimball, B.A. & Pinter, P.J. (2014) Nitrate assimilation is inhibited by elevated CO2 in field-grown wheat. *Nature Climate Change*, 4(6), 477–480. https://doi.org/10.1038/nclimate2183

Bloom, A.J., Burger, M., Rubio Asensio, J.S. & Cousins, A.B. (2010) Carbon dioxide enrichment inhibits nitrate assimilation in wheat and Arabidopsis. *Science*, 328(May), 899–904.

Bloom, A.J., Caldwell, R.M., Finazzo, J., Warner, R.L. & Weissbart, J. (1989) Oxygen and carbon dioxide fluxes from barley shoots depend on nitrate assimilation. *Plant Physiology*, 91(1), 352–356. https://doi.org/10.1104/pp.91.1.352

Bloom, A.J., Kasemsap, P. & Rubio-Asensio, J.S. (2020) Rising atmospheric CO2 concentration inhibits nitrate assimilation in shoots but enhances it in roots of C3 plants. *Physiologia Plantarum*, 168(4), 963–972. https://doi.org/10.1111/ppl.13040

Brooks, A. & Farquhar, G.D. (1985) Effect of temperature on the CO2/O2 specificity of ribulose-1,5-bisphosphate carboxylase/oxygenase and the rate of respiration in the light the light—estimates from gas-exchange measurements on spinach. *Planta*, 165(3), 397–406. https://doi.org/10.1007/BF00392238

Chen, G.Y., Yong, Z.H., Liao, Y., Zhang, D.Y., Chen, Y., Zhang, H.B. et al. (2005) Photosynthetic acclimation in rice leaves to free-air CO2 enrichment related to both ribulose-1,5-bisphosphate carboxylation limitation and ribulose-1,5-bisphosphate regeneration limitation. *Plant and Cell Physiology*, 46(7), 1036–1045. https://doi.org/10.1093/pcp/pci113

Cheng, S.H., Moore, B. & Seemann, J.R. (1998) Effects of short- and long-term elevated CO2 on the expression of ribulose-1,5-bisphosphate carboxylase/oxygenase genes and carbohydrate accumulation in leaves of *Arabidopsis thaliana* (L.) Heynh. *Plant Physiology*, 116(2), 715–723. https://doi.org/10.1104/pp.116.2.715

Cousins, A.B., Walker, B.J. & Pracharoenwattana, I. (2011) Peroxisomal hydroxypyruvate reductase is not essential for photorespiration in Arabidopsis but its absence causes an increase in the stoichiometry of photorespiratory CO2 release. *Photosynth Res*, 108, 91. https://doi.org/10.1007/s11120-011-9651-3

Dabu, X., Li, S., Cai, Z., Ge, T. & Hai, M. (2019) The effect of potassium on photosynthetic acclimation in cucumber during CO2 enrichment. *Photosynthetica*, 57(2), 640–645. https://doi.org/10.32615/ps.2019.073

Dellero, Y., Jossier, M., Schmitz, J., Maurino, V.G. & Hodges, M. (2016) Photorespiratory glycolate – glyoxylate metabolism. *Journal of Experimental Botany*, 67(10), 3041–3052. https://doi.org/10.1093/jxb/erw090

Douce, R., Bourguignon, J., Neuburger, M. & Rébeillé, F. (2001) The glycine decarboxylase system: a fascinating complex. *Trends in Plant Science*, 6(4), 167–176. https://doi.org/10.1016/S1360-1385(01)01892-1

Farré, E.M. & Weise, S.E. (2012) The interactions between the circadian clock and primary metabolism. *Current Opinion in Plant Biology*, 15(3), 293–300. https://doi.org/10.1016/j.pbi.2012.01.013

Fenton, F. & Karma, A. (1998) Erratum: "vortex dynamics in three-dimensional continuous myocardium with fiber rotation: filament instability and fibrillation" [Chaos 8, 20–47 (1998)]. *Chaos*, 8(4), 879–879. https://doi.org/10.1063/1.166374

Bauwe, H. (2017) Measurement of enzyme activities. In: Fernie, A., Bauwe, H. & Weber, A., *Photorespiration. Methods and Protocols*, Vol. 1653. New York, NY, U.S.A.: Humana Press. https://doi.org/10.1007/978-1-4939-7225-8

Gámez, A.L., Vicente, R., Sanchez-Bragado, R., Jauregui, I., Morcuende, R., Goicoechea, N. et al. (2020) Differential flag leaf and ear photosynthetic performance under elevated (CO2) conditions during grain filling period in durum wheat. *Frontiers in Plant Science*, 11(December), 1–12. https://doi.org/10.3389/fpls.2020.587958

Gauthier, P.P.G., Bligny, R., Gout, E., Mahé, A., Nogués, S., Hodges, M. et al. (2010) In folio isotopic tracing demonstrates that nitrogen assimilation into glutamate is mostly independent from current CO2 assimilation in illuminated leaves of *Brassica napus*. *New Phytologist*, 185(4), 988–999. https://doi.org/10.1111/j.1469-8137.2009.03130.x

Gibon, Y., Usadel, B., Blaesing, O.E., Kamlage, B., Hoehne, M., Trethewey, R. et al. (2006) Integration of metabolite with transcript and enzyme activity profiling during diurnal cycles in *Arabidopsis rosettes*. *Genome Biology*, 7(8), R76. https://doi.org/10.1186/gb-2006-7-8-r76

Gifford, R.M., Barrett, D.J. & Lutze, J.L. (2000) The effects of elevated [CO2] on the C:N and C:P mass ratios of plant tissues. *Plant and Soil*, 224(1), 1–14. https://doi.org/10.1023/A:1004790612630

Griffin, K.L. & Seemann, J.R. (1996) Plants, CO2 and photosynthesis in the 21st century. *Chemistry and Biology*, 3(4), 245–254. https://doi.org/10.1016/S1074-5521(96)90104-0

Gutiérrez, D., Morcuende, R., Del Pozo, A., Martínez-Carrasco, R. & Pérez, P. (2013) Involvement of nitrogen and cytokinins in photosynthetic acclimation to elevated CO2 of spring wheat. *Journal of Plant Physiology*, 170(15), 1337–1343. https://doi.org/10.1016/j.jplph.2013.05.006

Hanning, I. & Heldt, H.W. (1993) On the function of mitochondrial metabolism during photosynthesis in spinach (*Spinacia oleracea* L.) leaves. Partitioning between respiration and export of redox equivalents and precursors for nitrate assimilation products. *Plant Physiology*, 103(4), 1147–1154. https://doi.org/10.1104/pp.103.4.1147

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D. E. et al. (2005) SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software*, 31(3), 363–396. https://doi.org/10.1145/1089014.1089020

Hodges, M. (2002) Enzyme redundancy and the importance of 2-oxoglutarate in plant ammonium assimilation. *Journal of Experimental Botany*, 53(370), 905–916. https://doi.org/10.1093/jexbot/53.370.905

IPCC. (2014) Summary for policymakers. Climate change 2014: mitigation of climate change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. https://doi.org/10.1017/CBO9781107415324

Jauregui, I., Aparicio-Tejo, P.M., Avila, C., Cañas, R., Sakalauskiene, S. & Aranjuelo, I. (2016) Root-shoot interactions explain the reduction of leaf mineral content in Arabidopsis plants grown under elevated [CO2] conditions. *Physiologia Plantarum*, 158(1), 65–79. https://doi.org/10.1111/ppl.12417

Jauregui, I., Aparicio-Tejo, P.M., Avila, C., Rueda-López, M. & Aranjuelo, I. (2015) Root and shoot performance of *Arabidopsis thaliana* exposed to elevated CO2: a physiologic, metabolic and transcriptomic response. *Journal of Plant Physiology*, 189, 65–76. https://doi.org/10.1016/j.jplph.2015.09.012

Jauregui, I., Pozueta-Romero, J., Cordoba, J., Avice, J.-C., Aparicio-Tejo, P. M., Baroja-Fernández, E. et al. (2018) Unraveling the role of transient starch in the response of Arabidopsis to elevated CO2 under long-day conditions. *Environmental and Experimental Botany*, 155, 1–27.

Kanemoto, K., Yamashita, Y., Ozawa, T., Imanishi, N., Nguyen, N.T., Suwa, R. et al. (2009) Photosynthetic acclimation to elevated CO2 is dependent on N partitioning and transpiration in soybean. *Plant Science*, 177(5), 398–403. https://doi.org/10.1016/j.plantsci.2009.06.017

Keys, A.J. (2006) The re-assimilation of ammonia produced by photorespiration and the nitrogen economy of C3 higher plants. *Photosynthesis Research*, 87(2), 165–175. https://doi.org/10.1007/s11120-005-9024-x

Kizildeniz, T., Pascual, I., Irigoyen, J.J. & Morales, F. (2021) Future CO2, warming and water deficit impact white and red Tempranillo grapevine: photosynthetic acclimation to elevated CO2 and biomass allocation. *Physiologia Plantarum*, 172(3), 1779–1794. https://doi.org/10.1111/ppl.13388

Krämer, K., Krämer, J., & Heyer, A. (2020). Paropt: parameter optimizing of ODE-systems.

Kuehny, J.S., Peet, M.M., Nelson, P.V. & Willits, D.H. (1991) Nutrient dilution by starch in CO2-enriched chrysanthemum. *Journal of Experimental Botany*, 42(6), 711–716. https://doi.org/10.1093/jxb/42.6.711

Küstner, L., Nägele, T. & Heyer, A.G. (2019) Mathematical modeling of diurnal patterns of carbon allocation to shoot and root in *Arabidopsis thaliana*. *npj Systems Biology and Applications*, 5(1), 1–11. https://doi.org/10.1038/s41540-018-0080-1

Leegood, R.C., Lea, P.J., Adcock, M.D. & Häusler, R.E. (1995) The regulation and control of photorespiration. *Journal of Experimental Botany*, 46-(special_issue), 1397–1414. https://doi.org/10.1093/jxb/46.special_issue.1397

Li, J., Weraduwage, S.M., Preiser, A.L., Tietz, S., Weise, S.E., Strand, D.D. et al. (2019) A cytosolic bypass and g6p shunt in plants lacking peroxisomal hydroxypyruvate reductase1. *Plant Physiology*, 180(2), 783–792. https://doi.org/10.1104/pp.19.00256

Liang, Y., Wang, J., Zeng, F., Wang, Q., Zhu, L., Li, H. et al. (2021) Photorespiration regulates carbon-nitrogen metabolism by magnesium chelatase D subunit in rice. *Journal of Agricultural and Food Chemistry*, 69(1), 112–125. https://doi.org/10.1021/acs.jafc.0c05809

Liu, S., Ji, C., Wang, C., Chen, J., Jin, Y., Zou, Z. et al. (2018) Climatic role of terrestrial ecosystem under elevated CO2: a bottom-up greenhouse gases budget. *Ecology Letters*, 21(7), 1108–1118. https://doi.org/10.1111/ele.13078

Long, S.P., Ainsworth, E.A., Rogers, A. & Ort, D.R. (2004) Rising atmospheric carbon dioxide: plants FACE the future. *Annual Review of Plant Biology*, 55, 591–628. https://doi.org/10.1146/annurev.arplant.55.031903.141610

Misra, B.B. & Chen, S. (2015) Advances in understanding CO2 responsive plant metabolomes in the era of climate change. *Metabolomics*, 11(6), 1478–1491. https://doi.org/10.1007/s11306-015-0825-4

Nägele, T., Henkel, S., Hörmiller, I., Sauter, T., Sawodny, O., Ederer, M. et al. (2010) Mathematical modeling of the central carbohydrate metabolism in Arabidopsis reveals a substantial regulatory influence of vacuolar invertase on whole plant carbon metabolism. *Plant Physiology*, 153(1), 260–272. https://doi.org/10.1104/pp.110.154443

Nunes-Nesi, A., Fernie, A.R. & Stitt, M. (2010) Metabolic and signaling aspects underpinning the regulation of plant carbon nitrogen interactions. *Molecular Plant*, 3(6), 973–996. https://doi.org/10.1093/mp/ssq049

Paul, M.J. & Driscoll, S.P. (1997) Sugar repression of photosynthesis: the role of carbohydrates in signalling nitrogen deficiency through source: sink imbalance. *Plant, Cell & Environment*, 20(1), 110–116. https://doi.org/10.1046/j.1365-3040.1997.d01-17.x

Rachmilevitch, S., Cousins, A.B. & Bloom, A.J. (2004) Nitrate assimilation in plant shoots depends on photorespiration. *Proceedings of the National Academy of Sciences of the United States of America*, 101(31), 11506–11510. https://doi.org/10.1073/pnas.0404388101

Renné, P., Dreßen, U., Hebbeker, U., Hille, D., Flügge, U.I., Westhoff, P. et al. (2003) The Arabidopsis mutant dct is deficient in the plastidic glutamate/malate translocator DiT2. *Plant Journal*, 35(3), 316–331. https://doi.org/10.1046/j.1365-313X.2003.01806.x

Sage, R.F. (1994) Acclimation of photosynthesis to increasing atmospheric CO2: the gas exchange perspective. *Photosynthesis Research*, 39(3), 351–368. https://doi.org/10.1007/BF00014591

Seabra, A.R., Silva, L.S. & Carvalho, H.G. (2013) Novel aspects of glutamine synthetase (GS) regulation revealed by a detailed expression analysis of the entire GS gene family of *Medicago truncatula* under different physiological conditions. *BMC Plant Biology*, 13(1), 1–15. https://doi.org/10.1186/1471-2229-13-137

Sengupta, S., Basak, S. & Peters, R.A. (2018) Particle swarm optimization: a survey of historical and recent developments with hybridization perspectives. *ArXiv*, 1–8, 157–191. https://doi.org/10.3390/make1010010

Sharkey, T.D. (1988) Estimating the rate of photorespiration in leaves. *Physiologia Plantarum*, 73(1), 147–152. https://doi.org/10.1111/j.1399-3054.1988.tb09205.x

Shi, S., Xu, X., Dong, X., Xu, C., Qiu, Y. & He, X. (2021) Photosynthetic acclimation and growth responses to elevated CO2 associate with leaf nitrogen and phosphorus concentrations in mulberry (*Morus multicaulis* Perr .). *Forests*, 12(6), 1–14.

Shimono, H., Suzuki, K., Aoki, K., Hasegawa, T. & Okada, M. (2010) Effect of panicle removal on photosynthetic acclimation under elevated CO2 in rice. *Photosynthetica*, 48(4), 530–536. https://doi.org/10.1007/s11099-010-0070-z

Smith, N.G. & Dukes, J.S. (2013) Plant respiration and photosynthesis in global-scale models: incorporating acclimation to temperature and CO2. *Global Change Biology*, 19(1), 45–63. https://doi.org/10.1111/j.1365-2486.2012.02797.x

Solomon, S., Plattner, G.K., Knutti, R. & Friedlingstein, P. (2009) Irreversible climate change due to carbon dioxide emissions. *Proceedings of the National Academy of Sciences of the United States of America*, 106(6), 1704–1709. https://doi.org/10.1073/pnas.0812721106

Sweetlove, L.J., Beard, K.F.M., Nunes-Nesi, A., Fernie, A.R. & Ratcliffe, R.G. (2010) Not just a circle: flux modes in the plant TCA cycle. *Trends in Plant Science*, 15(8), 462–470. https://doi.org/10.1016/j.tplants.2010.05.006

Tcherkez, G., Gauthier, P., Buckley, T.N., Busch, F.A., Barbour, M.M., Bruhn, D., Heskel, M.A., Gong, X.Y. & Crous, K.Y. (2017) Leaf day respiration: low CO2 flux but high significance for metabolism and carbon balance. *New Phytologist*, 216, 986–1001. https://doi.org/10.1111/nph.14816

Tcherkez, G., Mahé, A., Gauthier, P., Mauve, C., Gout, E., Bligny, R., Cornic, G. & Hodges, M. (2009) In Folio Respiratory Fluxomics Revealed by 13C Isotopic Labeling and H/D Isotope Effects Highlight the Noncyclic Nature of the Tricarboxylic Acid "Cycle" in Illuminated Leaves. *Plant Physiology*, 151, 620–630. https://doi.org/10.1104/pp.109.142976

Timm, S., Florian, A., Jahnke, K., Nunes-Nesi, A., Fernie, A.R. & Bauwe, H. (2010) The hydroxypyruvate-reducing system in arabidopsis: multiple enzymes for the same end. *Plant Physiology*, 155(2), 694–705. https://doi.org/10.1104/pp.110.166538

Timm, S., Nunes-Nesi, A., Florian, A., Eisenhut, M., Morgenthal, K., Wirtz, M. et al. (2021) Metabolite profiling in *Arabidopsis thaliana* with moderately impaired photorespiration reveals novel metabolic links and compensatory mechanisms of photorespiration. *Metabolites*, 11(6), 391. https://doi.org/10.3390/metabo11060391

Timm, S., Nunes-Nesi, A., Pärnik, T., Morgenthal, K., Wienkoop, S., Keerberg, O. et al. (2008) A cytosolic pathway for the conversion of hydroxypyruvate to glycerate during photorespiration in Arabidopsis. *Plant Cell*, 20(10), 2848–2859. https://doi.org/10.1105/tpc.108.062265

Wang, F., Gao, J., Yong, J.W.H., Wang, Q., Ma, J. & He, X. (2020) Higher atmospheric CO2 levels favor C3 plants over C4 plants in utilizing ammonium as a nitrogen source. *Frontiers in Plant Science*, 11-(December), 1–16. https://doi.org/10.3389/fpls.2020.537443

Watanabe, C.K., Sato, S., Yanagisawa, S., Uesono, Y., Terashima, I. & Noguchi, K. (2014) Effects of elevated CO2 on levels of primary metabolites and transcripts of genes encoding respiratory enzymes and their diurnal patterns in *Arabidopsis thaliana*: possible relationships with respiratory rates. *Plant and Cell Physiology*, 55(2), 341–357. https://doi.org/10.1093/pcp/pct185

Wong, S.C. (1990) Elevated atmospheric partial pressure of CO2 and plant growth - II. Non-structural carbohydrate content in cotton plants and its effect on growth parameters. *Photosynthesis Research*, 23(2), 171–180. https://doi.org/10.1007/BF00035008

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Krämer, K., Kepp, G., Brock, J., Stutz, S. & Heyer, A.G. (2022) Acclimation to elevated CO_2 affects the C/N balance by reducing de novo N-assimilation. *Physiologia Plantarum*, e13615. Available from: https://doi.org/10.1111/ppl.13615

Supporting Information

Table S1: Means ± SD of dark respiration in μmol/gFW*h (n = 12, 10, 11, and 12 for Col-0 and hpr1-1 acclimated and non-acclimated to elevated CO_2 respectivly). ANOVA for genotype and treatment revealed a significant interaction ($P < 2^{-16}$). TukeyHSD is presented. Groups not sharing the same letter are significantly different.

Genotype and condition	Dark respiration
Col-0 non-acclimated	-20.99 ± 5.49[b]
Col-0 acclimated	-21.29 ± 5.90[b]
hpr1-1 non-acclimated	-30.38 ± 6.35[a]
hpr1-1 acclimated	-22.23 ± 4.08[b]

Table S2: Means ± SD of metabolites measured at ambient CO_2 concentrations vs. data from 48 h of elevated CO_2 concentrations (n = 5).

	Begin of light period				End of light period			
	Ambient CO_2		Elevated CO_2		Ambient CO_2		Elevated CO_2	
Metabolites	Col-0	*hpr1-1*	Col-0	*hpr1-1*	Col-0	*hpr1-1*	Col-0	*hpr1-1*
Gly	0.054 ± 0.015	0.119 ± 0.061	0.015 ± 0.003	0.019 ± 0.004	0.225 ± 0.039	1.512 ± 0.366	0.15 ± 0.055	0.33 ± 0.16
Ser	0.513 ± 0.124	10.141 ± 3.62	0.43 ± 0.17	0.91 ± 0.40	1.085 ± 0.238	21.571 ± 3.66	0.56 ± 0.15	2.29 ± 1.10
MF	9.45 ± 1.36	12.21 ± 2.63	7.627 ± 4.27	4.85 ± 0.91	19.55 ± 2.82	27.20 ± 2.76	12.33 ± 3.39	14.99 ± 3.68
Cit	11.76 ± 0.79	16.32 ± 1.13	11.426 ± 2.138	21.62 ± 1.28	6.65 ± 0.33	10.73 ± 1.22	11.177 ± 2.62	17.62 ± 2.45

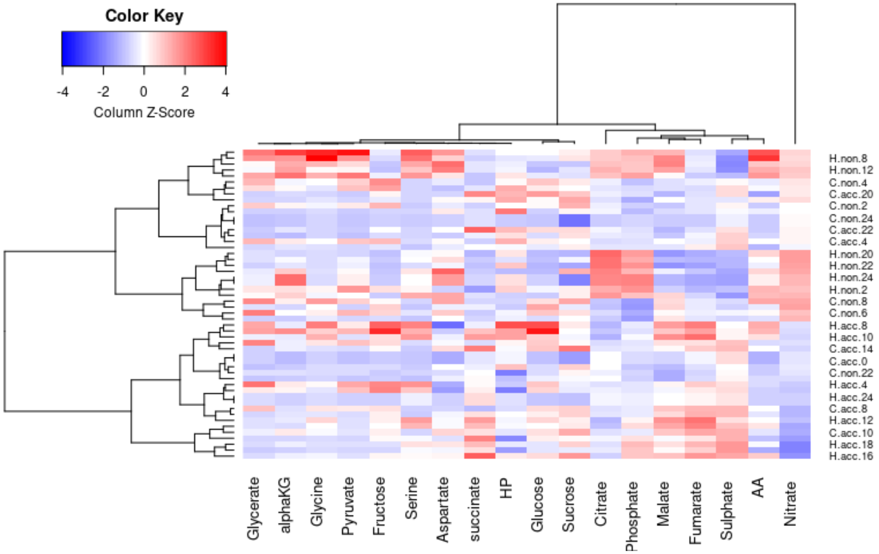


Figure S1: Heatmap of metabolite profiling data (n = 5). Data are centered and scaled for each metabolite. Dendrograms were calculated by hierarchical clustering. C = Col-0 and H = hpr1-1. Acclimated to eCO2 = acc and non-acclimated to eCO2 = non, respectively. Numbers display simulation time points. Light period is from 0 to 8 h Amino-acid pool = AA and hexosephosphates = HP

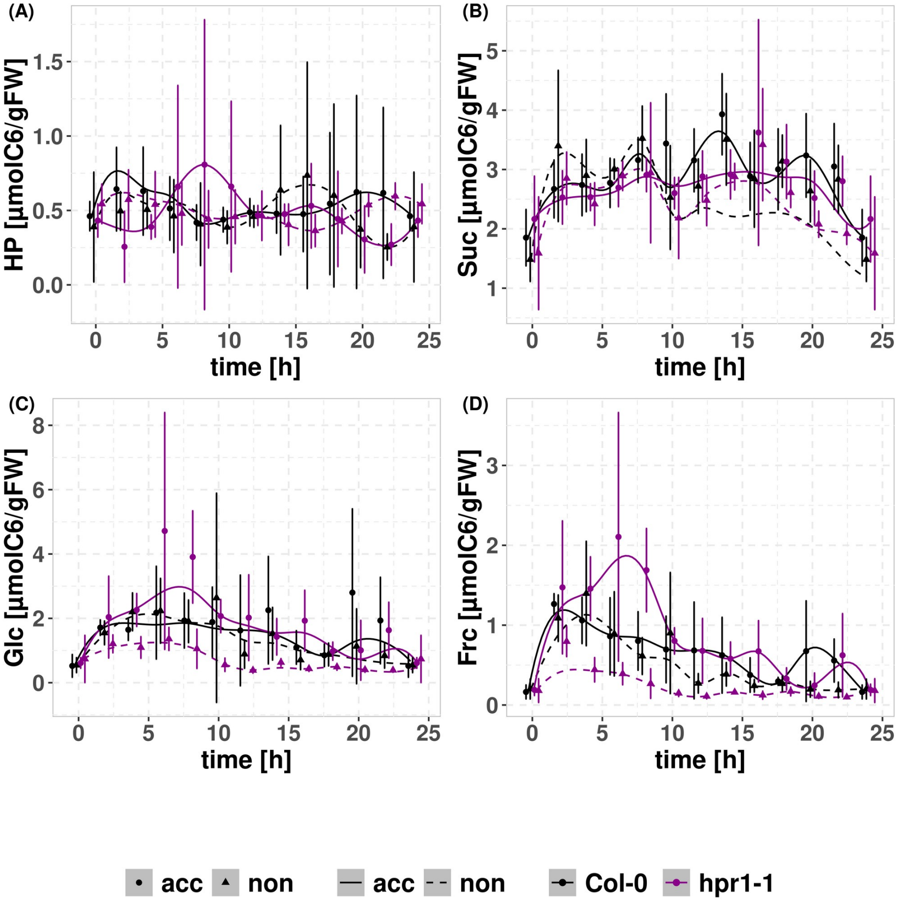


Figure S2: Means ± standard deviation of measured values (n = 5) and results of simulations for hexosephosphates (HP) (A), sucrose (Suc) (B), glucose (Glc) (C) and fructose (Frc) (D). Lines present the mean of 20 simulations. The measured values are indicated by dots. Light period is from 0 to 8 h. To prevent overlapping of error bars calculated for identical time points, bars were displaced by 0.25 h.

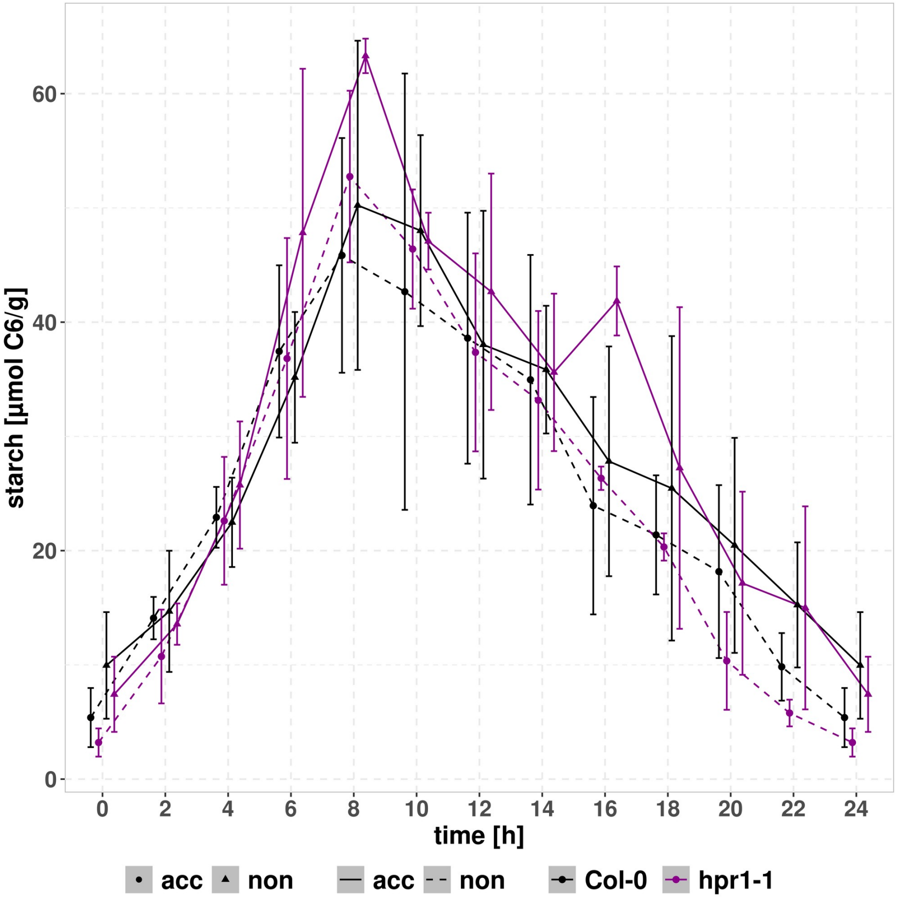


Figure S3: Diurnal dynamics of starch content. Means ± standard deviation of measured values (n = 6). Light period is from 0 to 8 h. To prevent overlapping of error bars calculated for identical time points, bars were displaced by 0.25 h.

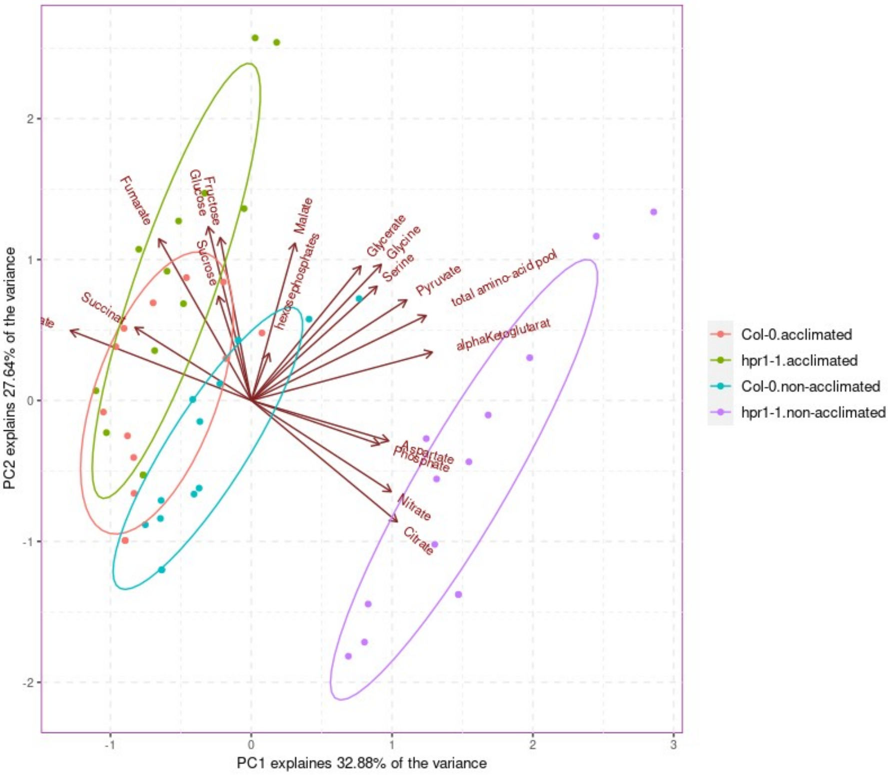


Figure S4: PCA of metabolite profiling data. Data are centered and scaled. (n = 5).

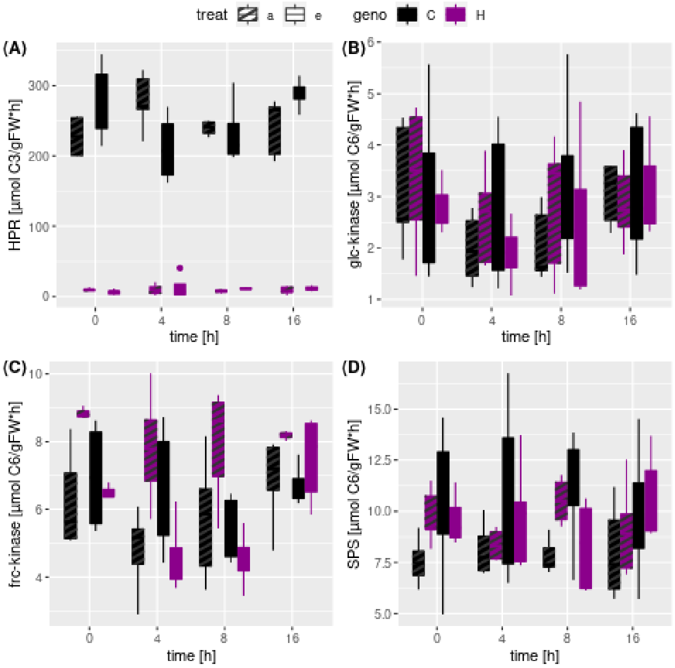


Figure S5: Measured values of enzymatic activities (n = 5) hydroxy-pyruvate-reductase (A), glucokinase (B), fructokinase (C) and sucrose-phosphate-synthase (D). Lightperiod for 0, 4 and 8 h. Darkperiod at 16 h.

Code of ODE-Model:

```
auto MODEL (double &t, std::vector<double> &params, std::vector<double> &states, double &
PS, double & PR, double & RESPIRATION, double & STARCH) -> void {

/*
Extracting the parameter
*/
// Sucrose cycling ======================
double SPS_VM = params[0];
double SPS_KM = params[1];

double INV_VM = params[2];
double INV_KM = params[3];
double INV_GLC_KI = params[4];
double INV_FRC_KI = params[5];

double HXK_GLC_VM = params[6];
double HXK_FRC_VM = params[7];
double HXK_GLC_KM = params[8];
double HXK_FRC_KM = params[9];
double HXK_GLC_KI = params[10];
double HXK_FRC_KI = params[11];

double SUC_EXP = params[12];

double HP_TO_MF = params[13];

double HP_TO_BM = params[14];

double AA_TO_BMEXP = 0.;
double AA_TO_BMEXP_Day = params[15];
double AA_TO_BMEXP_Night = params[16];
if(t < 8.2) {
AA_TO_BMEXP = AA_TO_BMEXP_Day;
} else {
AA_TO_BMEXP = AA_TO_BMEXP_Night;
}

// PR ==================================
double GLY_TO_SER_day = params[17];
double GLY_TO_SER_night = params[18];
double GLY_TO_SER = 0.;
if(t < 8.2) {
GLY_TO_SER = GLY_TO_SER_day;
} else {
```

```
GLY_TO_SER = GLY_TO_SER_night;
}
double HPR_VM = params[19];
double HPR_KM = params[20];

double alpha = params[21];

double PRIM_FIX;
double prim_day = params[22];
double prim_night = params[23];
if(t < 8.2) {
PRIM_FIX = prim_day;
} else {
PRIM_FIX = prim_night;
}

double CIT_TO_MF = params[24];
double MF_TO_CIT = params[25];

double SER_TO_BM = params[26];
double SWITCH_HPR = params[27];
/*
Extracting states
*/
double HP = states[0];
double SUC = states[1];
double GLC = states[2];
double FRC = states[3];
double CIT = states[4];
double MF = states[5];
double AA = states[6];
double GLY = states[7];
double SER = states[8];

double SPS = (SPS_VM*HP)/(SPS_KM + HP);
double HXK_GLC = (HXK_GLC_VM*GLC) / (HXK_GLC_KM + GLC*(1. +
HP/HXK_GLC_KI));
double HXK_FRC = (HXK_GLC_VM*FRC) / (HXK_FRC_KM + FRC*(1. +
HP/HXK_FRC_KI));
double INV = (INV_VM*SUC) / (INV_KM*(1. + FRC/INV_FRC_KI) + SUC*(1. +
GLC/INV_GLC_KI));
double HPR = (HPR_VM*SER) / (HPR_KM + SER);

double hpr_adjusted = HPR;
if(t > 8.) {
hpr_adjusted = HPR*SWITCH_HPR;
}
HPR = hpr_adjusted;

PR = PR + alpha;
```

```
double PR_prime = PR/2.;

double x = PR_prime - HPR;

if(x < 0.0) {

x = 0.;
HPR = PR_prime; // there is only a certain amount of glyoxylate available for HPR
}

double GLY_CLEAVAGE = GLY_TO_SER;

double G_T_N = GLY*GLY_CLEAVAGE;
if((PR_prime - HPR) < 0.) {
G_T_N = G_T_N + HPR - PR_prime;
}

double refixation = GLY*GLY_CLEAVAGE*4.878; // N1 flux. --> *4.878 = (1/1.23)*6

double RESP_from_CA = 0.;
if( ((RESPIRATION + 0.5*GLY*GLY_CLEAVAGE)/6.) > 0.) {
RESP_from_CA = 0.;
} else {
RESP_from_CA = (RESPIRATION + 0.5*GLY*GLY_CLEAVAGE)/6.;
}

double NETPS = PS/6. - (PR/6.) -(RESP_from_CA);
if(t > 8.) {
NETPS = PS/6.;
}


/*
Actual ODE System
*/
double ddtHP = states[0] = hpr_adjusted/2. + NETPS -STARCH - SPS + HXK_FRC + HXK_GLC
- HP_TO_MF*HP - HP*HP_TO_BM;// C6 N0
double ddtSuc = states[1] = SPS - INV - SUC_EXP*SUC; // C6 N0
double ddtGlc = states[2] = INV - HXK_GLC; // C6 N0
double ddtFrc = states[3] = INV - HXK_FRC; // C6 N0
double ddtCIT = states[4] = x*0.813*AA - PRIM_FIX*CIT + MF*MF_TO_CIT - refixation -
CIT*CIT_TO_MF; // C6 N0
double ddtMF = states[5] = HP_TO_MF*HP + CIT*CIT_TO_MF + RESP_from_CA -
MF*MF_TO_CIT; // C6 N0
double ddtAA = states[6] = - AA_TO_BMEXP*AA -x*0.813*AA + refixation +
PRIM_FIX*CIT; // C6 N1.23
double ddtGLY = states[7] = PR_prime -GLY*GLY_CLEAVAGE -GLY*GLY_TO_SER;//C2 N1!
double ddtSER = states[8] = GLY*GLY_TO_SER*(2./3.) + GLY*GLY_CLEAVAGE*(1./3.) -
hpr_adjusted -SER_TO_BM*SER; //C3 N1!
}
```

Interaction of Nitrate Assimilation and Photorespiration at Elevated CO_2

Authors

Konrad Krämer

E-mail: konrad_kraemer@yahoo.de

Judith Brock

E-mail: judith.brock@bio.uni-stuttgart.de

Arnd Heyer

E-mail: arnd.heyer@bio.uni-stuttgart.de

Biomolecular Systems, Department of Plant Biotechnology, University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany.

Summary

In this study plants were grown at either ambient or elevated CO_2 concentrations with nitrate (NO_3^-) as sole N source. To investigate the interaction of N assimilation with photorespiration (PR) the mutant *hpr1-1* was used. Using metabolic and enzymatic data of a full diurnal cycle mathematical simulations were conducted. The results demonstrates that PR stimulates the Glutamine-synthetase 2 (GS)/ Glutamine-Oxoglutarate-aminotransferase (GOGAT) cycle.

Therefore we propose the theory that the *hpr1-1* mutant suffers from the need to maintain high fluxes through the GS/GOGAT cycle as this demands a lot of cell energy. Beyond that we show that carbon (C) and nitrogen (N) are differently distributed within the mutant.

Moreover, we investigated the flux changes in N assimilation. We could show that nitrate assimilation is not inhibited at eCO_2. Furthermore, we analyzed that the ratio of biomass production from hexosephosphates (HP) to nitrate assimilation showing that the ratio is elevated at eCO_2. Therefore we suggest that N is diluted at eCO_2.

Significant statements

The mutant *hpr1-1* suffers from increased turnover in the GS/GOGAT cycle. Thereby limitin growth of the plant. Nitrate reduction is not limited at eCO_2 rather the ratio between C and N assimilation is affected.

Introduction

In the course of climate change a substantial increase of the atmospheric CO_2 concentration is expected in the 21th century (IPCC, 2014; Szulejko *et al.*, 2017). Because CO_2 is the substrate for plant photosynthesis, alteration in the CO_2 level have a direct impact on plant metabolism. The enzyme ribulose-1,5-bisphosphat-carboxylase/-oxygenase (RUBISCO) catalyses CO_2 fixation, but can also use O_2 as substrate, resulting in the production of 2-phosphoglycolate, which is then further processed in the so called photorespiration (PR) pathway (Shih *et al.*, 2015). As carboxylation is preferred over oxygenation (Sharkey, 1988) higher CO_2 concentrations will distinctly reduce the probability of oxygenation in the course of climate change (Sharkey, 1988)

In the course of PR two molecules of 2-phosphoglycolate (2-PG) are converted to one molecule of glyceraldehyd-3-phosphate, which is fed back into the Calvin-Benzen-cycle (Huma *et al.*, 2018). More precisely, 2-PG is converted to glycolate which is subsequently oxidized to glyoxylate, producing H_2O_2 as a byproduct in peroxisomes. Next, glyoxylate is transaminated to glycine (Gly). The N source is either serine (Ser) or glutamate (Glu). resulting in the production of hydroxypyruvate and α-Ketoglutarate (α-KG), respectively (Nunes-Nesi, Fernie and Stitt, 2010). Hydroxypyruvate is reduced by the hydroxypyruvate-reductase (HPR) forming glycerate which can be phosphorylated to glyceraldehyd-3-phosphate (Timm *et al.*, 2008a). Gly is transported to mitochondria, where it is used by the glycine-decarboxylase (GDC), converting Gly, NAD^+ and Tetrahydrofolic acid to NADH, CO_2, NH_4^+ and N^5, N^{10} -Methylene Tetrahydrofolic acid. The CO_2 evolution in this step is eponym for the PR (Rebeille, Neuburger and Douce, 1994). An second molecule of Gly is converted together with N^5, N^{10} -Methylene Tetrahydrofolic acid to serine (Ser) by the enzyme serine-hydroxymethyl transferase (SHMT). Ser is transported into the peroxisomes, where it serves as N donor for Gly production from glyoxylate (Nunes-Nesi, Fernie and Stitt, 2010). Besides the amount of energy consumed in PR, toxic intermediates occur such as glycolate and glyoxylate, which must be removed quickly (Dellero *et al.*, 2016). Even though the entire pathway, starting from oxygenation of Rubisco, seems futile there is strong evidence that PR is crucial in the response to abiotic stress (Voss *et al.*, 2013), is important for photoprotection (Guan *et al.*, 2004) and nitrogen (N) assimilation (Cousins and Bloom, 2004; Bloom, 2015; Kraemer *et al.*, 2021).

Various studies have shown that eCO_2 negatively affects N assimilation and that this is at least partly due to reduced PR (Bloom *et al.*, 2014; Bloom, 2015; Bloom, Kasemsap and Rubio-Asensio, 2020; Kraemer *et al.*, 2021). As mentioned above the GDC reaction releases NH_4^+ which has to be refixed. This is done in the Glutamine-synthetase 2 (GS)/ Glutamine-Oxoglutarate-aminotransferase (GOGAT) cycle. GS uses NH_4^+ and Glu to produce glutamine (Gln), which is used for transamination of α-KG by GOGAT, yielding two molecules Glu (Huma *et al.*, 2018). Thus, PR has an impact on α-KG production and amino acid homeostasis as well as the malate shuttle that transports α-KG into the plastids (Nunes-Nesi, Fernie and Stitt, 2010). Notably, Wallsgrove et al., 1987 showed that GS mutants suffer under non-photorespiratory conditions.

Bloom, 2015 proposed that increased PR flux causes higher Mal levels in the cytosol that maintain turnover in the GS/GOGAT cycle and, concomittantly, produce NADH serving as substrate for nitrate-reductase (NR). However, Andrews et al., 2019 showed that eCO_2 affects N assimilation independently on the form of N administered, and that no inhibition of nitrate (Nit) assimilation was detectable. Thus, themechanism by which eCO2 interfers with N assimilation remains unclear.

In this study we focus on the role of PR fluxes for N assimilation. We compare *Arabidopsis thaliana* plants grown at either eCO_2 or ambient CO_2 (aCO2) concentration. To achieve different PR fluxes, we used the *hpr1-1* mutant which lacks peroxisomale HPR (Timm *et al.*, 2008a). This results in elevated levels of Gly and Ser (Timm *et al.*, 2011), stunnted growth at aCO_2 and a chlorotic phenotype (Li *et al.*, 2019). The mutant shows elevated levels of α-KG and free amino-acids (Timm *et al.*, 2021). We propose that growth reduction in the mutant is at least partly due to a disturbed energy household resulting from insufficient α-KG levels.

Kraemer et al., 2021 identified kinetic parameters for which the ode-solver *(Hindmarsh et al., 2005)* yield an *in silico* solution of the state trajectories similar to the measured ones. This approach was previously used by *Nägele et al., 2010 and Küstner, Nägele and Heyer, 2019.*

Results

Diurnal dynamics of metabolites

Net photosynthetic rates (*PS*) increased when plants were grown at eCO_2. The wildtype had significant higher *PS* as compared to the mutant ($P < 2e\text{-}16$) under both conditions (Col-0 at aCO_2 = 85.41 ± 12.75 µmol/g * h, Col-0 at eCO2 = 128.13 ± 44.96 µmol/g * h, *hpr1-1* at aCO_2 = 76.78 ± 19.91 µmol/g * h, *hpr1-1* at eCO_2 = 98.56 ± 35.75 µmol/g * h). These results are in accordance with previous observations (Jauregui *et al.*, 2016). Nevertheless, a significant increase in MF over the day-time was observed for the *hpr1-1* mutant (P = 4.1e-10). Interestingly, also Cit amounts were elevated in the *hpr1-1* mutant as compared to the wildtype at aCO_2 (P = 2.47e-07) and showed a different response to eCO_2 with an increase in the wildtype (P = 0.0007089 t-test solely for wildtype) and a decline in the mutant (P = 5.075e-06 t-test solely for *hpr1-1*).

The PR intermediates Gly and Ser were elevated in the mutant under both conditions (Fig. 4 (a) and (b)). As expected, the concentrations were higher at aCO_2 compared to eCO_2. For NH_4^+, which is also produced during PR, a significant genotype and treatment effect was observed with *hpr1-1* showing elevated concentration and a general decrease for eCO_2 (P = 5.67e-05 and P = 4.38e-06, respectively).

Similar to the NH_4^+ concentration, Glu levels (Fig. 5 (a)) were elevated in the mutant compared to the wildtype (P = 0.000771). Additionally, the Glu concentration was also significantly increased in mutant plants growing at aCO_2 (P = 2.03e-10). This effect could not be observed for Col-0. For Gln (Fig. 5 (b)) neither genotype nor treatment effects were observed. However, when considering only the mutant data, a significant treatment effect became obvious, showing elevated levels at aCO_2 (P = 0.0163). The α-KG concentration (Fig. 5 (c)) was strongly increased for *hpr1-1* at aCO_2, thus clearly pointing to an increased Glu turnover for transamination of glyoxylate.

Furthermore, the AA pool (Fig. 5 (d)) was elevated in the mutant ($P < 2.0e\text{-}16$). Considering only the mutant, also a significant treatment effect was observed ($P < 2.0e\text{-}16$). Thus, no indications for N-starvation in the mutant were found, which must therefore be excluded as a reason for low biomass production of *hpr1-1*.

Diurnal dynamics of enzyme kinetics

Three enzyme kinetics were determined in order to use them as lower and upper boundaries in the optimization process (see below). For *NR* (Fig. S2 (a)) Col-0 at eCO_2 showed the highest activity. This constituted a significant genotype effect (P = 0.03461) that was unexpected considering the reported decrease of N assimilation at eCO_2 by Bloom et al., 2014. In addition, a time effect with highest activity at midday was observed independent of the condition (P = 0.00165). *GS* activity (Fig. S2 (b)) was clearly affected by treatment with higher activity at aCO_2 (P = 4.14e-13). Notably, *hpr1-1* had even higher *GS* activity than Col-0 for plants grown at aCO_2 (P = 0.00206). Probably, higher turnover rates of the GS/GOGAT cycle are required in order to supply enough Glu as N provider for PR. Not surprisingly a genotype effect was found for HPR activity (Fig,. S2 (c); P < 2e-16). In addition, a treatment effect was detected, which was dependend on the *hpr1-1* data (P = 0.0461).

Model construction

To investigate the impact of elevated CO_2 and altered PR flux on dynamics of metabolic networks, an ODE-model was constructed that covered not only carbon but also nitrogen metabolism. Simulations of the model can predict *in silico* time courses for all states, i.e. all metabolites over full diurnal cycles. The only information needed for solving the ODE-system are the starting condition for all states and a set of kinetic parameters.

The model used for this study is depicted in Fig. 1. Outside the system boundary lie the states for O_2, CO_2, BM, Starch, EXP, and NO_3^-. Within the system boundary the levels of Gly, Ser, HP, MF, Cit, α-KG, Glu, Gln, AA and NH_4^+ are considered for simulations. For the fluxes *hp2BM/EXP*, *hp2MF*, *Cit2MF*, *MF2Cit*, *Cit2KG*, *Ser2AA*, *AA2BM/EXP* mass balance kinetics were used. The remaining reactions are represented as Michaelis-Menten kinetics.

The PR is represented by the reaction yielding Gly and is calculated according to (Sharkey, 1988). Gly is converted to Ser, the state of which was used as the precursor for the *HPR* reaction that feeds into the HP pool. Ser can also flow into the AA pool during the night representing the reaction of Ser-Pyruvate aminotransferase.

The HP pool consists of Glucose-6-Phosphate and Fructose-6-Phosphate, which were also used to represent all short-lived intermediates, e.g., the intermediates of the Calvin-Benson-Cycle. Photosynthesis was the input into this pool. The HP pool was set as the precursor for cell wall synthesis and the supply of carbon to sinks. In the ODE-Model these reactions are represented by *hp2BM/EXP*. In addition, HP is the substrate and the product of starch production and degradation, respectively. Starch production and degradation was assumed to be linear. As suggested by Gauthier et al., 2010 and Tcherkez et al., 2009, Cit is not or only to a very low extent produced during the day. Therefore, HP was used as the precursor for the MF pool. In addition, Cit can also be used to produce MF. MF is used for respiration which was set constant during the entire simulation time as described by Küstner et al., 2019 and Kraemer et al., 2021. Due to the production of CO_2 during the *GDC* reaction the respiration was adjusted in order to prevent an erroneous C-balance.

During the reaction from Gly to Ser NH_4^+ is produced and, together with Glu can be used by the *GS* to produce Gln (Fig. S2). Gln can transaminate α-KG, catalyzed by *GOGAT*, to produce two molecules of Glu. Two reactions can yield α-KG. The first reaction is from Cit, representing the TCA. The second is from Glu and represents the reaction of Glu-Glyoxylate-Aminotransferase. For Gly production, one molecule of N is required. This can derive from Ser, which is represented by the *HPR* reaction in the model, or from Glu. The PR flux into the system, the *HPR* and the reaction from Glu to α-KG are all N1-fluxes (µmol N1/(gFW*h)]. Thus, the three fluxes can be balanced, and Glu to α-KG can be expressed as PR minus HPR.

Alternatively, NH_4^+ can be produced by reduction of nitrate through *NR* (Fig. S2). Because the large proportion of NO_3^- stored in the vacuole cannot be used by NR, the concentration was adjusted, based on the cellular proportion of the cytosole in mature leaves (Koffler et al., 2013), which is . This results in about 5% of the total cell volume. *NR* is the only flux that provides newly assimilated N. Thus, the step towards AA production was linked to *NR*. Accordingly, the flux from Glu to AA was implemented using NR multiplied by a correction factor (*correct*) in the interval [0.5, 1], meaning that at least 50% of new N is used for AA production. This factor was based on observations by Gauthier et al., 2010, who found that ca. 50% of Ser consists of new N. The photorespiration intermediates (Gly and Ser) are the only other N sinks, where new N can be stored. Thus the correction factor determines in which compounds new N is stored.

Because PR flux as defined by Sharkey, 1988, is directly dependent on photosynthesis, the summand α was introduced to allow turnover of photorespiratory intermediates such as glycolate during the night,

The step from Gly to Ser is catalyzed by the two enzymes *GDC* and *SHMT*. For both reactions Michaelis-Menten kinetics were employed. However, turnover of *SHMT* cannot exceed that of *GDC*, and thus *SHMT* was limited to the *GDC* value. It is, however, known from the literature that the *GDC* flux may well be substantially higher then the *SHMT* reaction (Rebeille, Neuburger and Douce, 1994).

Because the model structure presented in Fig. 1 does not allow the amino acids Gly and Ser to contribute to biomass via the *AA* route, a separate pathway was established for these compounds. Based on proteome information for *Arabidopsis* (Berardini ***et al.***, 2015), a proportion of about 10% of the amino acids in protein is Ser. However, it is higher in Rubisco, which makes up around one third of total protein (Atkinson ***et al.***, 2017), and thus ca. 14% of the proteome of *Arabidopsis* consists of Ser. Thus, the flux from Ser to BM was set as 14% of the flux from AA to BM/EXP.

Preliminary parameter identification

Several generations of parameter optimization using the above described model structure yielded acceptable results for most states. However, particularly Glu levels during the night were underestimated as can be seen in Fig. 2 (a). Choi et al., 1999 showed that the GS enzyme of *Canavalina lineata* is activated by a reduction at two cysteine residues conserved among all known plastidial GS sequences. The authors demonstrated that reductants like dithiothreitol increase the activity of the plastidial isoform. Because during during protein extraction, dithiothreitol was added (see Materials & Methods), it is very likely that GS activity especially during the night was overestimated. Although it is known that after light-off the redox milieu of the chloroplasts changes rapidly (Dietz and Hell, 2015), the extend of change in GS activity is unknown. To mathematically assess GS activity during the night, a numerical experiment was conducted (see Fig. 2 (b)). Activity of GS during the night was intentionally reduced in steps of 10% and 10 simulations conducted. As can be seen in Fig 2b, the cumulative error of simulations was minimized for nocturnal GS activity between 20% and 50% of the non-inactivated enzyme. This agrees with the data of Choi et al., 1999, who showed that dithiothreitol increases the activity by ca. a factor of two. Thus, we added a factor in the model by which the GS can be down-regulated by the optimizer during the night.

Photorespiration in the *hpr1-1* mutant

For identification of the k_m parameter of the *HPR* reaction (Fig S2 (c)) a broad interval was set, because this reaction integrates activity of several enzymes and transporters, for which no parameter boundaries are known. First Ser is transported out of the mitochondria and then into the peroxisomes, where it is de-aminated by Serine-Glyoxylate transaminase, before hydroxypyruvate is actually reduced by the HPR reaction. Finally the product glycerate is phosphorylated by the glycerate-kinase, before it re-enters the Calvin-Benson cycle.

Notably, is that it is unclear whether in Col-0 HPR is the limiting step in the PR pathway. The identified minimum of the k_m value for *HPR* in Col-0 was set as lower boundary for simulations of the *hpr1-1* mutant to prevent compensation of the lower v_{max} in the mutant as a purely mathematical means of generating higher fluxes despite of the mutation. However, using this approach the *in silico* Ser pool was slightly higher then the measured Ser pool for the mutant at eCO_2 (Fig. 3) albeit not at aCO_2. Based on this result we decreased the lower boundaries for the k_m value at eCO_2 until decent Ser time courses are obtained. After a decrease of 30% decent results were obtained (Fig. 4 (b)). Mentionable, we decreased the lower boundary to 40% of the original value but still got the same result as by a decrease of only 30%.

Observed flux dynamics

Using the parameter sets obtained from the simulations, flux rates at time points of harvest were calculated. Fig. 6 shows fluxes for PR relevant reactions. Surprisingly, *HPR* (Fig. 6 (c)) shows high fluxes in the mutant ($P < 2e-16$) compared to wildtype. This resulted from the extremely high substrate level and indicates a metabolic state clearly different from wildtype. For instance the Ser levels at the beginning of the day are substantially higher in the mutant compared to Col-0 at aCO_2. As expected, the *HPR* flux was increased at aCO_2 ($P < 2e-16$) due to the larger PR input flux. Similar *GDC* and *SHMT* (Fig. 6 (a) and (b)) are increased ($P < 2e-16$ and $P < 2e-16$, respectively) when comparing eCO_2 to aCO_2. Moreover the *hpr1-1* mutant shows increased levels for *GDC* ($P = 1.16e-10$) compared to Col-0. In contrast the *SHMT* reaction is increased in the wildtype compared to *hpr1-1* even though not significantly.

Flux within the GS/GOGAT cycle was tightly linked to PR (Fig. 7). For *GS* and *GOGAT* (Fig. 7 (a) and (b)) a genotype and treatment effect was observed (*GS* genotype = $P < 2e-16$, *GS* treatment = $P < 2e-16$, *GOGAT* genotype = $P < 2e-16$ and *GOGAT* treatment = $P < 2e-16$). The highest flux was obtained for *hpr1-1* at aCO_2 followed by Col-0 at aCO_2. Plants grown at aCO_2 showed a higher flux compared to plants grown at eCO_2 independent of the genotype. For plants grown at eCO_2 also a slightly higher flux is observed for *hpr1-1 as* compared to Col-0, supporting an earlier finding that PR takes place even at a CO_2 concentration of 1000 ppm (Kraemer *et al.*, 2021).

In summary, the mutant showed a higher turnover in the GS/GOGAT cycle compared to the wildtype. In addition, turnover in the GS/GOGAT cycle is decreased at eCO_2. However, the fluxes of *NR* (Fig. 7 (c)) behaves differently. Analyzing the *NR* flux revealed a significant genotype and treatment effect (P = 2.09e-06 and P = 0.00110), where Col-0 shows higher fluxes compared to the mutant and the flux is elevated at eCO_2 compared to aCO_2.

Discussion

Interaction of eCO_2 and N-assimilation

Several studies have shown that eCO_2 decreases the N content of plant biomass (Bloom *et al.*, 2014; Andrews *et al.*, 2020). However, it is unclear how N acquisition is affected. In the current study, we found a higher v_{max} for NR in Col-0 at eCO_2 and consistently elevated fluxes for this condition (Fig. 7 (c)).

Two possible reasons could explain why N assimilation did not catch up with C acquisition at eCO_2. Either reduced stomatal conductance could have led to reduced nitrate availability, or the lack of reducing equivalents could have slowed down the turnover of *NR*. In contrast to observations for wheat under eCO_2 (Del Pozo *et al.*, 2007), we found no indications for reduced mineral uptake in *Arabidopsis* wildtype plants (Fig S3 (a)). However, the Nitrate pool was significantly lower in the *hpr1-1* mutant (Fig S3 (a)), pointing to a decreased uptake capability in the mutant, probably because of energetic constraints (see below). Thus our results, suppor earlier reports (Andrews *et al.*, 2019) that showed reduced total N and shoot N, no matter of the N source supplied, to accompany increased biomass formation under eCO_2. For eCO_2, we found a higher ratio of NR and GDC, which both feed the pool of ammonium used for Gln synthesis (Fig. 8 (a)). A lower GDC flux would reduce the load on the GS/GOGAT cycle, and this should allow a higher proportion of *de novo* N assimilation. Indeed, we calculated a lower flux for the GS/GOGAT cycle at eCO_2 (Fig. 7a, b), demonstrating a tight link between PR and GS/GOGAT turnover. This link has also been shown by Häusler et al., 1994 and Wallsgrove et al., 1987 , who reported that barley lacking GS activity suffered at photorespiratory conditions. In addition, it is known that even short incubation at eCO_2 result in a reduction of GS and GOGAT activity (Guo *et al.*, 2013; Wu *et al.*, 2020). Our finding that the GS/GOGAT cycle contained significantly more newly assimilated N at eCO_2 (Fig. 8 (a)) indicates that N incorporation is improved at eCO_2.

In order to analyze whether enough N is assimilated at eCO_2 for the production of biomass the ratio between the flux *HP2BMEXP* and *NR* was analyzed (Fig. 8 (b)). The wildtype and the mutant showing an increased ratio at eCO_2 when compared to plants at aCO_2. Thus, more C is used for the production of structural carbon at eCO_2. This is also substantiated by the finding that plants at eCO_2 have increased starch levels (Fig. S3 (b)). Therefore, our results are in support of the hypotheses of (Gifford et al., 2000, Kuehny et al., 1991 and Wong, 1990, stating that decreased N content of wildtype plants at eCO_2 is a result of dilution by elevated C levels due to increased PS (Taub & Wang, 2008). In the mutant, however, we found a large pool of free amino acids (Fig. S1, 4 and 5), and this agrees with earlier findings (Timm *et al.*, 2008b). This contrasts with the stunted growth, reduced biomass and chlorotic phenotype of the mutant (Timm *et al.*, 2008a; Li *et al.*, 2019). Based on our results, we propose three different explanations for this phenotype.

Phenotype of the *hpr1-1* mutant

As mentioned above, an energetic constraint could be responsible for the metabolic disturbance in the mutant. As shown in Fig. 7, the fluxes for *GS* and *GOGAT* are increased in *hpr1-1* especially at aCO_2. Thus, large amounts of ATP and reduced ferredoxin are required to sustain synthesis of Glu, which is needed for the removal of glyoxylate. Fig. 8 (a) depicts the NR-to-GDC ratio, which is highest in Col-0 at eCO_2 followed by *hpr1-1* under the same condition and the wildtype at aCO_2. Based on a TukeyHSD test the wildtype at aCO_2 differ significantly from Col-0 at eCO_2. Furthermore, is Col-0 at eCO_2 significantly different when compared to the mutant at eCO_2 and aCO_2. However, the *hpr1-1* mutant at aCO_2 lacks far behind. The lower this ratio, the more NH_4^+ has to be refixed, consuming ATP and reducing equivalents without net gain of biomass. While this can easily explain the already known (Timm *et al.*, 2008a) fact that Col-0 at eCO_2 yields the highest biomass, followed - in this order - by *hpr1-1* at eCO_2, Col-0 at aCO_2 and *hpr1-1* at $aCO_{2,}$, it leaves unclear, why large amounts of amino acids accumulate in the mutant.

Alternatively, the mutant could be poisoned by photorespiratory intermediates such as phosphoglycolate, glycolate or glyoxylate. As can be seen in Fig. 6 the GDC and SHMT fluxes are strongly increased in *hpr1-1* at aCO_2. Nevertheless, the fact that the HPR flux is substantially extended into the night shows that even after light-off photorespiratory intermediates have to be removed, which could not be metabolized during the day in spite of increased GDC and SHMT activity. Accumulation of phosphoglycolate during the day would strongly inhibit triose phosphate isomerase (Anderson, 1971), and this would block C assimilation in the Calvin-Benson cycle. However, how should a bottleneck in the last step of the PR pathway increase buildup of the early metabolites? As described in

Kraemer et al., 2021 successful simulations of our metabolic model required implementation of an additional source of the PR intermediate glycolate, and this is most likely provided by the spontaneous reaction of hydroxypyruvate with H_2O_2 that yields glycolate (Walton and Butt, 1981). This reaction is favored by the large amount of Ser in the mutant at aCO_2. Thus, more glycolate is produced in the *hpr1-1* mutant, increasing the probability of a toxic effect and also leading to an additional loss of assimilated carbon in the form of CO_2. Besides intoxication by photorespiratory intermediates, it would also be possible that an increased level of NH_4^+ (Fig. 4 (c)) could interfere with ATP production, because it is in equilibrium with ammonia that acts as an uncoupling agent of ATP synthesis.

Finally, the metabolic bottleneck created by the *hpr1-1* mutation causes large amounts of C and N being bound in the form of Gly and Ser. As a consequence, the equilibrium of free amino-acids, brought about by transamination reactions, is severely disturbed, and this could interfere with protein synthesis in the shoot as well as the supply of the roots and other sinks with AA. Not only N but also C compounds show altered distributions in *hpr1-1* as compared to the wildtype (Fig. S1, and Fig. 5). Especially the carboxylates Cit, malate and fumarate, were significantly enriched, while among soluble sugars only glucose was reduced in *hpr1-1*. While this may argue against a disturbed sucrose supply to sink organs, the low nitrate content of *hpr1-1* in aCO_2 could indicate low carbon supply to the root system. The low glucose level in the shoot could result enhanced use in the pentose phosphate pathway as suggested by Li et al., 2019, who stated that high pentose-phosphate pathway activity could provide additional CO_2 that would alleviate the PR syndrome, but restrict biomass formation. Most likely a combination of all three effects contributes to the mutant phenotype at aCO_2.

Conclusion

We showed that the *hpr1-1* mutant suffers from several limitations. Besides a high demand for ATP and reducing equivalents for the increased turnover of the GS/GOGAT cycle, a possible intoxication by photorespiratory intermediates or NH_4^+ could interfer with biomass formation, and lastly but not least, an unfavorable redistribution of N and C compounds could impact protein and, ultimately, biomass production. Our study confirms a tight link between PR and the GS/GOGAT cycle and adds to our understanding of how plant N assimilation is affected by eCO_2.

Material and Methods

Plant growth

Arabidopsis thaliana wildtype Col-0 and the mutant *hpr1-1* (SALK067724) were grown in a hydroponic culture for 50 days in a growth chamber with 8 h/16 h light/dark regime (100 μmol m^{-2} s^{-1}; 22°C/16°C). For the first 17 days plants were grown at aCO_2 (450 ±20 ppm). Afterwards half of the plants were transferred to eCO_2 (1000 ±20 ppm). The hydroponic medium was as described in Brauner et al., 2014 with the difference that no NH_4^+ was supplied and the NO_3^- concentration was reduced to 30% of the originial value. This asured sufficient, but not excess supply of N (Tab. 1). Photosynthesis measurements were conducted one week before harvesting as described by Küstner, Nägele and Heyer, 2019. Using the result of the photosynthesis measurements the photorespiration was calculated according to Sharkey, 1988. For τ^* the values published by Kraemer et al., 2021 were used.

Table 1: Test for sufficient N supply for Col-0 after growth for 50 days in hydroponic media.

% N of original Media	100%	50%	30%	20%	10%
% of plants with deficiency syndrome	46	20	0	0	0

Metabolite measurements

Gly, Ser, α-KG, glucose, fructose and sucrose were measured by quantitative GC-MS/MS. Samples were extracted using 750 µL methanol with 25 nmol ribitol as internal standard. After 15 min at 70°C followed by shaking for 10 min at RT samples were centrifuged (5 min 17,000 g). The supernatant was transfered to a new vessel, and 400 µL of H_2O were added. After incubation for 10 min at 95°C samples were agitated for 10 min at RT. Following centrifugation (5 min, 17,000g) and the supernatants were pooled. Subsequently, 300 µL H_2O and 200 µL chloroform were added. After centrifugation (2 min, 17,000 g) the two phases were seperated and the polar phase was dried in a speedvac and used for analysis. Dried samples were derivatized using 20 µL of methoxamine dissolved in pyridine (40 mg/mL) by incubation for 90 min at 30°C. Next, 80 µL N-methyl-

N-(trimethylsilyl)trifluoracetamide (MSTFA) were added and the solution was incubated for 30 min at 50°C. Metabolites were measured by gas-chromatography coupled to mass-spectrometry (GC-MS/MS). For injection, 1 µL of the derivatized sample was used. The GC-MS/MS device was a GCMS-TQ8040 (Shimadzu, Munich, Germany) using helium as carrier gas at a flow of 1.12 mL/min. The stationary phase was a 30 m Optima 5MS-0.25 µm fused silica capillary column. Injection temperature was 230°C. The transfer line and ion source were set to 250°C and 200°C, respectively. The initial temperature of the column oven was 80°C and this was increased by 15°C/min until the final temperature of 330°C was reached, which was held for 6 min. After a solvent delay of 4.6 min, spectra of the MS device were recorded in the multiple reaction mode (MRM) with specific target-ions for each metabolite. External standards were used for quantification.

Starch, hexosephosphates, carbonic acids (fumaric acid, malic acid and citric acid) and the total amino-acid pool were quantified as described by Küstner, Nägele and Heyer, 2019. NH_4^+ was quantified according to Vega-Mas, Asier and Marino, 2015. Glu and Gln were measured according to (Graham and Aprison, 1966; Pérez-de la Mora *et al.*, 1989).

Enzymatic activities

Activity of hydroxy-pyruvate-reductase was determined according to Fernie, Bauwe and Weber, 2017. GS activity was measured according to Berteli et al., 1995; Gomes Silveira et al., 2003. Briefly, protein was extracted into a 100 mM Tris-HCl (pH 7.6) buffer containing 2.5 mM Dithiothreitol and 10 mM $MgCl_2$. The assay buffer contains 125 mM Tris-HCl (pH 7.6), 5 mM ATP, 80 mM $MgSO_4$, 125 mM Hydroxylamin-NaOH (pH 7) and 100 mM Glutamate (pH 7.2). To an assay buffer volume of 80 µL, 120 µL of protein extract were added and incubated for different time-points (0, 20, 25 and 30 min). To determine the background an assay buffer was used without Hydroxylamin-NaOH (pH 7). The reaction was stopped using 60 µL of a solution consisting of 1.5 mL 10% w/v $FeCl_3$*6 H_2O in 0.2 N HCl, 1.5 mL 24% w/v Trichloracetat and 1.5 mL 20% v/v HCl. Afterwards, the absorption was determined at 540 nm. For calibration curves L-Glutaminsäure-γ-Monohydroxamat was used

All enzyme activities were determined at the beginning, middle and end of the light phase– as well as in the middle of the night. Values for the remaining time-points were calculated by spline interpolation.

Data analysis and statistics

Data evaluation, normalization, visualization and statistics were performed in Microsoft Excel® (Microsoft Office version 2010, http://www.microsoft.com) and the R software (The R Project for Statistical Computing; http://www.r-project.org/). Parameter optimization was performed in a way that *in silico* time courses matched the measured time courses best. For optimization and simulations the R-package *paropt* was used (Krämer *et al.*, 2021).

ACKNOWLEDGMENTS

The authors would like to thank Prof Hermann Bauwe for a generous gift of seeds of the *hpr1-1* mutant used in this study. Nadja Beuttenmüller and Annika Allinger are acknowledged for expert plant cultivation. K.K. received a scholarship "Landesgraduiertenförderung (LGF)" of the Federal State of Baden-Wuerttemberg (Germany).

AUTHOR CONTRIBUTIONS

Konrad Krämer: Designed the study, conducted experiments, developed the model, and wrote the manuscript. Judith Brock: Conducted experiments. Arnd G. Heyer: Designed the study and wrote the manuscript.

DATA AVAILABILITY STATEMENT

Raw data can be made available by the authors upon request.

References

Anderson, L. (1971) ‘Chloroplast And Cytoplasmic Enzymes In Pea Leaf Triose Phosphate Isomerases’, *BIOCHIMICA ET BIOPHYSICA ACTA*, 22, pp. 237–244.

Andrews, M. *et al.* (2019) ‘Elevated CO_2 effects on nitrogen assimilation and growth of C3 vascular plants are similar regardless of N-form assimilated’, *Journal of Experimental Botany*, 70(2), pp. 683–690. doi: 10.1093/jxb/ery371.

Andrews, M. *et al.* (2020) ‘Will rising atmospheric CO_2 concentration inhibit nitrate assimilation in shoots but enhance it in roots of C3 plants?’, *Physiologia Plantarum*, 170(1), pp. 40–45. doi: 10.1111/ppl.13096.

Atkinson, N. *et al.* (2017) ‘Rubisco small subunits from the unicellular green alga Chlamydomonas complement Rubisco-deficient mutants of Arabidopsis’, *New Phytologist*, 214(2), pp. 655–667. doi: 10.1111/nph.14414.

Berardini, T. Z. *et al.* (2015) ‘The arabidopsis information resource: Making and mining the “gold standard” annotated reference plant genome’, *Genesis*, 53(8), pp. 474–485. doi: 10.1002/dvg.22877.

Berteli, F. *et al.* (1995) ‘Salt stress increases ferredoxin‐dependent glutamate synthase activity and protein level in the leaves of tomato’, *Physiologia Plantarum*, 93(2), pp. 259–264. doi: 10.1111/j.1399-3054.1995.tb02226.x.

Bloom, A. J. *et al.* (2014) ‘Nitrate assimilation is inhibited by elevated CO_2 in field-grown wheat’, *Nature Climate Change*, 4(6), pp. 477–480. doi: 10.1038/nclimate2183.

Bloom, A. J. (2015) ‘Photorespiration and nitrate assimilation: A major intersection between plant carbon and nitrogen’, *Photosynthesis Research*, 123(2), pp. 117–128. doi: 10.1007/s11120-014-0056-y.

Bloom, A. J., Kasemsap, P. and Rubio-Asensio, J. S. (2020) ‘Rising atmospheric CO_2 concentration inhibits nitrate assimilation in shoots but enhances it in roots of C3 plants’, *Physiologia Plantarum*, 168(4), pp. 963–972. doi: 10.1111/ppl.13040.

Brauner, K. *et al.* (2014) ‘Exaggerated root respiration accounts for growth retardation in a starchless mutant of Arabidopsis thaliana’, *Plant Journal*, 79(1), pp. 82–91. doi: 10.1111/tpj.12555.

Choi, Y. A., Kim, S. G. and Kwon, Y. M. (1999) ‘The plastidic glutamine synthetase activity is directly modulated by means of redox change at two unique cysteine residues’, *Plant Science*, 149(2), pp. 175–182. doi: 10.1016/S0168-9452(99)00163-6.

Cousins, A. B. and Bloom, A. J. (2004) ‘Oxygen consumption during leaf nitrate assimilation in a C3 and C4 plant: The role of mitochondrial respiration’, *Plant, Cell and Environment*, 27(12), pp. 1537–1545. doi: 10.1111/j.1365-3040.2004.01257.x.

Dellero, Y. *et al.* (2016) ‘Photorespiratory glycolate – glyoxylate metabolism’, *Journal of Experimental Botany*, 67(10), pp. 3041–3052. doi: 10.1093/jxb/erw090.

Dietz, K. J. and Hell, R. (2015) ‘Thiol switches in redox regulation of chloroplasts: Balancing redox state, metabolism and oxidative stress’, *Biological Chemistry*, 396(5), pp. 483–494. doi: 10.1515/hsz-2014-0281.

Fernie, A. R., Bauwe, H. and Weber, A. P. M. (2017) *Photorespiration Methods and Protocols*, *Methods in Molecular Biology*. Springer Nature. doi: 10.1007/978-1-4939-7225-8_12.

Gauthier, P. P. G. *et al.* (2010) ‘In folio isotopic tracing demonstrates that nitrogen assimilation into glutamate is mostly independent from current CO_2 assimilation in illuminated leaves of Brassica napus’, *New Phytologist*, 185(4), pp. 988–999. doi: 10.1111/j.1469-8137.2009.03130.x.

Gifford, R. M., Barrett, D. J. and Lutze, J. L. (2000) ‘The effects of elevated CO_2 on the C:N and C:P mass ratios of plant tissues’, *Plant and Soil*, 224(1), pp. 1–14. doi: 10.1023/A:1004790612630.

Gomes Silveira, J. A. *et al.* (2003) ‘Proline accumulation and glutamine synthetase activity are increased by salt-induced proteolysis in cashew leaves’, *Journal of Plant Physiology*, 160(2), pp. 115–123. doi: 10.1078/0176-1617-00890.

Graham, L. T. and Aprison, M. H. (1966) ‘Fluorometric Determination of Aspartate, Glutamate, and gamma-Aminobutyrate in Nerve Tissue Using Enzymic Methods’, *Analytical Biochemistry*, 15(3), pp. 487–497.

Guan, X. Q. *et al.* (2004) ‘Photoprotective function of photorespiration in several grapevine cultivars under drought stress’, *Photosynthetica*, 42(1), pp. 31–36. doi: 10.1023/B:PHOT.0000040566.55149.52.

Guo, H. *et al.* (2013) ‘Elevated CO_2 modifies N acquisition of Medicago truncatulaby enhancing N fixation and reducing nitrate uptake from soil’, *PLoS ONE*, 8(12). doi: 10.1371/journal.pone.0081373.

Häusler, R. E., Lea, P. J. and Leegood, R. C. (1994) ‘Control of photosynthesis in barley leaves with reduced activities of glutamine synthetase or glutamate synthase - II. Control of electron transport and CO_2 assimilation’, *Planta*, 194(3), pp. 418–435. doi: 10.1007/BF00197543.

Hindmarsh, A. C. *et al.* (2005) ‘SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers’, *ACM Transactions on Mathematical Software*, 31(3), pp. 363–396. doi: 10.1145/1089014.1089020.

Huma, B. *et al.* (2018) ‘Stoichiometric analysis of the energetics and metabolic impact of photorespiration in C3 plants’, *Plant Journal*, 96(6), pp. 1228–1241. doi: 10.1111/tpj.14105.

IPCC (2014) ‘Summary for Policymakers’, *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, pp. 1–33. doi: 10.1017/CBO9781107415324.

Jauregui, I. *et al.* (2016) ‘Root-shoot interactions explain the reduction of leaf mineral content in Arabidopsis plants grown under elevated [CO_2] conditions’, *Physiologia Plantarum*, pp. 65–79. doi: 10.1111/ppl.12417.

Koffler, B. E. *et al.* (2013) ‘High resolution imaging of subcellular glutathione concentrations by quantitative immunoelectron microscopy in different leaf areas of Arabidopsis’, *Micron*, 45, pp. 119–128. doi: 10.1016/j.micron.2012.11.006.

Kraemer, K. *et al.* (2021) ‘Acclimation to elevated CO_2 affects the C / N balance by reducing de novo N-assimilation’, *Physiologia Plantarum*, (August 2021), pp. 1–13. doi: 10.1111/ppl.13615.

Krämer, K. *et al.* (2021) ‘paropt: Parameter Optimizing of ODE-Systems’. Available at: https://cran.r-project.org/package=paropt.

Kuehny, J. S. *et al.* (1991) ‘Nutrient dilution by starch in CO_2-enriched chrysanthemum’, *Journal of Experimental Botany*, 42(6), pp. 711–716. doi: 10.1093/jxb/42.6.711.

Küstner, L., Nägele, T. and Heyer, A. G. (2019) ‘Mathematical modeling of diurnal patterns of carbon allocation to shoot and root in Arabidopsis thaliana’, *npj Systems Biology and Applications*, 5(1), pp. 1–11. doi: 10.1038/s41540-018-0080-1.

Li, J. *et al.* (2019) ‘A cytosolic bypass and g6p shunt in plants lacking peroxisomal hydroxypyruvate reductase1’, *Plant Physiology*, 180(2), pp. 783–792. doi: 10.1104/pp.19.00256.

Nägele, T. *et al.* (2010) ‘Mathematical Modeling of the Central Carbohydrate Metabolism in Arabidopsis Reveals a Substantial Regulatory Influence of Vacuolar Invertase on Whole Plant Carbon Metabolism’, *Plant Physiology*, 153(1), pp. 260–272. doi: 10.1104/pp.110.154443.

Nunes-Nesi, A., Fernie, A. R. and Stitt, M. (2010) ‘Metabolic and signaling aspects underpinning the regulation of plant carbon nitrogen interactions’, *Molecular Plant*, 3(6), pp. 973–996. doi: 10.1093/mp/ssq049.

Pérez-de la Mora, M. *et al.* (1989) ‘A glutamate dehydrogenase-based method for the assay of l-glutamic acid: Formation of pyridine nucleotide fluorescent derivatives’, *Analytical Biochemistry*, 180(2), pp. 248–252. doi: 10.1016/0003-2697(89)90425-9.

Del Pozo, A. *et al.* (2007) ‘Gas exchange acclimation to elevated CO_2 in upper-sunlit and lower-shaded canopy leaves in relation to nitrogen acquisition and partitioning in wheat grown in field chambers’, *Environmental and Experimental Botany*, 59(3), pp. 371–380. doi: 10.1016/j.envexpbot.2006.04.009.

Rebeille, F., Neuburger, M. and Douce, R. (1994) ‘Interaction between glycine decarboxylase, serine hydroxymethyltransferase and tetrahydrofolate polyglutamates in pea leaf mitochondria’, *Biochemical Journal*, 302(1), pp. 223–228. doi: 10.1042/bj3020223.

Sharkey, T. D. (1988) ‘Estimating the rate of photorespiration in leaves’, *Physiologia Plantarum*, 73(1), pp. 147–152. doi: 10.1111/j.1399-3054.1988.tb09205.x.

Shih, P. M. *et al.* (2015) ‘Biochemical characterization of predicted Precambrian RuBisCO’, *Nature Communications*, 7. doi: 10.1038/ncomms10382.

Stitt, M. and Krapp, a (1999) ‘The interaction between elevated carbon dioxide and nitrogen nutrition: the physiological and molecular background’, *Plant, Cell and Environment*, 22, pp. 553–621. doi: 10.1046/j.1365-3040.1999.00386.x.

Szulejko, J. E. *et al.* (2017) ‘Global warming projections to 2100 using simple CO2 greenhouse gas modeling and comments on CO_2 climate sensitivity factor’, *Atmospheric Pollution Research*, 8(1), pp. 136–140. doi: 10.1016/j.apr.2016.08.002.

Taub, D. R. and Wang, X. (2008) ‘Why are nitrogen concentrations in plant tissues lower under elevated CO_2? A critical examination of the hypotheses’, *Journal of Integrative Plant Biology*, 50(11), pp. 1365–1374. doi: 10.1111/j.1744-7909.2008.00754.x.

Tcherkez, G. *et al.* (2009) ‘In folio respiratory fluxomics revealed by 13C isotopic labeling and H/D isotope effects highlight the noncyclic nature of the tricarboxylic acid “cycle” in illuminated leaves’, *Plant Physiology*, 151(2), pp. 620–630. doi: 10.1104/pp.109.142976.

Timm, S. *et al.* (2008a) ‘A cytosolic pathway for the conversion of hydroxypyruvate to glycerate during photorespiration in Arabidopsis’, *Plant Cell*, 20(10), pp. 2848–2859. doi: 10.1105/tpc.108.062265.

Timm, S. *et al.* (2011) ‘The hydroxypyruvate-reducing system in arabidopsis: Multiple enzymes for the same end’, *Plant Physiology*, 155(2), pp. 694–705. doi: 10.1104/pp.110.166538.

Timm, S. *et al.* (2021) ‘Metabolite profiling in arabidopsis thaliana with moderately impaired photorespiration reveals novel metabolic links and compensatory mechanisms of photorespiration’, *Metabolites*, 11(6). doi: 10.3390/metabo11060391.

Vega-Mas, I., Asier, S. and Marino, D. (2015) ‘High-throughput Quantification of Ammonium Content in Arabidopsis’, *bio-protocol*, 5, pp. 14–17.

Voss, I. *et al.* (2013) ‘Emerging concept for the role of photorespiration as an important part of abiotic stress response’, *Plant Biology*, 15(4), pp. 713–722. doi: 10.1111/j.1438-8677.2012.00710.x.

Wallsgrove, R. M. *et al.* (1987) ‘Barley Mutants Lacking Chloroplast Glutamine Synthetase —Biochemical and Genetic Analysis’, *Plant Physiology*, 83(1), pp. 155–158. doi: 10.1104/pp.83.1.155.

Walton, N. J. and Butt, V. S. (1981) ‘Metabolism and decarboxylation of glycollate and serine in leaf peroxisomes’, *Planta*, 153(3), pp. 225–231. doi: 10.1007/BF00383891.

Wong, S. C. (1990) ‘Elevated atmospheric partial pressure of CO_2 and plant growth - II. Non-structural carbohydrate content in cotton plants and its effect on growth parameters’, *Photosynthesis Research*, 23(2), pp. 171–180. doi: 10.1007/BF00035008.

Wu, F. *et al.* (2020) ‘Response of nitrogen metabolism in masson pine needles to elevated CO2’, *Forests*, 11(4), pp. 1–13. doi: 10.3390/F11040390.

Figures

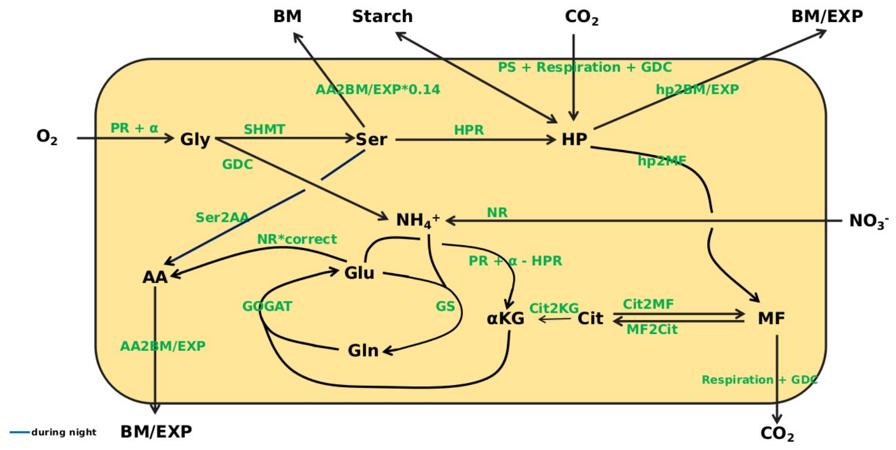


Figure 1: Overview of the model. In black are the states indicated. Outside system boundary: O_2, CO_2, BM, Starch, EXP, and NO_3^-. Within the system boundary: Gly, Ser, HP, MF, Cit, α-KG, Glu, Gln, AA and NH_4^+. For the following fluxes mass balance kinetics are used: hp2BM/EXP, hp2MF, Cit2MF, MF2Cit, Cit2KG, Ser2AA, AA2BM/EXP. The remaining reactions are represented as Michaelis-Menten kinetics.

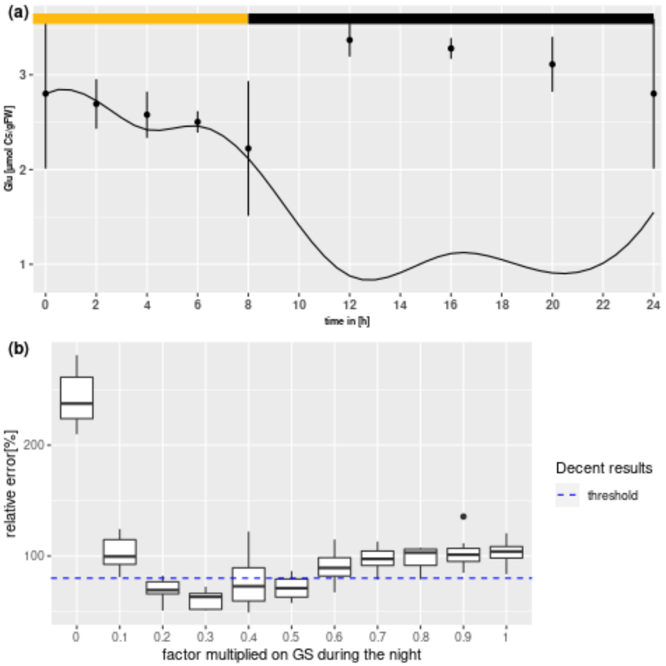


Figure 2: GS inactivation. (a) Diurnal course of Glu for Col-0 at ambient CO_2 concentrations. Dots represent mean of measured values (n = 5 per group). Bares indicate the error of measurements. Line represent the mean of 10 simulations. Light phase indicated by yellow bar and dark phase indicated by black bar. (b) Error plotted against the factor multiplied on GS during the night. Each boxplot represents 10 simulations for Col-0 at ambient CO_2 concentrations. The dashed line indicates the threshold at which decent results for all states can be found.

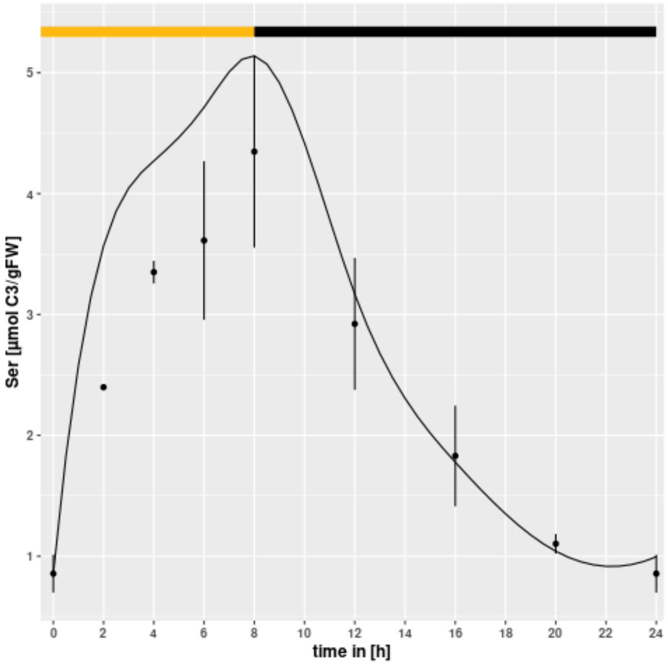


Figure 3: PR in hpr1-1 aCO$_2$. Diurnal course of Ser for hpr1-1 at eCO$_2$ concentrations. Dots represent mean of measured values (n = 5 per group). Bares indicate the error of measurements. Line represent the mean of 20 simulations using the minimum of the km value of the simulations of Col-0 aCO$_2$ as lower boundary. Light phase indicated by yellow bar and dark phase indicated by black bar.

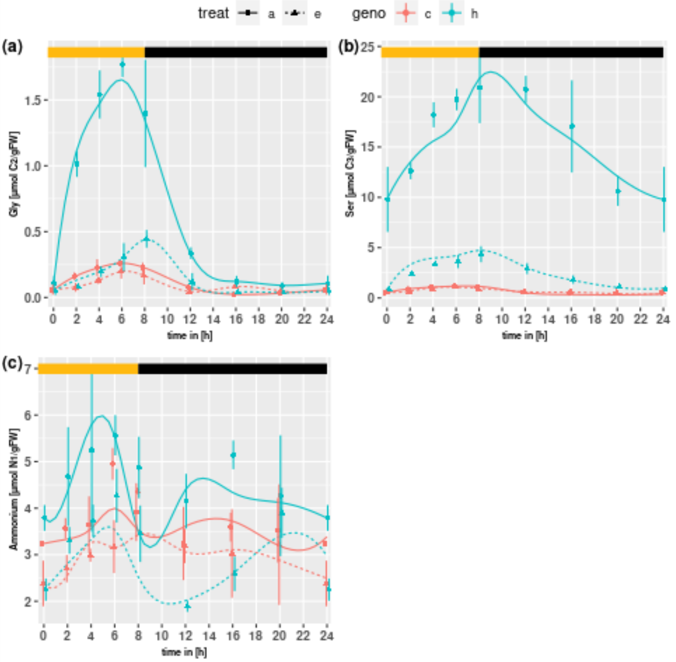


Figure 4: Diurnal course of photorespiration intermediates [(a) Gly, (b) Ser] and ammonium [(c)]. In red the wildtype and in blue the hpr1-1 mutant. Dashed lines and triangles represent elevated CO_2 *concentrations. Continous lines and closed circles represent ambient* CO_2 *concentrations. Dots or triangles represent mean of measured values (n = 5 per group). Bares indicate the error of measurements. Line represent the mean of 20 simulations. Light phase indicated by yellow bar and dark phase indicated by black bar.*

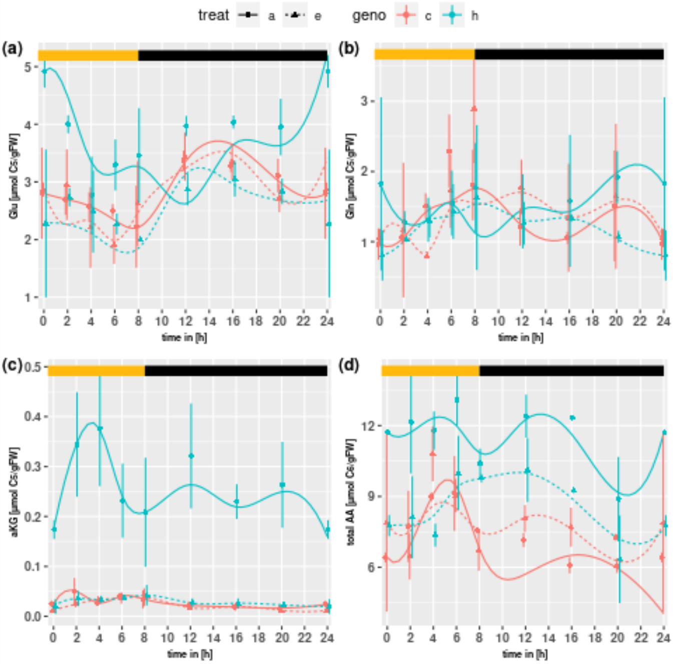


Figure 5: Diurnal course of (a) Glu, (b) Gln, α-KG (c) and total AA (d). In red the wildtype and in blue the hpr1-1 mutant. Dashed lines and triangles represent elevated CO_2 concentrations. Continous lines and closed circles represent ambient CO_2 concentrations. Dots or triangles represent mean of measured values (n = 5 per group). Bares indicate the error of measurements. Line represent the mean of 20 simulations. Light phase indicated by yellow bar and dark phase indicated by black bar.

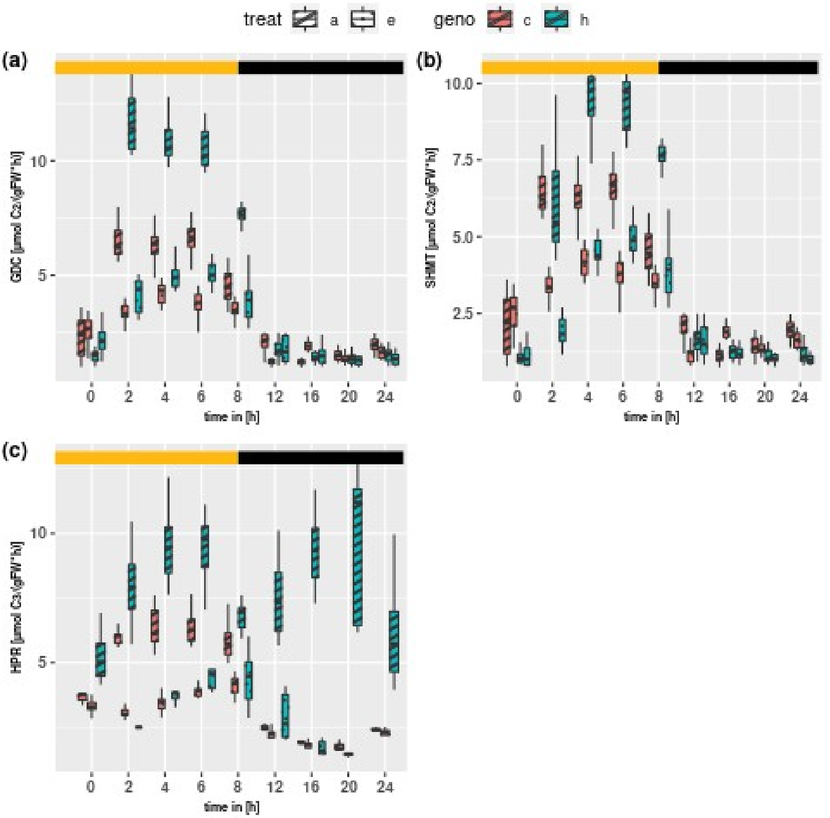


Figure 6: Diurnal course of fluxes for GDC (a), SHMT (b) and HPR (c). In red the wildtype and in blue the hpr1-1 mutant. Dashed boxplots represent ambient CO_2 concentrations. Dotted boxplots represent elevated CO_2 concentrations. Data based on 20 simulations. Light phase indicated by yellow bar and dark phase indicated by black bar. Boxplots are dodged by 0.25 h in order to prevent overlapping.

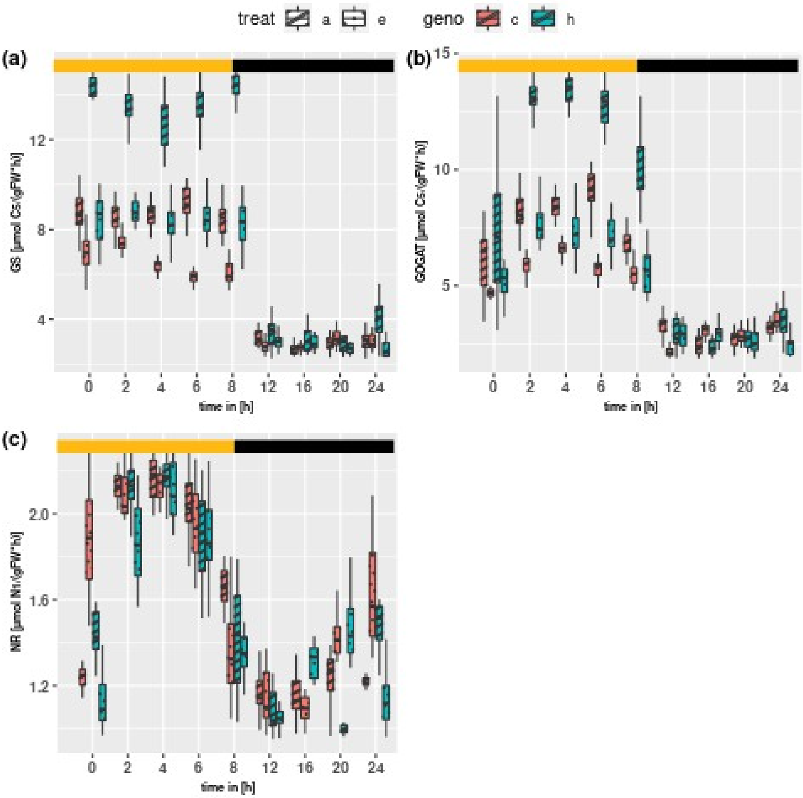


Figure 7: Diurnal course of fluxes for GS (a), GOGAT (b) and NR (c). In red the wildtype and in blue the hpr1-1 mutant. Dashed boxplots represent ambient CO_2 concentrations. Dotted boxplots represent elevated CO_2 concentrations. Data based on 20 simulations. Light phase indicated by yellow bar and dark phase indicated by black bar. Boxplots are dodged by 0.25 h in order to prevent overlapping.

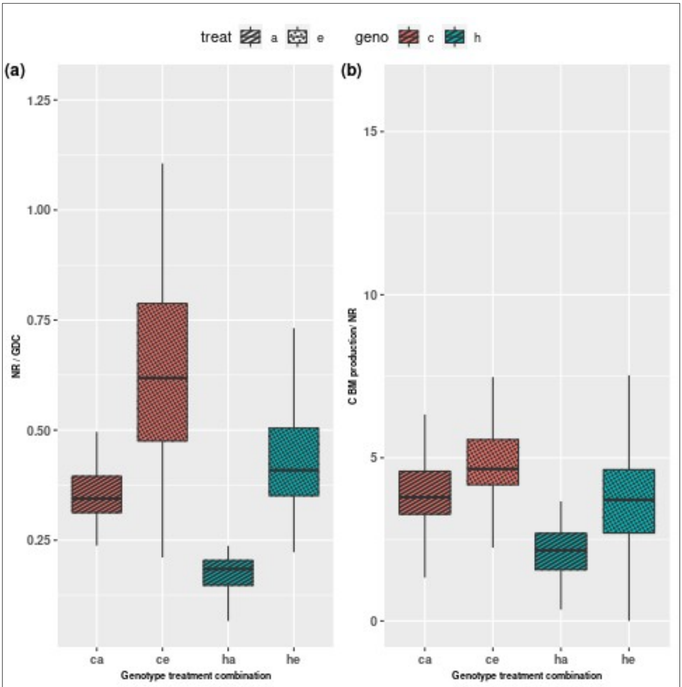


Figure 8: Ratio of NR to GDC during the day (a). Ratio of HP2BMEXP to NR during the day (b). Dashed boxplots represent ambient CO_2 concentrations. Dotted boxplots represent elevated CO_2 concentrations. Data based on 20 simulations. Boxplots are dodged by 0.25 h in order to prevent overlapping.

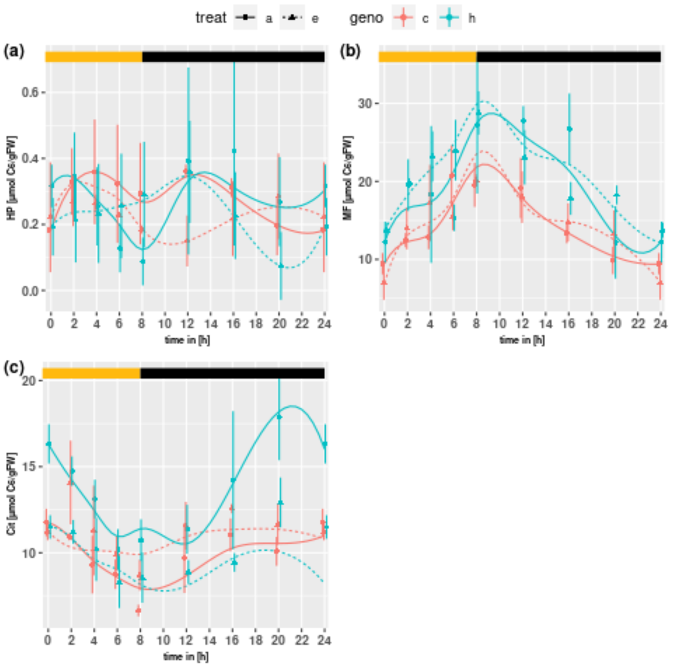


Figure S1: Diurnal course of (a) HP, (b) MF, and Cit (c). In red the wildtype and in blue the hpr1-1 mutant. Dashed lines and triangles represent elevated CO_2 concentrations. Continous lines and closed circles represent ambient CO_2 concentrations. Dots or triangles represent mean of measured values (n = 5 per group). Bares indicate the error of measurements. Line represent the mean of 20 simulations. Light phase indicated by yellow bar and dark phase indicated by black bar.

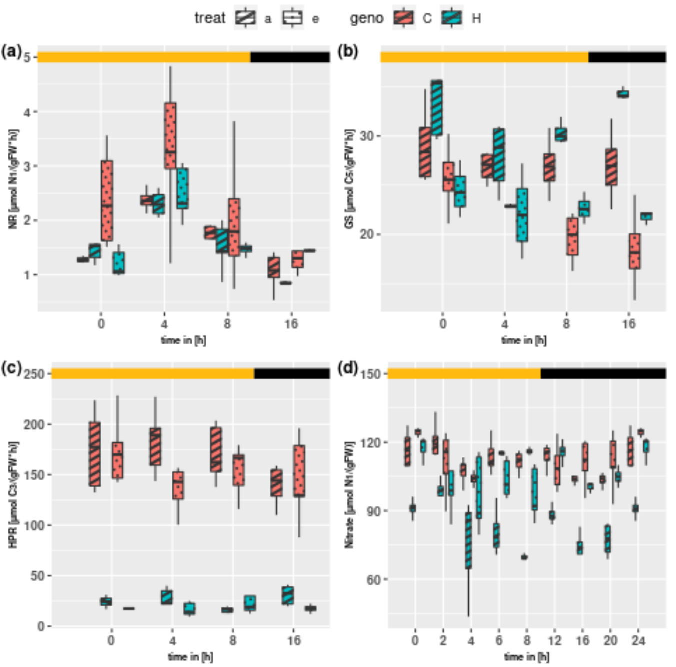


Figure S2: Diurnal course of measured activities (at least n = 5 per group) for (a) NR, (b) GS, and HPR (c). In red the wildtype and in blue the hpr1-1 mutant. Dashed boxplots represent ambient CO_2 concentrations. Dotted boxplots represent elevated CO_2 concentrations. Light phase indicated by yellow bar and dark phase indicated by black bar. Boxplots are dodged by 0.25 h in order to prevent overlapping.

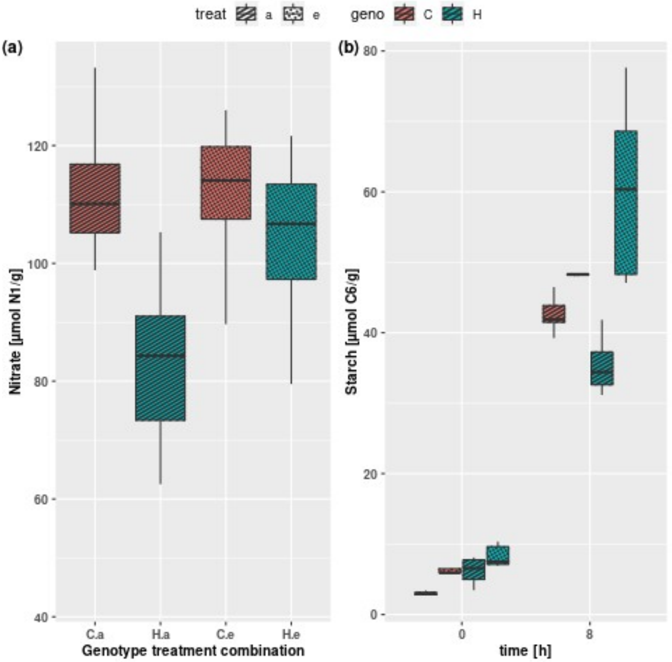


Figure S3: Nitrate conc. (n = 40) (a) Starch conc. at the beginning ad end of day (n = 5 per group) (b). In red the wildtype and in blue the hpr1-1 mutant. Dashed boxplots represent ambient CO_2 concentrations. Dotted boxplots represent elevated CO_2 concentrations. Light phase indicated by yellow bar and dark phase indicated by black bar. Boxplots are dodged by 0.25 h in order to prevent overlapping.

Package ‘paropt’

January 24, 2022

Type Package

Title Parameter Optimizing of ODE-Systems

Version 0.2

Date 2021-05-26

Author Krämer Konrad [aut, cre],
Krämer Johannes [aut],
Heyer Arnd [ths],
University of Stuttgart [uvp],
Institute of Biomaterials and Biomolecular Systems at the University of Stuttgart [his]
| file AUTHORS

Maintainer Krämer Konrad `<Konrad_kraemer@yahoo.de>`

Copyright file COPYRIGHTS

BugReports `https://github.com/Konrad1991/paropt`

Description Enable optimization of parameters of ordinary differential equations. Therefore, using 'SUNDIALS' to solve the ODE-System (see Hindmarsh, Alan C., Peter N. Brown, Keith E. Grant, Steven L. Lee, Radu Serban, Dan E. Shumaker, and Carol S. Woodward. (2005) <doi:10.1145/1089014.1089020>). Furthermore, for optimization the particle swarm algorithm is used (see: Akman, Devin, Olcay Akman, and Elsa Schaefer. (2018) <doi:10.1155/2018/9160793> and Sengupta, Saptarshi, Sanchita Basak, and Richard Peters. (2018) <doi:10.3390/make1010010>). The ODE-System has to be passed as 'Rcpp'-function. The information for the parameter boundaries and states are conveyed using data.frames.

License GPL-3 | file LICENSE

Imports Rcpp (>= 1.0.4)

LinkingTo Rcpp, RcppArmadillo, RcppThread, testthat

Suggests knitr, rmarkdown, xml2

VignetteBuilder knitr

RoxygenNote 7.1.2

Encoding UTF-8

R topics documented:

optimizer 2
optimizer_pointer 4
solve_ode_system 6
solve_ode_system_pointer 7

Index **10**

`optimizer` *Optimize parameters of ode-systems*

Description

Optimize parameters used in an ode equation in order to match values defined in the state-data.frame

Usage

```
optimizer(
  integration_times,
  ode_system,
  relative_tolerance,
  absolute_tolerances,
  lb,
  ub,
  states,
  npop,
  ngen,
  error,
  solvertype
)
```

Arguments

`integration_times`	a vector containing the time course to solve the ode-system (see Details for more Information)
`ode_system`	the ode-system which will be integrated by the solver (see Details for more Information).
`relative_tolerance`	a number defining the relative tolerance used by the ode-solver.
`absolute_tolerances`	a vector containing the absolute tolerance(s) for each state used by the ode-solver.
`lb`	a data.frame containing the lower bounds for the parameters (see Details for more Information).

`ub`	a data.frame containing the upper bounds for the parameters (see Details for more Information).
`states`	a data.frame containing the predetermined course of the states (see Details for more Information).
`npop`	a number defining the number of particles used by the Particle Swarm Optimizer.
`ngen`	a number defining the number of generations the Particle Swarm Optimizer (PSO) should run.
`error`	a number defining a sufficient small error. When the PSO reach this value optimization is stopped.
`solvertype`	a string defining the type of solver which should be used (bdf, ADAMS, ERK or ARK. see Details for more Information).

Details

The vector containing the time course to solve the ode-system should contain the same entries as the time vector in the state-data.frame (it can be also be a different variable instead of time).

The ode system should be a Rcpp-function with a specific signature Rcpp::NumericVector ode(double time, std::vector<double> parameter, Rcpp::NumericVector states). The first entry defines the time point when the function is called. The second argument defines the parameter which should be optimized. There exist two different types of parameters. Parameters can be either constant or variabel. In order to calculate a variable parameter at a specific timepoint the Catmull-Rom-Spline is used. This vector contains the already interpolated parameters at the specific time-point, in the same order as defined in the data.frames containing the lower- and upper-boundaries. The last argument is a vector containing the states in the same order as defined in the data.frame containing the state-information. Thus, it is obligatory that the state-derivates in the ode-system are in the same order defined as in the data.frame. Furthermore, it is mandatory that the function return a Rcpp::NumericVector with the same dimension as the input vector containing the states. The resulting vector has to contain the right hand side of the ode-system.

For constant parameters use only the first row (below the headers) if other parameters are variable use “NA“ in the following rows for the constant parameters.

For variable parameters at least four points are needed. If a variable parameter is not available at every time point use “NA“ instead.

The two data.frames containg lower and upper-boundaries need the parameter in the same order.

The data.frame containing the state information should hold the time course in the first column. The header-name time is compulsory. The following columns contain the states. Take care that the states are in the same order defined in the ode system. If a state is not available use “NA“. This is possible for every time points except the first one. The ode solver need a start value for each state which is extracted from the first row of this file (below the headers).

The error between the solver output and the measured states is the sum of the absolute differences divided by the number of time points. It is crucial that the states are in the same order in the text file cointaining the state-information and in the ode-system to compare the states correctly!

For solving the ode system the SUNDIALS Software is used (https://computing.llnl.gov/projects/sundials). The last argument defines the solver-type which is used during optimization: “bdf“, “ADAMS“, “ERK“ or “ARK“. bdf = Backward Differentiation Formulas, ADAMS = Adams-Moulton, ERK =

explicite Runge-Kutta and ARK = implicite Runge-Kutta. All solvers are used in the NORMAL-Step method in a for-loop using the time-points defined in the text-file containing the states as output-points. The bdf- and ARK-Solver use the SUNLinSol_Dense as linear solver. Notably here is that for the ARK-Solver the ode system is fully implicit solved (not only part of it).

Examples can be found in the vignette.

`optimizer_pointer` *Optimize parameters of ode-systems*

Description

Optimize parameters used in an ode equation in order to match values defined in the state-data.frame

Usage

```
optimizer_pointer(
  integration_times,
  ode_sys,
  relative_tolerance,
  absolute_tolerances,
  lower,
  upper,
  states,
  npop,
  ngen,
  error,
  solvertype
)
```

Arguments

`integration_times`	a vector containing the time course to solve the ode-system (see Details for more Information)
`ode_sys`	the ode-system which will be integrated by the solver (see Details for more Information).
`relative_tolerance`	a number defining the relative tolerance used by the ode-solver.
`absolute_tolerances`	a vector containing the absolute tolerance(s) for each state used by the ode-solver.
`lower`	a data.frame containing the lower bounds for the parameters (see Details for more Information).
`upper`	a data.frame containing the upper bounds for the parameters (see Details for more Information).

`states`	a data.frame containing the predetermined course of the states (see Details for more Information).
`npop`	a number defining the number of particles used by the Particle Swarm Optimizer.
`ngen`	a number defining the number of generations the Particle Swarm Optimizer (PSO) should run.
`error`	a number defining a sufficient small error. When the PSO reach this value optimization is stopped.
`solvertype`	a string defining the type of solver which should be used (bdf, ADAMS, ERK or ARK. see Details for more Information).

Details

The vector containing the time course to solve the ode-system should contain the same entries as the time vector in the state-data.frame (it can be also be a different variable instead of time).

The ode system should be of type Rcpp::XPtr<OS>. The OS is predefined in the package. The function should possess the following signature: int ode(double &time, std::vector<double> &parameter, std::vector<double> &states). The first entry defines the time point when the function is called. The second argument defines the parameter which should be optimized. There exist two different types of parameters. Parameters can be either constant or variabel. In order to calculate a variable parameter at a specific timepoint the Catmull-Rom-Spline is used. This vector contains the already interpolated parameters at the specific time-point, in the same order as defined in the data.frames containing the lower- and upper-boundaries. The last argument is a vector containing the states in the same order as defined in the data.frame containing the state-information. Thus, it is obligatory that the state-derivates in the ode-system are in the same order defined as in the data.frame. Within the function the new states have to be saved in the states-vector. Please be aware that when using the approach with the Rcpp::XPtr the optimization is run in parallel. Thus, the function has to be thread-safe (among other things don't use any R Code)!

For constant parameters use only the first row (below the headers) if other parameters are variable use “NA“ in the following rows for the constant parameters.

For variable parameters at least four points are needed. If a variable parameter is not available at every time point use “NA“ instead.

The two data.frames containg lower and upper-boundaries need the parameter in the same order.

The data.frame containing the state information should hold the time course in the first column. The header-name time is compulsory. The following columns contain the states. Take care that the states are in the same order defined in the ode system. If a state is not available use “NA“. This is possible for every time points except the first one. The ode solver need a start value for each state which is extracted from the first row of this file (below the headers).

The error between the solver output and the measured states is the sum of the absolute differences divided by the number of time points. It is crucial that the states are in the same order in the text file cointaining the state-information and in the ode-system to compare the states correctly!

For solving the ode system the SUNDIALS Software is used (https://computing.llnl.gov/projects/sundials). The last argument defines the solver-type which is used during optimization: “bdf“, “ADAMS“, “ERK“ or “ARK“. bdf = Backward Differentiation Formulas, ADAMS = Adams-Moulton, ERK = explicite Runge-Kutta and ARK = implicite Runge-Kutta. All solvers are used in the NORMAL-Step method in a for-loop using the time-points defined in the text-file containing the states as output-points. The bdf- and ARK-Solver use the SUNLinSol_Dense as linear solver. Notably here is that for the ARK-Solver the ode system is fully implicit solved (not only part of it).

Examples can be found in the vignette.

`solve_ode_system` *Solves ode-system and compare result to measured states*

Description

Solves ode-system and compare result to measured states

Usage

```
solve_ode_system(
  integration_times,
  ode_system,
  relative_tolerance,
  absolute_tolerances,
  start,
  states,
  solvertype
)
```

Arguments

`integration_times`	a vector containing the time course to solve the ode-system (see Details for more Information)
`ode_system`	the ode-system which will be integrated by the solver (see Details for more Information).
`relative_tolerance`	a number defining the relative tolerance used by the ode-solver.
`absolute_tolerances`	a vector containing the absolute tolerance(s) for each state used by the ode-solver.
`start`	a data.frame containing a parameter-set (see Details for more Information).
`states`	a data.frame containing the predetermined course of the states (see Details for more Information).
`solvertype`	a string defining the type of solver which should be used (bdf, ADAMS, ERK or ARK. see Details for more Information).

Details

The vector containing the time course to solve the ode-system should contain the same entries as the time vector in the state-data.frame (it can be also be a different variable instead of time).

The ode system should be a Rcpp-function with a specific signature Rcpp::NumericVector ode(double time, std::vector<double> parameter, Rcpp::NumericVector states). The first entry defines the time point when the function is called. The second argument defines the parameter which should be optimized. There exist two different types of parameters. Parameters can be either constant or variabel. In order to calculate a variable parameter at a specific timepoint the Catmull-Rom-Spline is used. This vector contains the already interpolated parameters at the specific time-point, in the same order as defined in the data.frames containing the lower- and upper-boundaries. The last argument is a vector containing the states in the same order as defined in the data.frame containing the state-information. Thus, it is obligatory that the state-derivates in the ode-system are in the same order defined as in the data.frame. Furthermore, it is mandatory that the function return a Rcpp::NumericVector with the same dimension as the input vector containing the states. The resulting vector has to contain the right hand side of the ode-system.

For constant parameters use only the first row (below the headers) if other parameters are variable use “NA“ in the following rows for the constant parameters.

For variable parameters at least four points are needed. If a variable parameter is not available at every time point use “NA“ instead.

The data.frame containing the state information should hold the time course in the first column. The header-name time is compulsory. The following columns contain the states. Take care that the states are in the same order defined in the ode system. If a state is not available use “NA“. This is possible for every time points except the first one. The ode solver need a start value for each state which is extracted from the first row of this file (below the headers).

The error between the solver output and the measured states is the sum of the absolute differences divided by the number of time points. It is crucial that the states are in the same order in the text file cointaining the state-information and in the ode-system to compare the states correctly!

For solving the ode system the SUNDIALS Software is used (https://computing.llnl.gov/projects/sundials). The last argument defines the solver-type which is used during optimization: “bdf“, “ADAMS“, “ERK“ or “ARK“. bdf = Backward Differentiation Formulas, ADAMS = Adams-Moulton, ERK = explicite Runge-Kutta and ARK = implicite Runge-Kutta. All solvers are used in the NORMAL-Step method in a for-loop using the time-points defined in the text-file containing the states as output-points. The bdf- and ARK-Solver use the SUNLinSol_Dense as linear solver. Notably here is that for the ARK-Solver the ode system is fully implicit solved (not only part of it).

Examples can be found in the vignette.

`solve_ode_system_pointer` *Solves ode-system and compare result to measured states*

Description

Solves ode-system and compare result to measured states

Usage

```
solve_ode_system_pointer(
  integration_times,
  fctptr,
  relative_tolerance,
  absolute_tolerances,
  start,
  states,
  solvertype
)
```

Arguments

`integration_times`	a vector containing the time course to solve the ode-system (see Details for more Information)
`fctptr`	is a pointer to the ode-system which will be integrated by the solver (see Details for more Information).
`relative_tolerance`	a number defining the relative tolerance used by the ode-solver.
`absolute_tolerances`	a vector containing the absolute tolerance(s) for each state used by the ode-solver.
`start`	a data.frame containing a parameter-set (see Details for more Information).
`states`	a data.frame containing the predetermined course of the states (see Details for more Information).
`solvertype`	a string defining the type of solver which should be used (bdf, ADAMS, ERK or ARK. see Details for more Information).

Details

The vector containing the time course to solve the ode-system should contain the same entries as the time vector in the state-data.frame (it can be also be a different variable instead of time).

The ode system should be of type Rcpp::XPtr<OS>. The OS is predefined in the package. The function should possess the following signature: int ode(double &time, std::vector<double> &parameter, std::vector<double> &states). The first entry defines the time point when the function is called. The second argument defines the parameter which should be optimized. There exist two different types of parameters. Parameters can be either constant or variabel. In order to calculate a variable parameter at a specific timepoint the Catmull-Rom-Spline is used. This vector contains the already interpolated parameters at the specific time-point, in the same order as defined in the data.frames containing the lower- and upper-boundaries. The last argument is a vector containing the states in the same order as defined in the data.frame containing the state-information. Thus, it is obligatory that the state-derivates in the ode-system are in the same order defined as in the data.frame. Within the function the new states have to be saved in the states-vector.

For constant parameters use only the first row (below the headers) if other parameters are variable use “NA“ in the following rows for the constant parameters.

For variable parameters at least four points are needed. If a variable parameter is not available at every time point use “NA“ instead.

The data.frame containing the state information should hold the time course in the first column. The header-name time is compulsory. The following columns contain the states. Take care that the states are in the same order defined in the ode system. If a state is not available use “NA“. This is possible for every time points except the first one. The ode solver need a start value for each state which is extracted from the first row of this file (below the headers).

The error between the solver output and the measured states is the sum of the absolute differences divided by the number of time points. It is crucial that the states are in the same order in the text file cointaining the state-information and in the ode-system to compare the states correctly!

For solving the ode system the SUNDIALS Software is used (https://computing.llnl.gov/projects/sundials). The last argument defines the solver-type which is used during optimization: “bdf“, “ADAMS“, “ERK“ or “ARK“. bdf = Backward Differentiation Formulas, ADAMS = Adams-Moulton, ERK = explicite Runge-Kutta and ARK = implicite Runge-Kutta. All solvers are used in the NORMAL-Step method in a for-loop using the time-points defined in the text-file containing the states as output-points. The bdf- and ARK-Solver use the SUNLinSol_Dense as linear solver. Notably here is that for the ARK-Solver the ode system is fully implicit solved (not only part of it).

Examples can be found in the vignette.

Index

`optimizer`, 2
`optimizer_pointer`, 4

`solve_ode_system`, 6
`solve_ode_system_pointer`, 7

paropt

Konrad Krämer

- How to use paropt
- Using a Rcpp-function
- Using an external pointer to a C++ function
- State-input
- Parameter-input
- What happens during an evaluation of a parameterset
- Result of simulations
- The arguments for function optimizer and optimizer_pointer
- Particle swarm optimizer (PSO)
- References

How to use paropt

The package paropt is built to optimize the parameters of ode-systems. The aim is to match the output of the **in silico** solution to previously measured states. The user has to supply an ode-system in the form of an Rcpp-function or as an external pointer to a C++ function. The information about states and parameters are passed on as data.frames. In this vignette, a predator-prey system is used as an example to demonstrate how the functions of 'paropt' can be used.

Using a Rcpp-function

If using a Rcpp-function the following signature has to be fulfilled: 'Rcpp::NumericVector ode(double t, std::vector params, Rcpp::NumericVector y)'. The first entry defines the time point on which the function is called. The second argument defines the parameter which will be optimized. Notably, the parameters are in the same order as in the data.frames containing the information about the boundaries. The last argument is a vector containing the states in the same order as defined in the data.frame containing the state-information. Thus, the state-derivates in the ode-system must be defined in the same order as in the data.frame. Furthermore, it is mandatory that the function return a Rcpp::NumericVector with the same dimension as the input vector containing the states. The vector should contain the right-hand side of the ode-system.

```
#include <Rcpp.h>

// [[Rcpp::export]]
Rcpp::NumericVector ode_system(double t, std::vector<double> params,
                               Rcpp::NumericVector states) {
  Rcpp::NumericVector states_deriv(2);
  double a = params[0];
  double b = params[1];
  double c = params[2];
  double d = params[3];
  double predator = states[0];
  double prey = states[1];
  double ddtpredator = states_deriv[0] =
          predator*c*prey - predator*d;
  double ddtprey = states_deriv[1] =
         prey*a - prey*b*predator;
  return states_deriv;
}
/*** R
path <- system.file("examples",
        package = "paropt")

# states
df <- read.table(paste(path,
      "/states_LV.txt", sep = ""),
      header = TRUE)

# parameter
lb <- data.frame(time = 0, a = 0.8,
      b = 0.3, c = 0.09, d = 0.09)
ub <- data.frame(time = 0, a = 1.3,
      b = 0.7, c = 0.4, d = 0.7)

# Optimizing
library(paropt)
set.seed(1)
output <- optimizer(integration_times = df$time,
                    ode_system = ode_system,
                    relative_tolerance = 1e-6,
                    absolute_tolerances = c(1e-8, 1e-8),
                    lb = lb, ub = ub, states = df,
                    npop = 40, ngen = 1000,
                    error = 0.0001, solvertype = "bdf")
kableExtra::kbl(output)

# Plots
par(mfrow = c(2,1))
plot(df$time, output$States[,1],
      pch = 19, type = 'l',
      ylab = "predator",
      xlab = "time",
      ylim = c(0, 30))
points(df$time,
      states$n1, pch = 19,
      col = "darkred", type = 'p')
legend(80, 30,
  legend=c("in silico", "measured"),
  col=c("black", "darkred"), lty=1, cex=0.8)

plot(df$time, output$States[,2],
      pch = 19, type = 'l',
      ylab = "prey",
      xlab = "time", ylim = c(0, 65))
points(df$time,
      states$n2,
      pch = 19, col = "darkred",
      type = 'p')
legend(80, 60,
       legend=c("in silico", "measured"),
       col=c("black", "darkred"),
       lty=1, cex=0.8)

# Solving
optimized_parameter <-
    as.data.frame(output$Parameter)
names(optimized_parameter) <-
    c("time", "a", "b", "c", "d")
output_solver <- solve_ode_system(
      integration_times = df$time,
      ode_system = ode_system,
      relative_tolerance = 1e-6,
      absolute_tolerances = c(1e-8, 1e-8),
      start = optimized_parameter,
      states = df, solvertype = "bdf")
output_solver
*/
```

Using an external pointer to a C++ function

If using an external pointer to a C++ function, the following signature is mandatory: int ode(double &t, std::vector& params,std::vector& states). The arguments are in principle the same as in the Rcpp-function approach. However, the right-hand side has to be stored in the input vector states. For instance, one could first extract the content of the state vector by storing it in suitable variables. Afterwards, the vector states can be filled with the right-hand side information which is defined by using the previously defined variables. The order of the states and parameters has to be the same as in the data.frames to calculate the error correctly and use the correct parameters.

It is crucial that the function is thread-safe!

```
// [[Rcpp::depends(RcppArmadillo)]]
#include <RcppArmadillo.h>
// [[Rcpp::depends(paropt)]]
// [[Rcpp::plugins(cpp11)]]

typedef int (*OS)(double &t,
             std::vector<double> &params,
             std::vector<double> &states);

int ode_system(double &t,
               std::vector<double> &params,
               std::vector<double> & states) {

  double a = params[0];
  double b = params[1];
  double c = params[2];
  double d = params[3];
  double predator = states[0];
  double prey = states[1];
  states[0] = predator*c*prey - predator*d;
  states[1] = prey*a - prey*b*predator;
  return 0;
}

// [[Rcpp::export]]
Rcpp::XPtr<OS> test_optimization() {
  Rcpp::XPtr<OS> xpfun = Rcpp::XPtr<OS>(new OS(&ode_system));
  return xpfun;
}

/*** R
#states
path <- system.file("examples",
            package = "paropt")
states <- read.table(
        paste(path,"/states_LV.txt", sep = ""),
        header = T)

# parameter
lb <- data.frame(time = 0, a = 0.8,
        b = 0.3, c = 0.09, d = 0.09)
ub <- data.frame(time = 0, a = 1.3,
          b = 0.7, c = 0.4, d = 0.7)

# Optimizing
library(paropt)
set.seed(1)
df <- optimizer_pointer(
      integration_times = states$time,
      ode_sys = test_optimization(),
      relative_tolerance = 1e-6,
      absolute_tolerances = c(1e-8, 1e-8),
      lower = lb, upper = ub, states = states,
      npop = 40, ngen = 1000, error = 0.0001,
      solvertype = "bdf")

df
# Plots
par(mfrow = c(2,1))
plot(df$time, output$States[,1],
      pch = 19, type = 'l',
      ylab = "predator",
      xlab = "time",
      ylim = c(0, 30))
points(df$time,
      states$n1, pch = 19,
      col = "darkred", type = 'p')
legend(80, 30,
  legend=c("in silico", "measured"),
  col=c("black", "darkred"), lty=1, cex=0.8)

plot(df$time, output$States[,2],
      pch = 19, type = 'l',
      ylab = "prey",
```

Table 1: Example for states

time	predator	prey
0	10.0000000	10.0000000
2	0.5195882	0.2421567
4	0.4386158	0.2367276
6	0.4670779	0.1002269
8	0.8659140	0.6587734
10	0.5328101	0.1171064
12	0.5941835	NA
14	0.7752122	0.4206482
16	0.7011195	0.7855264
18	0.4713157	0.0121704
20	0.2323933	0.9040727
22	0.3483769	0.8297733
24	0.0598382	0.0510475

```
      xlab = "time", ylim = c(0, 65))
points(df$time,
      states$n2,
      pch = 19, col = "darkred",
      type = 'p')
legend(80, 60,
       legend=c("in silico", "measured"),
       col=c("black", "darkred"),
       lty=1, cex=0.8)
# Solving
optimized_parameter <-
    as.data.frame(df$Parameter)
names(optimized_parameter) <-
    c("time", "a", "b", "c", "d")

output_solver <- solve_ode_system_pointer(
        integration_times = states$time,
        fctptr = test_optimization(),
        relative_tolerance = 1e-6,
        absolute_tolerances = c(1e-8, 1e-8),
        start = optimized_parameter,
        states = states, solvertype = "bdf")

output_solver
*/
```

State-input

The information of the states has to be supplied as a data.frame. The first column must be namee *time*. This column contains the independent variable across which the solver integrates (it does not have to be the time. In this document it is always called time). It is followed by the information of the states at the specific time points. Notably, the order of the states is the same as in the ode-system to correctly calculate the error. If a state is not available at all time points use 'NA' to ignore this state for error calculation at the specified time point. However, the first line below the header cannot contain 'NA' as it contains the start values for the ode-solver.

Table 2: Example for the lower boundaries

time	const	variable
0	1	1
2	NA	2
4	NA	3
6	NA	4
8	NA	5
10	NA	6
12	NA	7
14	NA	8
16	NA	9
18	NA	10
20	NA	11
22	NA	12
24	NA	13

Table 3: Example for the upper boundaries

time	const	variable
0	2	20
2	NA	20
4	NA	20
6	NA	20
8	NA	20
10	NA	20
12	NA	20
14	NA	20
16	NA	20
18	NA	20
20	NA	20
22	NA	20
24	NA	20

Parameter-input

Constant parameters do not change their value during the integration of the ode-system. The boundaries for these parameters have to be defined in the first row. Notably, the first column of the data.frames containing the lower or upper boundaries is the time column. The name 'time' is mandatory for this column. If a parameter is not constant, at least four different time-points are needed. The information for the parameter at the time-points are fed into an interpolation-function to calculate values at each time-point the ode-system is called. The interpolation is conducted in a wrapper-function around the actual ode-system. Thus, the parameters passed on to the ode-system are the previously splined values for the specific time-point. For instance if the ode-system is called at t = 3.5 the parameter 'variable' is not defined. In this case the parameters have to be interpolated. This is conducted using a Catmull-Rom-Spline. The parameter-vector passed to the ode-system already contains the splined parameters at timepoint t.

What happens during an evaluation of a parameterset

During the optimization the optimizer creates a bunch of possible solutions within the parameter boundaries. Each solution is passed to the ode-solver which integrates along the time and returns the states at the time-points specified in the data.frame containing the state-information. The *in silico* solution is compared to the measured states to evaluate the parameterset. The error used is the sum of the absolute differences between *in silico* and measured states, divided by the number of timepoints.

Result of simulations

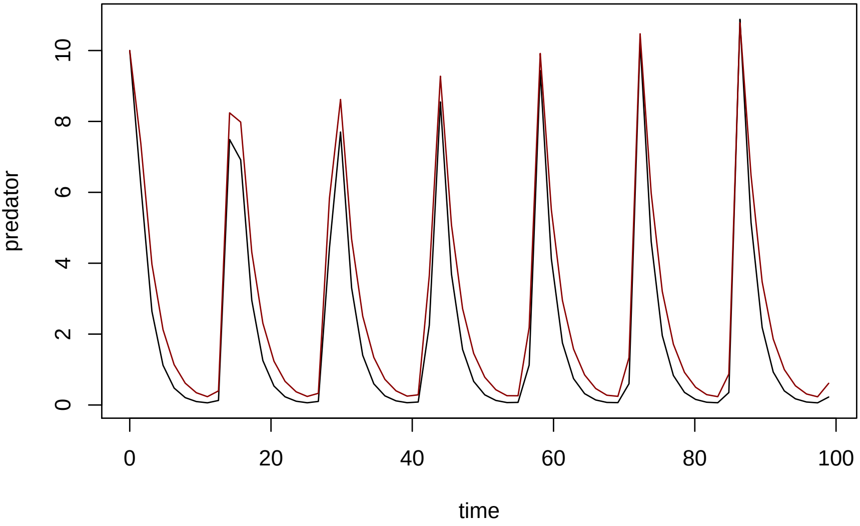


Figure 1: Result for the predator data. Black: in silico values, Red: measured values

The arguments for function optimizer and optimizer_pointer

```
optimizer(integration_times, ode_system,
          relative_tolerance, absolute_tolerances,
          lb, ub, states,
          npop, ngen, error, solvertype)
```

The first argument is the time-vector containing either the same information as defined in the time-column defined in the data.frame containing the states (see Table1) or only a *subset* (it can only be shortened at the end).
It is mandatory to start with the first entry of the time-column, however it is possible to stop at a certain time-point before the last one. Thus, it is possible to optimize only a part of the problem.

The next argument is the compiled ode-system followed by the relative tolerance and the absolute tolerances that are used by the ode-solver. These are followed by the data.frames defining the lower and upper boundaries of the parameter. Next the data.frame containing the state information is passed to the function.

To optimize the parameters a particle swarm optimizer (PSO) is used. Therefore, the number of particles (npop = 40 is a good starting point for many problems) and the number of generations (ngen) have to be passed to the function. The number of generations defines the maximum number of generations the PSO should run. However, if the PSO finds a suitable parameter set which has an error below or equal a threshold defined by the user it stops. This threshold is defined by the error-argument.

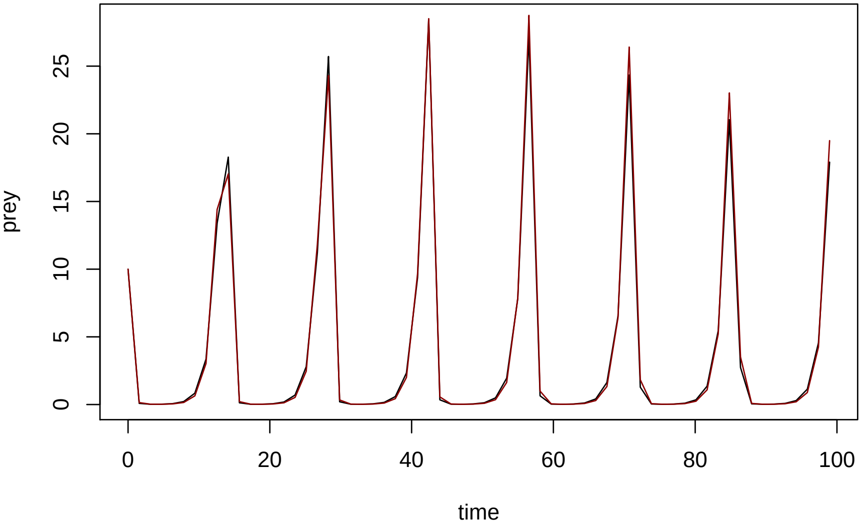


Figure 2: Result for the prey data. Black: in silico values, Red: measured values

The last argument defines the type of solver to be used. Four different solver exist: ‘bdf’, ‘ADAMS’, ‘ERK’ and ‘ARK. All solvers are part of the SUNDIALS project. For details see ’https://computing.llnl.gov/projects/sundials’ and Hindmarsh et al. (2005). The bdf solver is the backward differential formular solver of CVODE and is best suited for stiff problems. It uses a dense matrix module (SUNDense_Matrix) and the corresponding nonlinear solver (SUNLinsol_Dense). The ADAMS solver is the ADAMS-MOULTON solver of CVODE and is most suitable for non-stiff problems. The ‘ERK’ solver is an explicit Runge-Kutta solver which is like the ADAMS-MOULTON solver used for non-stiff-problems. The ‘ARK’ solver is a fully implicite Runge-Kutta-solver that uses the same matrix and nonlinear solver module as ‘bdf’. The integration itself occurs in a for-loop using the ‘CV_NORMAL’ step-function for all four solvers. If integration for a specific parameter set is not possible the error is set to 1.79769e+308 (which is the maximum of a double). If you want to test a specific parameter set just call the function ode_solving. The function requires the same parameter as the optimizer. Naturally, the arguments for the PSO, the error-threshold as well as the parameter lower- and upper-bounds are not needed.

In principle, the optimizer_pointer and solve_ode_system_pointer do the same as the functions optimizer and solve_ode_system. However, they use an external pointer to a C++ function. Thus, they show a better performance. Notably, it was necessary to change the PSO function for optimizer_pointer in order to permit parallelization. Therefore even with the same seed, the results differ between optimizer and optimizer_pointer.

Particle swarm optimizer (PSO)

This PSO-implementation is a modified version of ‘https://github.com/kthohr/optim’ (for a general overview see Sengupta, Basak, and Peters (2018)). The PSO has several key features. First of all, a bunch (number of particles defined by the user) of possible parameter sets is created within the boundaries defined by the user. Each parameter set is called a particle. All particles together are called the swarm. Each possible parameter set is passed to the solver which integrates the system. The result is used to calculate the error. Thus, each particle has a current solution and a current personal best value. Furthermore, each particle possesses a neighborhood that consists of 0-3 other particles (for details see Akman, Akman, and Schaefer (2018)). The PSO uses a combination of its history (personal best value) and the information of the best particle of the neighborhood to change its current value. Notably, in this package, a randomly adaptive topology is used. This means, that the neighborhood is recalculated each time when the global best solution (best solution of the entire swarm) has not improved within one generation.

References

Akman, Devin, Olcay Akman, and Elsa Schaefer. 2018. "Parameter Estimation in Ordinary Differential Equations Modeling via Particle Swarm Optimization." *Journal of Applied Mathematics* 2018. https://doi.org/10.1155/2018/9160793.

Hindmarsh, Alan C., Peter N. Brown, Keith E. Grant, Steven L. Lee, Radu Serban, Dan E. Shumaker, and Carol S. Woodward. 2005. "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers." *ACM Transactions on Mathematical Software* 31 (3): 363–96. https://doi.org/10.1145/1089014.1089020.

Sengupta, Saptarshi, Sanchita Basak, and Richard Peters. 2018. "Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives." *Machine Learning and Knowledge Extraction* 1 (1): 157–91. https://doi.org/10.3390/make1010010.

Package ‘ast2ast’

January 19, 2022

Type Package

Title Translates a R Function into an External Pointer to a C++ Function

Version 0.1

Date 2022-01-04

Author Krämer Konrad [aut, cre]

Maintainer Krämer Konrad <`konrad_kraemer@yahoo.de`>

BugReports `https://github.com/Konrad1991/ast2ast`

URL `https://github.com/Konrad1991/ast2ast`

Description Translates a R function into a C++ function. An external pointer to the C++ function is returned to the user.

License GPL-2

Imports Rcpp (>= 1.0.4), purrr, R6, RcppXPtrUtils

LinkingTo Rcpp, RcppArmadillo, RcppXPtrUtils

VignetteBuilder knitr

Suggests knitr, rmarkdown, tinytest, microbenchmark, r2sundials

Encoding UTF-8

RoxygenNote 7.1.2

R topics documented:

translate 2

Index **5**

`translate` *Translates a R function into a C++ function and returns an external pointer (XPtr) to this function. Further information can be found in the vignette: 'Detailed Documentation'.*

Description

Translates a R function into a C++ function and returns an external pointer (XPtr) to this function. Further information can be found in the vignette: 'Detailed Documentation'.

Usage

```
translate(f, verbose = FALSE, reference = FALSE)
```

Arguments

`f`	The function which should be translated from R to C++.
`verbose`	If set to true the output of RcppXPtrUtils::cppXPtr is printed.
`reference`	If set to true the arguments are passed by reference.

Details

The following types are supported:

1. numeric vectors
2. numeric matrices

Variables can be either numeric vectors or matrices. Notably, it is possible that the variable change the type within the function. **It is possible to declare a variable of a scalar numeric data type. This is done by adding '_db' to the end of the variable. Each time '_db' is found the variable is declared as a scalar numeric data type. In this case the object cannot change its type!**

The following functions are supported:

1. assignment: = and <-
2. allocation: vector and matrix
3. information about objects: length and dim
4. Basic operations: +, -, *, /
5. Indices: [] and at
6. mathematical functions: sin, asin, sinh, cos, acos, cosh, tan, atan, tanh, log, ^ and exp
7. concatenate objects: c
8. comparison: ==, !=, >, <, >= and <=
9. printing: print
10. returning objects: return
11. catmull-rome spline: cmr

12. to get a range of numbers the ':' function can be used

Some details about the implemented functions

- allocation of memory works: Following forms are possible: vector(size_of_elements), vector(value, size_of_elements), vector(other_vec, size_of_other_vec), matrix(nrows, ncols), matrix(value, nrows, ncols) and matrix(vector, nrows, ncols). The latter fills the matrix or the vector with the specified 'value'.
- For indices squared brackets '[]' can be used as common in R. **Despite the results of calculations cannot be used!** Beyond that the function 'at' exists which accepts as first argument a variable and as the second argument you pass the desired index. The caveat of using 'at' is that only one entry can be accessed. Whereas '[]' can return more then one element. **The 'at'function returns a reference to the vector entry. Therefore variable[index] can behave differently then at(variable, index). The function has to be use carefully when 'at' is used. Especially if '[]' and 'at' are mixed the function behaviour is difficult to predict. Please test it before using in a serious project.**
- For loops can be written as used in R
  - Nr.1
    for(index in variable){
    # do whatever
    }
  - Nr.2
    for(index in 1:length(variable){
    # do whatever
    }
- Be aware that it is not possible to assign the result of a comparison to a variable.
- The print function accepts either a scalar, vector, matrix, string, bool or nothing (empty line).
- In order to return an object use the 'return' function (The last object is not returned automatically as in R).
- In order to interpolate values the 'cmr' function can be used. The function needs three arguments.
  1. the first argument is the point of the independent variable (x) for which the dependent variable should be calculated (y). This has to be a vector of length one.
  2. the second argument is a vector defining the points of the independent variable (x). This has to be a vector of at least length four.
  3. the third argument is a vector defining the points of the dependent variable (y). This has to be a vector of at least length four.

Be aware that the R code is translated to ETR. An expression template library which tries to mimic R. However, it does not behave exactly like R! Please check your compiled function before using it in a serious project. If you want to see how ast2ast differs from R in detail check the vignette: 'Detailed Documentation'.

Value

The external pointer of the generated C++ function

Examples

```
#Further examples can be found in the vignette: 'Detailed Documentation'.
#Hello World
## Not run:
f <- function() { print("Hello World!")}
pointer_to_f_cpp <- ast2ast::translate(f)
Rcpp::sourceCpp(code = "
#include <Rcpp.h>
typedef void (*fp)();

// [[Rcpp::export]]
void call_fct(Rcpp::XPtr<fp> inp) {
  fp f = *inp;
  f();
}
")
call_fct(pointer_to_f_cpp)

## End(Not run)
```

Index

`translate`, 2

Detailed Documentation

Konrad Krämer

- Documentation
    - Objects
    - Variable declaration
    - Basic arithmetics
    - Subsetting
    - Helper functions
    - Comparison functions
    - Control flow
    - Printing
    - Math functions
    - Interpolation

Documentation

The package *ast2ast* translates an R function into a C++ function and returns an external pointer (XPtr) to this function. The scope of *ast2ast* is to generate functions that can be used during solving ode-systems (derivative function or jacobian function) or during optimization. More generally, the translated function can be used in fields where it is necessary to evaluate a function very often. Especially when the function is evaluated by C++ the generated external pointer is very sufficient.
First of all, the supported objects and functions listed below are explained in detail. Here, the arguments which have to be passed to the functions are described and it is explained what the function returns. Furthermore, for each function, a small example is given showing how to use it. Moreover, it is explained how the function differs from R equivalents. If another differences are detected please report them.
This paragraph explains how the examples in this vignette are executed. First of all, an Rcpp function is created, which executes the output of the function *translate*. If you want to know how these functions work in detail you can go to the vignette *Information for Package authors*. In the examples, only the R code is shown to show how to write the code.

Supported objects:

- vectors (containing numbers)
- matrices (containing numbers)

Supported functions:

- assignment: = and <-
- allocation: vector and matrix
- information about objects: length and dim
- Basic operations: +, -, *, /
- Indices: [] and at
- mathematical functions: sin, asin, sinh, cos, acos, cosh, tan, atan, tanh, log, ^ and exp
- concatenate objects: c
- control flow: for, if, else if, else
- comparison: ==, !=, >, <, >= and <=
- printing: print
- returning objects: return

Objects

There exist two containers that can be used in *ast2ast* functions. Both containers can only hold the *numeric* type of R (which is equivalent to double). The first container is a vector and the second one is a matrix. **It is possible to declare a variable of a scalar numeric data type. This is done by adding _db (e.g. varname_db) to the end of the variable. Each time *_db* is found the variable is declared as a scalar numeric data type. In this case, the object cannot change its type!**
It is pivotal to follow the rules of variable naming in C++. For instance, it is not allowed to use '.' in variable names. Moreover, the following functions are implemented in C++ and thus it is not possible to use these names:

getlength	VVSIN	VVacos	VVtanh
getattributes	sinus	acosinus	tangensh
is_matrix	VVsinh	VVCOSH	VVMINUS
VEC	sinush	cosinush	VSMINUS
at	VVasin	VVtan	SVMINUS
d2i	asinus	tangens	VVPLUS
i2d	VVCOS	VVatan	VSPLUS
ass	cosinus	atangens	SVPLUS

VVTIMES	It	exp	matrix
VSTIMES	STORE	combine	NA
SVTIMES	for_	coca	NA
VVDIV	li	cd	NA
VSDIV	cmr	colon	NA
SVDIV	VVEXP	length	NA
subassign	VVlog	dim	NA
subset	ln	vector	NA

Features of *ast2ast*

- The variables can change the type within a function. This is usually not possible when using C++.
- The index of vectors and matrices starts at 1 as in R.
- The index has to be in the boundaries of the vector or matrix. Even though this is different from the behavior in R it is a nice feature. If you access an element outside the boundaries of a vector in R *NA* is returned.
- The memory of the matrices is arranged columnwise as in R.
- In R arguments passed to a function are always copied. In *ast2ast* functions, it is possible to pass only the memory address of an object (called a reference). To do this, you have to set the *reference* parameter of the *translate* function to *TRUE*. If you pass a function by reference you can modify the object without returning it (see **Example 1**). In the Rcpp function, the variable x is printed before and after the call of the function *fetr*. Notably, if no *return* is used in the R code translated by *ast2ast* nothing is returned (in R the last object is returned in this case). You see that x is 10 before the call of the function and it is 1 after the call of the function. But the function does not return anything. Thus, the object x is modified in the function without copying it.

Example 1

```
f <- function(variable) {
  variable <- 1
}
library(ast2ast)
fetr <- translate(f)
x <- 10
output <- byref(fetr, x)
```

```
## x before call of function:
## 10
## x after call of function:
## 1
```

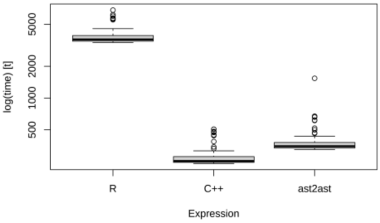


Figure 1: Benchmark

```
output
```

```
## NULL
```

Caveats:

- Sometimes large overhead of the containers
  - Variables which are scalars are represented as vectors of length 1. This is also how R handles scalar variables. As in C++ scalar variables are not defined as vectors the speed of the translated R function can be substantially lower compared to a native C++ function.

Variable declaration

In **Example 2** the various ways of declaring variables are presented. To assign a value to a variable you can use <- or =. As already mentioned only numeric values are possible. If you want to assign a vector you can use either the *c* or *vector* function. The *c* function works in the same way as R and can handle any combinations of scalars, vectors or matrices. The function *vector* differs in two ways from the R equivalent. First of all, you cannot use terms such as *vector(length = size)* as this is not possible in C++. In contrast, you just write *vector(size)*. The R function *rep* is not available in *ast2ast* but it is possible to write *vector(value, size)* which in R would be written as *rep(value, size)*. A third way to use the *vector* function is to pass another vector and the size e.g. *vector(other_vector, size)*. The *matrix* function works in the same way as the *vector* function. However, instead of the size, two arguments are needed the number of rows and the number of columns.

Example 2

```
f <- function() {
  a <- 1
  a_db <- 3.14
  b = 2
  c <- c(1, 2, 3)
  d = vector(2)
  e <- vector(3.14, 4)
  f <- vector(c, 3)
  g <- matrix(2, 2)
  h <- matrix(6, 2, 2)
  i <- matrix(e, 2, 2)

  print("a")
  print(a)
  print(a_db)
  print()
  print("b")
  print(b)
  print()
  print("c")
  print(c)
  print()
  print("d")
  print(d)
  print()
  print("e")
  print(e)
  print()
  print("f")
  print(f)
  print()
  print("g")
  print(g)
  print()
  print("h")
  print(h)
  print()
  print("i")
  print(i)
  print()
}
library(ast2ast)
fetr <- translate(f)
vardec(fetr)
```

Basic arithmetics

As usual in R it is possible to use basic arithmetic operations on scalars, vectors and matrices (**Example 3**).

Example 3

```
f <- function() {

a <- 2
b <- 3
print("scalar operations")
print(a + b)
print(a - b)
print(a / b)
print(a * b)

print()

print("vector & scalar operations")
a <- c(1, 2, 3)
b <- 4
print(a + b)
print(b - a)

print()

print("2 vectors (same length)")
a <- 6:8
b <- 1:3
print(a / b)
a <- 1:6
b <- 1:3
print(a / b)
print("2 vectors (different length)")
print("multiple of each other")
a <- 1:6
b <- 1:3
print(a / b)
print("not a multiple of each other")
a <- 1:5
b <- 1:3
print(a / b) # different to R no warning

print()

print("matrix & scalar operations")
a <- 3
b <- matrix(3, 2, 2)
print(a*b)
print(b + 4)

print()

print("matrix & vector operations")
a <- 5:6
b <- matrix(3, 2, 2)
print(b - a)
print(a / b)
```

```r
print()

print("matrix & matrix operations")
a <- matrix(3, 2, 2)
b <- matrix(4, 2, 1) # difference to R!
print(a + b)

print()

print("mixed operations")
a <- 1
b <- 2:5
c <- matrix(50, 2, 2)
d <- a + b - c/2
print(d)
}

library(ast2ast)
fetr <- translate(f)
call_fct(fetr)
```

Subsetting

If you want to subset a vector or a matrix object you can use either *[]* or the *at* function. The *[]* is slower than *at* but more powerful (**Example 4**).

The following objects can be passed to *[]* when using a vector or matrix:

- nothing
- numeric scalar
- logical
- vector
- matrix
- result of comparison
- **caveat: it is not possible to pass the results of calculations!**

In case of a matrix, it is possible to pass one of the above objects to access specific rows or columns respectively ([rows, cols]).

In contrast to *[]*, the *at* function accepts only a scalar or two scalars for vectors or matrices, respectively. Thus, only a single element is accessed by this function! However, this function works faster. The result of *at* cannot be subsetted further. The *at* function returns the numeric type which is used when a variable is declared with the extension *_db*.

Example 4

```r
f <- function() {

print("pass nothing")
a <- 1:8
print(a)
a[] <- 100
print(a)
print()

print("pass logical")
a <- 1:8
print(a)
a[TRUE] <- 100
print(a)
print()

print("pass scalar")
a <- 1:8
print(a)
a[1] <- 100
print(a)
print()


print("pass vector")
a <- 1:8
b <- 2:5
print(a)
a[b] <- 100
print(a)
print()

print("pass result of ==")
a <- 1:8
a[a < 5] <- 100
print(a)
print()


print("pass result of !=")
a <- 1:8
b <- c(1, 2, 3, 0, 0, 0, 0, 8)
a[a != b] <- 100
print(a)
print()


print("pass result of <=")
a <- 1:8
b <- c(1, 2, 3, 0, 0, 0, 0, 8)
a[a <= b] <- 100
print(a)
print()
print("pass result of >=")
a <- 1:8
b <- c(1, 2, 3, 0, 0, 0, 0, 9)
a[a >= b] <- 100
print(a)
print()


print("pass result of >")
a <- 1:8
b <- c(0, 2, 3, 0, 0, 0, 0, 9)
a[a > b] <- 100
print(a)
print()


print("pass result of <")
a <- 1:8
b <- c(0, 2, 3, 0, 0, 0, 0, 9)
a[a < b] <- 100
print(a)
print()


print("pass scalar, scalar")
a <- matrix(3, 4, 4)
a[1, 1] <- 100
print(a)
print()


print("pass vector, vector")
a <- matrix(3, 4, 4)
b <- c(1, 3)
c <- c(2, 4)
a[b, c] <- 100
print(a)
print()


print("pass ==, >=")
a <- matrix(1:16, 4, 4)
b <- 1:4
c <- c(1, 8, 3, 8)
a[b == c, b >= c] <- 100
print(a)
print()


print("at")
a <- 1:16
at(a, 2) <- 100
print(a)
print()


print("at")
a <- matrix(1:16, 4, 4)
at(a, 1, 4) <- 100
print(a)
print()
}

library(ast2ast)
fetr <- translate(f)
call_fct(fetr)
```

Helper functions

There exist three helper function. The *length* function returns the number of elements of a vector or matrix. The *dim* function returns the number of rows and columns of a matrix. The *:* function can be used to create a range of numbers. For example 1:3 creates a vector with the elements 1, 2 and 3 or 1.1:5.3 returns a vector with the elements 1.1, 2.1, 3.1, 4.1 and 5.1. See **Example 5** in order to see how the functions can be applied.

Example 5

```
f <- function() {
  a <- 1:4
  print(a)
  a <- 1.1:5.2
  print(a)

  a <- 1:16
  print(length(a))

  b <- matrix(1:4, 2, 2)
  print(dim(b))
}

library(ast2ast)
fetr <- translate(f)
call_fct(fetr)
```

Comparison functions

As usual in R it is possible to compare two objects using one of the following options (**Example 6**):

- ==
- <=
- >=
- !=

- <
- >

Example 6

```
f <- function() {
  a <- 1:4
  b <- c(1, 2, 3, 5)
  c <- 9
  print(a == b)
  print(a <= b)
  print(a >= b)
  print(a != b)
  print(a < c)
  print(a > c)
}

library(ast2ast)
fetr <- translate(f)
call_fct(fetr)
```

Control flow

It is possible to write for-loops and ‘if’, ‘else if’, and ‘else’ branches as in native R (**Example 7**).

```
for(index in variable){
# do whatever
}
for(index in 1:length(variable){
# do whatever
}
```

Example 7

```
f <- function() {
  a <- 1:4
  for(i in a) {
    print(i)
  }

  for(i in 1:length(a)) {
    a[i] <- i + i
  }

  for(i in 1:length(a)) {
    if(i < 4 && i > 1) {
      print(i)
    }
  }

}

library(ast2ast)
fetr <- translate(f)
call_fct(fetr)
```

Printing

Using the function print as common in R (see **Examples 2, 3, 4, 5, 6**):

- print() is different to R
- print(“string”)
- print(logical)
- print(scalar)
- print(vector) is different to R
- print(matrix)

Math functions

Following mathematical functions are available (see **Example 8**):

- sin
- asin
- sinh
- cos
- acos
- cosh
- tan
- atan
- tanh
- log
- ^ and exp

Example 8

```
f <- function() {
  a <- 1:4
  print(sin(a))
  print(cos(a))
  print(a^2)
  print(exp(a, 3))
}

library(ast2ast)
fetr <- translate(f)
call_fct(fetr)
```

Interpolation

To interpolate values, the ‘cmr’ function can be used. The function needs three arguments (see **Example 9**):

- the first argument is the point of the independent variable (x) for which the dependent variable should be calculated (y). This has to be a vector of length one.

- the second argument is a vector defining the points of the independent variable (x). This has to be a vector of at least length four.
- the third argument is a vector defining the points of the dependent variable (y). This has to be a vector of at least length four.

Example 9

```
f <- function() {
  dep <- c(0, 1, 0.5, 2.5, 3.5, 4.5, 4)
  indep <- 1:7

  evalpoints <- c(0.5, 1, 1.5, 2, 2.5,
                  3, 3.5, 4, 4.5, 5,
                  5.5, 6, 6.5)
  for(i in evalpoints) {
    print(cmr(i, indep, dep))
  }
}

library(ast2ast)
fetr <- translate(f)
call_fct(fetr)
```

Information for package authors

Konrad Krämer

- Information for Package authors
- How to use ast2ast
- Rcpp Interface
- RcppArmadillo Interface
- Pointer Interface
- Examples
    - r2sundials
    - paropt

Information for Package authors

This section of the documentation describes how the external pointers produced by *ast2ast* can be used in packages. This information is intended for package authors who want to use *ast2ast* to enable a simple user interface as it is not necessary anymore for the user to write C++. The R code is translated to a modified version of an expression template library called *ETR* (https://github.com/Konrad1991/ETR) which tries to mimic R.

The core class object is named VEC and can be initialized with any type. Furthermore, is a typedef defined for VEC called *sexp* alluding to the fact that all R objects are *SEXP* objects at C level. This class contains another class called STORE which manages memory. To do this a raw pointer of the form *T** is used. Thus, all objects are located at the heap. Beyond that, it is important that no static methods are implemented or that memory is associated with functions or global variables. Therefore, the VEC objects can be used in parallel. However, the user has to take care that only one thread edits an object at a time. VEC a.k.a. *sexp* can be converted to Rcpp::NumericVectors, Rcpp::NumericMatrices, arma::vec and arma::mat. Moreover, it is also possible to copy the data from Rcpp::NumericVectors, Rcpp::NumericMatrices, arma::vec or arma::mat to a *sexp* variable. Currently, the constructors for doing this are not implemented, only the *operator=* is implemented. Thus, the variable of type *sexp* has to be defined before the information of the Rcpp or RcppArmadillo variables are passed. It is possible to construct a *sexp* object by using a *double** pointer and a size information (= int). The information about the Rcpp-, RcppArmadillo- and pointer-interface is explained in depth in the sections below.

After translating the R function each variable is of type *sexp* except if the user uses the *_db* extension. In this case, the variable is of type *double*. All arguments passed to a function are of type *sexp* or *sexp&*. It is not possible to transfer one variable as *sexp* and another one as *sexp&*. The function returns either *void* or a *sexp* object.

How to use ast2ast

In this paragraph, a basic example demonstrates how to write the R code, translate it and call it from C++. Particular emphasis is placed on the C++ code. First of all, the R function is defined which accepts one argument called *a*, adds two to *a* and stores it into *b*. The variable *b* is returned at the end of the function. The R function called *f* is translated to an external pointer to the *C++* function.

```r
f <- function(a) {
  b <- a + 2
  return(b)
}
library(ast2ast)
f_cpp <- translate(f)
```

The C++ function depends on RcppArmadillo and ast2ast therefore the required macros and headers were included. Moreover, *ETR* requires std=c++17 therefore the corresponding plugin is added. The function *getXPtr* is defined by the function *RcppXPtrUtils::cppXPtr*. In the last 5 lines, the translated code is depicted. The function *f* returns a *sexp* and gets one argument of type *sexp* called *a*. The body of the function looks almost identical to the R function. Except that the variable *b* is defined in the first line of the body with the type *sexp*. The function *i2d* converts an integer to a double variable. This is necessary since C++ would identify the *2* as an integer which is not what the user wants in this case.

```cpp
// [[Rcpp::depends(ast2ast, RcppArmadillo)]]
#include <RcppArmadillo.h>
#include <Rcpp.h>
```

```
// [[Rcpp::plugins(cpp17)]]
using namespace Rcpp;

#include <etr.hpp>
// [[Rcpp::export]]
SEXP getXPtr();

sexp f(sexp a) {
  sexp b;
  b = a + i2d(2);
  return(b);
}
```

Afterwards, the translated R function has to be used in C++ code. This would be your package code for example. First, the macros were defined for *RcppArmadillo* and *ast2ast*. Subsequently, the necessary header files were included. As already mentioned *ast2ast* requires std=c++17 thus the required plugin is included. To use the function, it is necessary to dereference the pointer. The result of the dereferenced pointer has to be stored in a function pointer. Later the function pointer can be used to call the translated function. Therefore, a function pointer called *fp* is defined. It is critical that the signature of the function pointer matches the one of the translated function. Perhaps it would be a good idea to check the R function before it is translated. After defining the function pointer, a function is defined which is called later by the user (called *call_package*). This function accepts the external pointer. Within the function body, a variable *f* is defined of type *fp* and *inp* is assigned to it. Next, a *sexp* object called *a* is defined which stores a vector of length 3 containing 1, 2 and 3. The function *coca* is equivalent to the *c* function R. Afterwards *a* is printed. Followed by the call of the function *f* and storing the result in *a*. The variable *a* is printed again to show that the values are changed according to the code defined in the R function.

```
// [[Rcpp::depends(RcppArmadillo, ast2ast)]]
#include "etr.hpp"
// [[Rcpp::plugins("cpp17")]]


typedef sexp (*fp)(sexp a);


// [[Rcpp::export]]
void call_package(Rcpp::XPtr<fp> inp) {
  fp f = *inp;
  sexp a = coca(1, 2, 3);
  print(a);
  a = f(a);
  print("a is now:");
  print(a);
}
```

The user can call now the package code and pass the R function to it. Thus, the user only has to install the compiler or Rtools depending on the operating system. But it is not necessary to write the function in Rcpp.

```
call_package(f_cpp)
```

```
## 1
## 2
## 3
## a is now:
## 3
## 4
## 5
```

Rcpp Interface

In the last section, the usage of *ast2ast* was described. However, only *sexp* variables were defined. Which are most likely not used in your package. Therefore interfaces to common libraries are defined. First of all, *ast2ast* can communicate with Rcpp which alleviates working with the library substantially. The code below shows that it is possible to pass a *sexp* object to a variable of type *NumericVector* or *NumericMatrix* and *vice versa*. Here, the data is always copied.

```
// [[Rcpp::depends(ast2ast, RcppArmadillo)]]
#include <RcppArmadillo.h>
#include <Rcpp.h>
// [[Rcpp::plugins(cpp17)]]
using namespace Rcpp;
#include <etr.hpp>

// [[Rcpp::export]]
void fct() {
  // NumericVector to sexp
  NumericVector a{1, 2};
  sexp a_; // sexp a_ = a; Error!
  a_ = a;
  print(a_);

  // sexp to NumericVector
  sexp b_ = coca(3, 4);
  NumericVector b = b_;
  Rcpp::Rcout << b << std::endl;

  // NumericMatrix to sexp
  NumericMatrix c(3, 3);
  sexp c_; // sexp c_ = c; Error!
  c_ = c;
  print(c_);

  // sexp to NumericMatrix
  sexp d_ = matrix(colon(1, 16), 4, 4);
  NumericMatrix d = d_;
  Rcpp::Rcout << d << std::endl;
}
```

```
trash <- fct()
```

```
## 1
## 2
## 3 4
## 0    0    0
## 0    0    0
## 0    0    0
## 1.00000 5.00000 9.00000 13.0000
## 2.00000 6.00000 10.0000 14.0000
## 3.00000 7.00000 11.0000 15.0000
## 4.00000 8.00000 12.0000 16.0000
```

RcppArmadillo Interface

Besides Rcpp types, *sexp* objects can transfer data to *RcppArmadillo* objects and it is also possible to copy the data from *RcppArmadillo* types to *sexp* objects using the operator =. The code below shows that it is possible to pass a *sexp* object to a variable of type *vec* or *mat* and *vice versa*. Here the data is always copied.

```
// [[Rcpp::depends(ast2ast, RcppArmadillo)]]
#include <RcppArmadillo.h>
#include <Rcpp.h>
// [[Rcpp::plugins(cpp17)]]
using namespace arma;
#include <etr.hpp>

// [[Rcpp::export]]
void fct() {
  // vec to sexp
  arma::vec a(4, fill::value(30.0));
  sexp a_; // sexp a_ = a; Error!
  a_ = a;
  print(a_);

  // sexp to vec
  sexp b_ = coca(3, 4);
  vec b = b_;
  b.print();

  // mat to sexp
  mat c(3, 3, fill::value(31.0));
  sexp c_; // sexp c_ = c; Error!
  c_ = c;
  print(c_);

  // sexp to mat
  sexp d_ = matrix(colon(1, 16), 4, 4);
  mat d = d_;
  d.print();
}
```

```
trash <- fct()
```

```
## 30
## 30
## 30
## 30
##    3.0000
##    4.0000
## 31   31   31
## 31   31   31
## 31   31   31
##      1.0000     5.0000     9.0000    13.0000
##      2.0000     6.0000    10.0000    14.0000
##      3.0000     7.0000    11.0000    15.0000
##      4.0000     8.0000    12.0000    16.0000
```

Pointer Interface

You can pass the information of data stored on heap to a *sexp* object. The constructor for type vector accepts 3 arguments:

- *int* defining the size of the data.
- a pointer ($T*$) to the data
- *int* called cob (copy, ownership, borrow).

The constructor for the type matrix accepts 4 arguments:

- *int* defining number of rows
- *int* defining number of cols
- a pointer ($T*$) to the data
- *int* called cob (copy, ownership, borrow).

If cob is 0 then the data is copied. Else if cob is 1 then the pointer itself is copied. Meaning that the ownership is transferred to the *sexp* object and the user should not call delete [] on the pointer. Be aware that only one *sexp* variable can take ownership of one vector otherwise the memory is double freed. Else if cob is 2 the ownership of the pointer is only borrowed. Meaning that the *sexp* object cannot be resized. The user is responsible for freeing the memory! The code below shows how the pointer interface works in general. Showing how *sexp* objects can be created by passing the information of pointers ($double*$) which hold data on the heap. Currently, only constructors are written which can use the pointer interface.

```
// [[Rcpp::depends(ast2ast, RcppArmadillo)]]
#include <RcppArmadillo.h>
#include <Rcpp.h>
// [[Rcpp::plugins(cpp17)]]
using namespace arma;
#include <etr.hpp>

// [[Rcpp::export]]
void fct() {
    int size = 3;

    // copy
    double* ptr1;
    ptr1 = new double[size];
    int cob = 0;
    sexp a(size, ptr1, cob);
    delete [] ptr1;
    a = vector(3.14, 5);
    print(a);

    print();

    // take ownership
    double* ptr2;
    ptr2 = new double[size];
    cob = 1;
    sexp b(size, ptr2, cob);
    b = vector(5, 3);
    print(b);

    print();

    // borrow ownership
    double* ptr3;
    ptr3 = new double[size];
    cob = 2;
    sexp c(size, ptr3, cob);
    //error calls resize
    //c = vector(5, size + 1);
    c = vector(4, size);
    print(c);

    print();
    sexp d(size, ptr3, cob);
    d = d + 10;
    print(d);
    print();

    delete[] ptr3;
}
```

```
trash <- fct()
```

```
## 3.14
## 3.14
## 3.14
## 3.14
## 3.14
##
## 5
## 5
## 5
##
## 4
## 4
## 4
##
## 14
## 14
## 14
```

The pointer interface is particularly useful if the user function has to change the data of a vector or matrix of type *NumericVector*, *vec*, *NumericMatrix* or *mat*. Assuming that the user passes a function that accepts its arguments by reference it is easy to modify any variable which has a type that can return a pointer to its data. In the code below it is shown how *sexp* objects are constructed using the pointer interface. Thereby changing the content of variables which has an Rcpp type, a RcppArmadillo type, or is of type std::vector.

```
// [[Rcpp::depends(ast2ast, RcppArmadillo)]]
#include <RcppArmadillo.h>
#include <Rcpp.h>
// [[Rcpp::plugins(cpp17)]]
using namespace Rcpp;
using namespace arma;
#include <etr.hpp>

typedef void (*fp)(sexp& a);

// [[Rcpp::export]]
void call_package(Rcpp::XPtr<fp> inp) {
  fp f = *inp;

  // NumericVector
  NumericVector a_rcpp{1, 2, 3};
  sexp a(a_rcpp.size(), a_rcpp.begin(), 2);
  f(a);
  Rcpp::Rcout << a_rcpp << std::endl;

  //arma::vec
  vec a_arma(2, fill::value(30));
  sexp b(2, a_arma.memptr(), 2);
  f(b);
  a_arma.print();

  // NumericMatrix
  NumericMatrix c_rcpp(2, 2);
  sexp c(2, 2, c_rcpp.begin(), 2);
  f(c);
  Rcpp::Rcout << c_rcpp << std::endl;

  //arma::mat
  mat d_arma(3, 2, fill::value(30));
  sexp d(3, 2, d_arma.memptr(), 2);
  f(d);
  d_arma.print();
}
```

```
f <- function(a) {
  a <- a + 2
}

library(ast2ast)
fa2a <- translate(f, reference = TRUE)
trash <- call_package(fa2a)
```

```
## 3 4 5
##    32.0000
##    32.0000
## 2.00000 2.00000
## 2.00000 2.00000
##
##    32.0000   32.0000
##    32.0000   32.0000
##    32.0000   32.0000
```

Examples

In this section two examples are shown to illustrate how *ast2ast* could be used in R packages. First, it is shown how ODE-systems can be solved very efficiently. In order to do this the R package *r2sundials* is used. The second examples point out how *ast2ast* can be used together with the R package *paropt* to optimize parameters of ODE-systems. Both examples only construct wrapper functions that are used by the package code. Probably it would be more efficient when the package code itself would take care of the transfer of data between *Rcpp/RcppArmadillo* and *ast2ast*.

r2sundials

In this example it is shown how ODE-systems can be solved by using *r2sundials* and *ast2ast*. The code below shows how *r2sundials* is used normally. Either using an R function or an external pointer to a C++ function.

```
library(Rcpp)
library(ast2ast)
library(r2sundials)
```

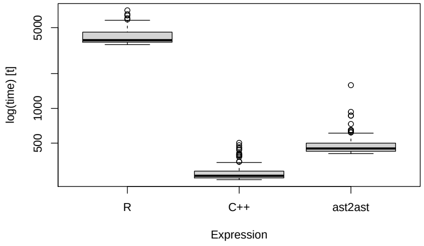
```
## Loading required package: rmumps
```

```
library(RcppXPtrUtils)
library(microbenchmark)

# R version
ti <- seq(0, 5, length.out=101)
p <- list(a = 2)
p <- c(nu = 2, a = 1)
y0 <- 0
frhs <- function(t, y, p, psens) {
  -p["nu"]*(y-p["a"])
}

res_exp <- r2cvodes(y0, ti,
                    frhs, param = p)
attributes(res_exp) <- NULL

# External pointer
ptr_exp <- cppXPtr(code = '
int rhs_exp(double t, const vec &y,
            vec &ydot,
            RObject &param,
            NumericVector &psens) {

  double a = 1;
  double nu = 2;
  ydot[0] = -nu*(y[0] - a);
  return(CV_SUCCESS);
}
',
depends=c("RcppArmadillo",
          "r2sundials","rmumps"),
includes="using namespace arma;\n
          #include <r2sundials.h>",
cacheDir="lib", verbose=FALSE)

pv <- c(a = 1)
res_exp2 <- r2cvodes(y0, ti,
                     ptr_exp, param = pv)
attributes(res_exp2) <- NULL
```

In the code below is the wrapper function defined which is later called by *r2sundials*. This function is called *rhs_exp_wrapper* and has the correct function signature. Furthermore, a global function pointer named *Fct* is defined of type *void (**user_fct) (sexp& y__, sexp& ydot__). *Within* rhs_exp_wrapper* the data of the vectors *y* and *ydot* are used to construct two *sexp* objects which are passed to *Fct*. Thus, the vector *ydot* is modified by the function passed from the user. Furthermore, another function called *solve_ode* is defined. Which calls the code from *r2sundials*, solves the ODE-system, and returns the output to R. In R the user defines the R function *ode*. Next, the function is translated and passed to *solve_ode*. Comparing the results shows that all three approaches (R, C++, *ast2ast*) generate the same result. Afterwards a benchmark is conducted showing that R is substantially slower than C++ and that the translated function is almost as fast as C++. Mentionable, it is possible to increase the speed of the *ast2ast* version by using the *at* function and the *_db* extension.

```
// [[Rcpp::depends(RcppArmadillo)]]
// [[Rcpp::depends(rmumps)]]
// [[Rcpp::depends(r2sundials)]]
// [[Rcpp::depends(ast2ast)]]
// [[Rcpp::plugins("cpp17")]]
#include "etr.hpp"
#include "RcppArmadillo.h"
#include "r2sundials.h"
using namespace arma;
using namespace Rcpp;

typedef int (*fp)(double t, const vec &y,
             vec &ydot, RObject &param,
             NumericVector &psens);

typedef void (*user_fct)(sexp& y_,
              sexp& ydot_);
user_fct Fct;

int rhs_exp_wrapper(double t, const vec &y,
                    vec &ydot, RObject &param
                    NumericVector &psens) {
  NumericVector p(param);
  const int size = y.size();
  sexp ydot_(size, ydot.memptr(), 2);

  double* ptr = const_cast<double*>(
              y.memptr());
  sexp y_(size, ptr, 2);
  Fct(y_, ydot_);
  return(CV_SUCCESS);
}


// [[Rcpp::export]]
NumericVector solve_ode(XPtr<user_fct> inp,
                        NumericVector time,
                        NumericVector y) {
Fct = *inp;
XPtr<fp> ptr = XPtr<fp>(new fp(
            &rhs_exp_wrapper));

Environment pkg =
  Environment::namespace_env("r2sundials");
Function solve = pkg["r2cvodes"];
NumericVector output = solve(y, time,
                             ptr, time);

return output;
}
```

```
# ast2ast version
ti <- seq(0, 5, length.out=101)
y0 <- 0

library(ast2ast)
ode <- function(y, ydot) {
  nu <- 2
  a <- 1
  ydot[1] <- -nu*(y[1] - a)
}
pointer_to_ode <- translate(ode,
                            reference = TRUE)
res_exp3 <- solve_ode(pointer_to_ode,
                      ti, y0)
attributes(res_exp3) <- NULL

stopifnot(identical(res_exp,
                    res_exp2,
                    res_exp3))
out <- microbenchmark(
  r2cvodes(y0, ti,
           frhs, param = p),
  r2cvodes(y0, ti,
           ptr_exp, param = pv),
  solve_ode(pointer_to_ode,
            ti, y0))

boxplot(out, names=c("R", "C++", "ast2ast"))
```



paropt

In the second example, it is shown how *ast2ast* is used together with *paropt*. Again the code below consists of an Rcpp part and a part written in R. Within Rcpp the C++ code is defined which describes the Lotka-Volterra model (*ode_system*). For details check the documentation of *paropt*. Subsequently, a function is defined that returns an external pointer to the ODE-system (*test_optimization*). As already described in the last example a wrapper function is created which calls the actual ODE-system (*ode_system_wrapper*). Furthermore, a function is defined which calls the code from the package *paropt* called *optimize_paropt* and conducts the optimization. Within R the ODE-system is defined, translated and used for optimization. Both versions the pure C++ function and the translated R code yield the same result. The C++ function is almost as fast as the translated R code. Interesting is the fact that *paropt* calls the ODE-solver in parallel. This example demonstrates that the translated function can be called in parallel. Which is not possible with R functions. Admittedly, the printing is conducted by Rcpp. Therefore it is unknown how printing behaves in parallel.

```
// [[Rcpp::depends(RcppArmadillo)]]
// [[Rcpp::depends(ast2ast)]]
// [[Rcpp::plugins("cpp17")]]
// [[Rcpp::depends(paropt)]]
#include "etr.hpp"
#include "RcppArmadillo.h"
using namespace Rcpp;

// paropt version
typedef int (*OS)(double &t,
             std::vector<double> &params,
             std::vector<double> &states);

int ode_system(double &t,
               std::vector<double> &params,
               std::vector<double> & states)

  double a = params[0];
  double b = params[1];
  double c = params[2];
  double d = params[3];

  double n1 = states[0];
  double n2 = states[1];

  states[0] = n1*c*n2 - n1*d;
  states[1] = n2*a - n2*b*n1;
  return 0;
}

// [[Rcpp::export]]
XPtr<OS> test_optimization() {
  XPtr<OS> xpfun = XPtr<OS>(new OS(
                     &ode_system));
  return xpfun;
}

// ast2ast version
typedef int (*paropt_fct)(double &t,
             std::vector<double> &params,
             std::vector<double> & states);

typedef void (*user_fct2)(sexp& p,
                          sexp& y);

user_fct2 Fct_paropt;

int ode_system_wrapper(
    double &t,
    std::vector<double> &params,
    std::vector<double> & states) {

  sexp p(params.size(), params.data(), 2);
  sexp s(states.size(), states.data(), 2);
  Fct_paropt(p, s);
  return 0;
}


// [[Rcpp::export]]
List optimize_paropt(XPtr<user_fct2> inp,
                     NumericVector time,
                     DataFrame lb,
                     DataFrame ub,
                     DataFrame states) {
  Fct_paropt = *inp;
  XPtr<paropt_fct> ptr = XPtr<paropt_fct>(
    new paropt_fct(&ode_system_wrapper));

  Environment pkg =
    Environment::namespace_env("paropt");
  Function optim =
    pkg["optimizer_pointer"];

  NumericVector abs_tols{1e-8, 1e-8};

  List output = optim(time, ptr, 1e-6,
                      abs_tols, lb, ub,
                      states, 40,
                      1000, 0.0001,
                      "bdf");

  return output;
}
```

```
#states
path <- system.file("examples",
                    package = "paropt")
states <- read.table(paste(
                      path,
                      "/states_LV.txt",
                      sep = ""),
                     header = T)

# parameter
lb <- data.frame(time = 0,
                 a = 0.8,
                 b = 0.3,
                 c = 0.09,
                 d = 0.09)
ub <- data.frame(time = 0,
                 a = 1.3,
                 b = 0.7,
                 c = 0.4,
                 d = 0.7)

suppressMessages(library(paropt))
set.seed(1)
start_time <- Sys.time()
df_cpp <- optimizer_pointer(
  integration_times = states$time,
  ode_sys = test_optimization(),
  relative_tolerance = 1e-6,
  absolute_tolerances = c(1e-8, 1e-8),
  lower = lb, upper = ub, states = states,
  npop = 40, ngen = 1000, error = 0.0001,
  solvertype = "bdf")
end_time <- Sys.time()
cpp_time <- end_time - start_time

# ast2ast with at and _db
ode <- function(params, states) {
  a_db = at(params, 1)
  b_db = at(params, 2)
  c_db = at(params, 3)
  d_db = at(params, 4)
  n1_db = at(states, 1)
  n2_db = at(states, 2)
  at(states, 1) = n1_db*c_db*n2_db -
    n1_db*d_db;
  at(states, 2) = n2_db*a_db -
    n2_db*b_db*n1_db;
}

pointer_to_ode <- ast2ast::translate(
                        ode,
                        reference = TRUE)
set.seed(1)
start_time <- Sys.time()
df_ast2ast <- optimize_paropt(pointer_to_ode,
                              states$time,
                              lb, ub, states)
end_time <- Sys.time()
a2a_time <- end_time - start_time

stopifnot(identical(df_cpp[[8]],
                    df_ast2ast[[8]]) )

times <- data.frame(cpp = cpp_time,
                    ast2ast = a2a_time)
kableExtra::kbl(times)
```

cpp	ast2ast
27.93203 secs	35.86573 secs

10 Appendix A

A.1 Communication of R with C and C++

R is a programming language which has a large set of packages useful for statistical modeling, visualization and data import and manipulation (Wickham, 2015). The R language is a successor of the programming language S, which was created in the 1980s and was used for statistical analysis (R Core Team, 2017). As in any programming language, R uses variables to store data. Notably, R does not provide direct access to the memory of the computer. Rather it uses data structures called objects. Mentionable, the user does not have to care about memory management as it is the case in the programming language C. More precisely the memory of an object which is not needed anymore is deleted by the *Garbage collector* (GC).

As R was written in C, all objects are represented at the C level as pointers to a structure called SEXPREC. (R Core Team, 2017). These pointers are called S-expressions (SEXP). SEXPs can hold every R object, because it comprises subtypes for every R data structure. For instance, the types REALSXP, INTSXP or VECSXP hold at R level a numeric vector, an integer vector and a list, respectively (Wickham, 2015). Therefore, a function accepts and returns only SEXPs at the C-level. Beyond that it is necessary within a C function exposed to R to protect all SEXP objects from the GC. Otherwise the GC assumes that they are not needed anymore and tries to delete them which results in unexpected behaviour. In order to protect a SEXP object one has to put the object on the so called protect stack. Afterwards, when the object is not needed anymore it has to be removed from the stack, otherwise the memory will not be deleted. The process of writing C functions for R is simplified using the package *tidyCpp (Dirk Eddelbuettel, 2021)* as it offers a clean and simple C++ layer over R’s complicated C API. In addition using *tidyCpp* does not require to put SEXP objects on the protect stack in order to prevent the garbage collector to delete them. It is also not necessary to remove objects from the protect stack at the end of functions.

For instance, the calculation of the sum of a numeric vector (Fig. AI A) requires the extraction of the length of the vector. Afterwards a new SEXP object is created which contains the result and is returned at the end. Subsequently two double pointers have to be created and directed to the memory location of the input vector and the output vector. Using these variables the actual operation is conducted in a for loop

(Fig. AI A). The last step is the return of the SEXP object output. Even if using *tidyCpp* R’s C interface is rather complicated and affords a deep knowledge of R and C. An alternative is the use of the R package *Rcpp (D Eddelbuettel & Francois, 2011; Dirk Eddelbuettel & Balamuta, 2018)*. This package conducts the translation of R objects into C++ data structures and *vice versa*. Thus, the software developer can concentrate on the actual code. For example, in Fig. AI B the code for calculating a sum of a numeric vector is more compact and does not contain the extraction steps or the need to use raw pointers. Thus, the programs are less error prone.

A

```
#include <tidyCpp>
// [[Rcpp::depends(tidyCpp)]]

// [[Rcpp::export]]
SEXP sum(SEXP x)
{
  int nx = R::length(x);

  R::Protect output(R::allocVectorNumeric(1) );
  out[0] = 0.0;
  double* rx = R::numericPointer(x);
  double* out = R::numericPointer((output));


  for(int i = 0; i < nx; i++) {
    out[0] += rx[i];
  }
  return output;
}
```

B

```
#include <Rcpp.h>
using namespace Rcpp;

// [[Rcpp::export]]
double sum(NumericVector x) {
  double sum = 0.0;
  for(auto& i : x) {
    sum += i;
  }
  return sum;
}
```

Figure AI: Calculating sum of a vector. A: Using R’s C interface. B: Using the R package Rcpp.

paropt

Internal structure of the R package *paropt*

A.2 paropt

The R package *paropt* was developed for easy and fast optimization of parameters of ODE-systems (Krämer *et al.*, 2021). The program is written in C++, which is one of the programming languages yield the fastest executing code, and also has a simple interface to R objects (Dirk Eddelbuettel & Balamuta, 2018). Several objects must be created to conduct simulations. A function describing the ODE-system, information about the states, and boundaries of the parameter space. The ODE-system can either be a normal Rcpp-function (Fig. AII A) or an external pointer to a C++ function (Fig. II B).

An external pointer to a C++ function enables storing a C++ function in an R object of the type *external pointer*. This object can be passed from R to a C++ program, which can extract the pointer and use the C++ function. In contrast, the Rcpp-function always has a thin R wrapper around the C++ function in use.

In principal, both functions are structured in the same way. However, the Rcpp-function returns the result of the right hand side (rhs) in a new vector, while in case of using an external pointer, the result is stored in the vector previously containing the state information at timepoint t.

The advantage of using the external-pointer interface is that the thin R-wrapper around the Rcpp-function is removed. Furthermore, it is possible to call the extracted external pointer C++ function in parallel as no R objects are used during the function call. This allows to solve the ODE-system with different parameter sets on several CPU cores simultaneously. However, the caveat is that the user cannot use any R objects (e.g. Rcpp::NumericVector) or any R functions within the ODE-system. In addition, the user must make sure that the function can be called in parallel.

A

```
#include <Rcpp.h>

 // [[Rcpp::export]]
 Rcpp::NumericVector ode_system(
                              double t,
                              std::vector<double>
params,
                              Rcpp::NumericVector
states) {

  Rcpp::NumericVector states_deriv(2);

   double a = params[0];
   double b = params[1];
   double c = params[2];
   double d = params[3];

   double predator = states[0];
   double prey = states[1];

   double ddtpredator = states_deriv[0] =
        predator*c*prey - predator*d;
   double ddtprey = states_deriv[1] = prey*a
-
        prey*b*predator;

   return states_deriv;
 }
```

B

```
// [[Rcpp::depends(RcppArmadillo)]]
 #include <RcppArmadillo.h>
 // [[Rcpp::depends(paropt)]]
 // [[Rcpp::plugins(cpp11)]]

typedef int (*OS)(double &t, std::vector<double>
&params, std::vector<double> &states);

 int ode_system(double &t,
                         std::vector<double>  &params,
                         std::vector<double> & states) {

  double a = params[0];
  double b = params[1];
  double c = params[2];
  double d = params[3];

   double predator = states[0];
   double prey = states[1];

      double ddtpredator = states[0] =
         predator*c*prey - predator*d;
    double ddtprey = states[1] = prey*a -
         prey*b*predator;

   return 0;
 }


 // [[Rcpp::export]]
 Rcpp::XPtr<OS> test_optimization() {
   Rcpp::XPtr<OS> xpfun = Rcpp::XPtr<OS>(new
OS(&ode_system));

   return xpfun;}
```

Figure AII: Ode-system describing the Lotka-Volterra model as used within the package paropt. A: Using a normal Rcpp function. B: Using an external pointer to a C++ function.

Besides the ODE-system the user has to pass the state information as a data.frame called *dfStates*. The *dfStates* object is linearized, i.e. all entries appearing in *dfStates* are put in one consecutive vector (*stateVector)*. This is conducted by concatenating the columns of *dfStates* together.

An additional vector called *cutVector* is used in order to safe the length of the respective columns. In addition, a vector called *timeVector* is created which stores the independent variable entries of the *stateVector* (most often it is the time). In conclusion, the object *dfStates* is now represented by three vectors. This allows a more efficient memory access. The three vectors are needed for the calculation of the error term for a chosen parameter set. The error term is given as the difference between the *in silico* state trajectories and the measured state courses, which are stored in the three vectors *stateVector*, *timeVector* and *cutVector*. Notably, an extra vector is defined which contains only the intial conditions at the starting point.

In addition, *paropt* needs the lower and upper boundaries of the parameters that shall be optimized. Both boundary sets are passed as data frames and are also linearized as described for the state information. Using the lower and upper boundaries, henceforth called *lb* and *ub*, the parameters are optimized using a particle swarm optimization algorithm.

Particle swarm optimization

The particle swarm optimization (PSO) algorithm is a tool for global optimization which was first described in 1995 (Eberhart & Kennedy, 1995). The algorithm is inspired by the behavior of bird flocks looking for corn (Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995). For describing how the PSO works several terms have to be defined. The so called *search space* is defined in the case of *paropt* by *lb* and *ub*. In case only one parameter (one dimensional: 1-D) had to be optimized the *search space* would be a line in case of two parameters (two dimensional: 2-D) an area would be defined and so forth. At each point of the *search space* the function, for which the parameters should be optimized, can be evaluated and returns a scalar numerical value, the so called *fitness*. The point where the maximum or minimum of the *fitness* value is located within *search space* is the global optimum (Clerc, 2012).

Within the *search space* exists a swarm of particles. Each particle consists of a current position within the *search space* (a N dimensional (= N-D) vector) and the corresponding *fitness* value of the current position (1-D). Moreover, the particle contains a *parameter set* describing the personal best position (a N-D vector) and the corresponding *fitness value* (1D). Besides, each particle contains a velocity (a N-D vector) which is used to update the current position of the particle (Clerc, 2012; Sengupta *et al.*, 2018). The swarm itself contains a vector (a N-D vector) which defines the up to now best parameter set and a scalar value bearing the *fitness* for these parameters.

Before operation, the PSO must be initialized by attributing each particle in the *search space* a position at random. For each particle the *fitness* is calculated and a random velocity is ascribed. Subsequently, the local best parameters and *fitness* values are stored. Next, the global best particle is identified and the corresponding parameters and *fitness* are stored in the *memory* of the swam (Zambrano-Bigiarini *et al.*, 2013).

After initialization the swarm explores the parameter space for a certain amount of times. Each iteration is called a *generation*. During each iteration the following steps are conducted. A new velocity for each particle is calculated and its current position is updated based on the new velocity. Using the new position the *fitness* is calculated, this is done for all particles of the swarm. Afterwards, it is checked for each particle whether a new personal best position is found. Next, it is checked whether one of the personal best positions represent a global best position.

Velocity calculation

In the canonical PSO the velocity of particle i is calculated based on the following equation (Sengupta *et al.*, 2018):

Vi_new = Vi + 2.0*rand*(personal_best – current_position) + 2.0*rand*(global_best – current_position)

Where:

Vi_new: new velocity of particle i
Vi: Velocity of particle i
rand: random scalar
personal_best: parameter defining personal best *fitness* for particle i
current_position: current parameters in *search space* of particle i
global_best: parameters of the best fitness of the entire swarm

The mode of updating the velocity has been improved repeatedly. Zambrano-Bigiarini, Clerc and Rojas, 2013, represented the following form of updating the velocity of particle i:

Vi_new = ω*Vi + cog*rand*(personal_best – current_position) + soc*rand*(global_best – current_position)

Where:

Vi_new: new velocity of particle i
Vi: Velocity of particle I
ω: factor ω
rand: random scalar
cog: cognitive factor
soc: social factor
personal_best: parameter defining personal best *fitness* for particle i
current_position: current parameters in *search space* of particle i
global_best: parameters of the best fitness of the entire swarm

In its improved version, *swarm explosion,* which is the uncontrolled increase of particle velocities is prevented by the ω factor which constraints the own velocity (Zambrano-Bigiarini *et al.*, 2013). The ω factor in the PSO used by *paropt* starts at 0.9 and linearly decrease to 0.4 during the simulations. Thus, at the beginning each particle explore the *search space* whereas at the end the communication between the particles becomes more important.

The cognitive factor regulates the contribution of the personal best parameters for velocity calculation. A high proportion of this factor would thus lead to conducting a local search (Kennedy, 1997). This parameter is set to 2.5 as a start value and linearly decreases during each generation to a final value of 0.5 (Sengupta *et al.*, 2018). The social factor describes the orientation of a single particle i towards the global best particle of the swarm. This parameter is linearly increased from 0.5 to 2.5 during the optimization (Sengupta *et al.*, 2018).

Topologies used in PSO

Despite improvement, the above presented approach suffers from premature convergence. Thus, the swarm may only find local minima (or maxima) and become trapped without being able to further improve the *fitness* value (Zambrano-Bigiarini *et al.*, 2013). To address this issue, different topologies were used in case of *paropt* a so called randomly adapted topology was implemented (Gong & Zhang, 2013). Akman, Akman and Schaefer, 2018, showed that a random adaptive topology is best suited for the optimization of parameters of ODE-systems. This approach is also used in *paropt (Krämer et al., 2021)*. It implies that the current position of a particle is not compared with the global best solution, which would result in a star topology. Instead, for each particle a random neighborhood is calculated. Each particle has a certain amount of other particles called neighbors. These are usually between zero and three. For each neighborhood the best particle is identified and used instead of the global best solution. Neighborhoods are used until the global best *fitness* value does not improve for a previously defined number of generations (Akman *et al.*, 2018; Sengupta *et al.*, 2018).

Calculating the fitness value

As described above, for each generation the *fitness* value is calculated for each particle. Thus, the ODE-system has to be solved using the initial conditions of the states and a parameter set given by the current position of a particle. For solving the ODE-system *paropt* uses the SUNDIALS collection of nonlinear and differential/algebraic equation solvers (Hindmarsh *et al.*, 2005), that very efficiently yield high quality results (Rackauckas, 2018). Parameters being variable within the simulation time were interpolated using a Catmull-Rom-Spline (Twigg, 2003). Comparing the *in silico* solution of the ODE-solver to the real set of the states given in the *stateVector* results in the *fitness* value used by the PSO algorithm. Different error calculations were tested with the sum of the absolute differences and a relative error (percentage error) were most efficient for optimization.

A.3 ast2ast

During parameter optimization using the R package *paropt (Krämer et al., 2021)* the ODE-system under consideration is solved very often. Thus, the function describing the system should allow fast calculation. To achieve this, the ODE-system has to be coded in a fast programming language such as Fortran, C, C++ or Julia. The R package *paropt* has an interface for functions written in C++. However, the caveats of using a fast language are the increased complexity during programming and the lack of safeguards such as access to computer memory, which lies outside the boundaries of a vector. In addition, a weakness of *paropt* lies in the fact that both, the parameter and state arguments, are vectors. In case of large parameter or state spaces this bears the risk of index errors. However, these errors cannot be handled by *paropt*, because the user transfers an already compiled function.

To offer an application interface which allows writing the ODE-system in an interpreted language and prevent index errors, the package *ast2ast* was developed (Krämer K. 2022c). The idea behind is that the user writes the function in the R language, which is then translated into C++. Subsequently the compiled C++ function is transferred to *paropt*. The program consist of two parts, one written in R and the other in C++. The C++ part is an expression template library, called *ETR* (Expression template library R), which mimics the behavior of R (https://github.com/Konrad1991/ETR), whereas the R part is responsible for transfer from R to *ETR*.

Metaprogramming in R

Metaprogramming is a technique to inspect and manipulate code. Thus, code is used as a kind of data. This enables many features specific for R, e.g., expressions like: ‘y ~ a*b*c’ can be used (Wickham, 2015). Furthermore, the code in R is internally represented as an abstract syntax tree (AST). Within the AST, function calls are the branches and constants or symbols represent the leaves (Fig. AIII) (Wickham, 2015). It is possible to inspect and manipulate the AST of R code. For example in C++ it is not possible to use ‘<-’ to assign a value to a variable. Thus, the assignment operator ‘<-’ has to be replaced by ‘=’ in order to translate R to C++/ETR.

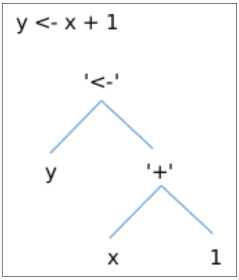


Figure AIII: Abstract syntax tree representation of R code.

ETR

Expression templates are a technique to efficiently implement basic arithmetic operations (+, -, *, /) and other functions for vectors, matrices or other objects. There exists numerous expression template libraries like, e.g., *armadillo (Sanderson & Curtin, 2018)*. C++ offers the possibility to write expression template libraries as it includes several features: object orientated programming, templates for function and classes and the ability to overload operators (Härdtlein *et al.*, 2010).

Object orientated programming

Object orientated programming is the use of types called classes, which encapsulate attributes and methods (Loudon & Grimm, 2018). The code of the classes are only blueprints, which means that only after the instantiation memory is allocated for the class (see Fig. AIV: Rectangle r(3, 4) as an example).

```
class Rectangle {
        public:

        int length;
        int width;

        Rectangle(int l, int w) : length(l), width(w) {}

        int square() {
              return length * width;
        }

};

int main() {

        Rectangle r(3, 4);

        std::cout << r.square() << std::endl;

}
```

Figure AIV: Example for a class Rectangle and how to instantiate the class and call a method.

Templates

A template allows implementation of a function or class only once and call it with different types. The different types are constructed by the compiler (Loudon & Grimm, 2018). For instance, Fig. AV shows a function called *printer* which accepts a template argument as parameter. In the main function the user indicates the type in <> brackets or the compiler deduces the type of the function (as for the type bool).

```
template<typename T>
void printer(T inp) {
	std::cout << inp << std::endl;
}

int main() {

	int i = 1;
	double d = 5.3;
	bool b = true;

	printer<int>(i);
	printer<double>(d);
	printer(b);

}
```

Figure AV: Example for a function template and how to call it.

Operator overloading

The last technique required to implement expression template libraries is the ability to overload operators i.e. defining the behavior of operators e.g. '+' for a certain type (Härdtlein *et al.*, 2010; Loudon & Grimm, 2018). For instance one can overload the '+' operator for vectors. Meaning that a function is defined which is called when '+' is used for vectors. Thus, it is possible to add the entries of two vectors using only the operator '+' and not a for loop.

In Fig. AVI a class VEC is defined which has a std::vector as an attribute. This std::vector contains a template type T which depends on the instantiation of the class VEC. In the main function the class VEC is always called with the type double. The example in Fig. AVI shows three different operators which are overloaded. The first operator is the '=' operator. The function defined for operator '=' now accepts another

instance of class VEC. Within the body of the ‘=’ function the data stored in the other vector is stored within the std::vector data. Notably, the function ‘=’ shows this behavior only for objects of type VEC. In case one would add two objects of type double the ‘normal’ ‘=’ function would be called.
Next the operator[ ] is overloaded. This enables access to a specific member of the std::vector data. The last operator which is overloaded is the ‘+’ operator. The function ‘+’ can now be called with two vectors. Within the function-body a loop iterates over both vectors whereby adding the entries at index i and storing it in a new vector called res. In the main function three vectors are defined v1, v2 and v3. Afterwards, v1 and v2 are added and stored in v3.

```
template<typename T>
class VEC {
 public:
  std::vector<T> data;

  VEC(int size, double value) : data(size, value) {}

  VEC& operator=(const VEC<T>& inp) {
   for(int i = 0; i< inp.data.size(); i++) {
    data[i] = inp.data[i];
   }

   return *this;
  }

  double& operator[](int i) {
   return data[i];
  }

};


VEC<double> operator+(VEC<double>& left, VEC<double>& right) {
 VEC<double> res(left.data.size(), 0);
 for(int i = 0; i< res.data.size(); i++) {
  res[i] = left[i] + right[i];
 }

 return res;
}

int main() {

 VEC<double> v1(2, 2.5);
 VEC<double> v2(2, 10.2);
 VEC<double> v3(2, 0.0);

 v3 = v1 + v2;
 std::cout << v3[0] << " " << v3[1] << std::endl;

}
```

Figure AVI: Example for operator overloading in classes.

Expression template libraries

One could argue that it is only necessary to overload the operators in order to create a nice interface for numerical operations. For instance, defining how two vectors are added one would iterate over the two vectors, adding the scalar elements and store it in a new vector. In case one would add two vectors the approach would be efficient. However, as C++ is eagerly evaluated the approach would be very inefficient for longer expressions. For example if three vectors are added the first step would be to add the first two vectors. Next, the resulting vector would be stored on the heap and added to the third one. Thus, the approach needs a lot of memory and it is necessary to iterate two instead of one time over the length of the vectors.

To avoid these problems the data involved in operations is stored in special classes. For example the following expression should be calculated: a * b + c - d. Where a, b, c and d are all vectors. To do this three classes were instantiated. The first one is from type *multiply and* hereafter called *MUL* and contains the data from a and b. The second class is from type *subtract and* called *SUB* in the following. SUB contains the data from c and d. The third class is of type *add* and is called *ADD.* The instance *ADD* contains *MUL* and *SUB* as data. For each of the classes *add*, *subtract* and *multiply* the operator [] is overloaded. If the operator [] is called at position x the operation is conducted. In the case of *add* the vector entries at position x are added in case of *subtract* the entries are subtracted and in the case of *multiply* the entries are multiplied. Thus, an expression tree is constructed (Fig. AVII) which has similarities with the AST. If the object *ADD* is passed to another vector (Fig. AVI) the operator [] is called which in turn calls the *, - and + operation for each entries in the vectors. Thus, the loop is only conducted once using the expression templates approach.

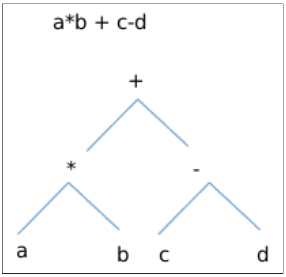


*Figure AVII: Expression tree of the operation a*b + c-d*

11 References

Ainsworth, E., & Rogers, A. (2007). The response of photosynthesis and stomatal conductance to rising CO_2: Mechanisms and environmental interactions. *Plant, Cell and Environment*, *30*(3), 258–270. https://doi.org/10.1111/j.1365-3040.2007.01641.x

Akman, D., Akman, O., & Schaefer, E. (2018). Parameter Estimation in Ordinary Differential Equations Modeling via Particle Swarm Optimization. *Journal of Applied Mathematics*, *2018*. https://doi.org/10.1155/2018/9160793

Alseekh, S., Aharoni, A., Brotman, Y., Contrepois, K., D'Auria, J., Ewald, J., C. Ewald, J., Fraser, P. D., Giavalisco, P., Hall, R. D., Heinemann, M., Link, H., Luo, J., Neumann, S., Nielsen, J., Perez de Souza, L., Saito, K., Sauer, U., Schroeder, F. C., Fernie, A. R. (2021). Mass spectrometry-based metabolomics: a guide for annotation, quantification and best reporting practices. *Nature Methods*, *18*(7), 747–756. https://doi.org/10.1038/s41592-021-01197-1

Andrews, M., Condron, L. M., Kemp, P. D., Topping, J. F., Lindsey, K., Hodge, S., & Raven, J. A. (2019). Elevated CO_2 effects on nitrogen assimilation and growth of C3 vascular plants are similar regardless of N-form assimilated. *Journal of Experimental Botany*, *70*(2), 683–690. https://doi.org/10.1093/jxb/ery371

Andrews, M., Condron, L. M., Kemp, P. D., Topping, J. F., Lindsey, K., Hodge, S., & Raven, J. A. (2020). Will rising atmospheric CO_2 concentration inhibit nitrate assimilation in shoots but enhance it in roots of C3 plants? *Physiologia Plantarum*, *170*(1), 40–45. https://doi.org/10.1111/ppl.13096

Asensio, J. S. R., Rachmilevitch, S., & Bloom, A. J. (2015). Responses of arabidopsis and wheat to rising CO_2 depend on nitrogen source and nighttime CO2 levels. *Plant Physiology*, *168*(1), 156–163. https://doi.org/10.1104/pp.15.00110

Bauwe, H., Hagemann, M., & Fernie, A. R. (2010). Photorespiration: players, partners and origin. *Trends in Plant Science*, *15*(6), 330–336. https://doi.org/10.1016/j.tplants.2010.03.006

Berteli, F., Corrales, E., Guerrero, C., Ariza, M. J., Pliego, F., & Valpuesta, V. (1995). Salt stress increases ferredoxin-dependent glutamate synthase activity and protein level in the leaves of tomato. *Physiologia Plantarum*, *93*(2), 259–264. https://doi.org/10.1111/j.1399-3054.1995.tb02226.x

Bittsánszky, A., Pilinszky, K., Gyulai, G., & Komives, T. (2015). Overcoming ammonium toxicity. *Plant Science*, *231*, 184–190. https://doi.org/10.1016/j.plantsci.2014.12.005

Bloom, A. J. (2015). Photorespiration and nitrate assimilation: A major intersection between plant carbon and nitrogen. *Photosynthesis Research*, *123*(2), 117–128. https://doi.org/10.1007/s11120-014-0056-y

Bloom, A. J., Burger, M., Kimball, B. A., & Pinter, P. J. (2014). Nitrate assimilation is inhibited by elevated CO_2 in field-grown wheat. *Nature Climate Change*, *4*(6), 477–480. https://doi.org/10.1038/nclimate2183

Bloom, A. J., Burger, M., Rubio Asensio, J. S., & Cousins, A. B. (2010). Carbon Dioxide Enrichment Inhibits Nitrate Assimilation in Wheat and Arabidopsis. *Science*, *328*(May), 899–904.

Bloom, A. J., Kasemsap, P., & Rubio-Asensio, J. S. (2020). Rising atmospheric CO_2 concentration inhibits nitrate assimilation in shoots but enhances it in roots of C3 plants. *Physiologia Plantarum*, *168*(4), 963–972. https://doi.org/10.1111/ppl.13040

Chamizo-Ampudia, A., Sanz-Luque, E., Llamas, A., Galvan, A., & Fernandez, E. (2017). Nitrate Reductase Regulates Plant Nitric Oxide Homeostasis. *Trends in Plant Science*, *22*(2), 163–174. https://doi.org/10.1016/j.tplants.2016.12.001

Chen, G. Y., Yong, Z. H., Liao, Y., Zhang, D. Y., Chen, Y., Zhang, H. B., Chen, J., Zhu, J. G., & Xu, D. Q. (2005). Photosynthetic acclimation in rice leaves to free-air CO_2 enrichment related to both ribulose-1,5-bisphosphate carboxylation limitation and ribulose-1,5-bisphosphate regeneration limitation. *Plant and Cell Physiology*, *46*(7), 1036–1045. https://doi.org/10.1093/pcp/pci113

Cheng, C., Yun, K. Y., Ressom, H. W., Mohanty, B., Bajic, V. B., Jia, Y., Yun, S. J., & de los Reyes, B. G. (2007). An early response regulatory cluster induced by low temperature and hydrogen peroxide in seedlings of chilling-tolerant japonica rice. *BMC Genomics*, *8*, 1–18. https://doi.org/10.1186/1471-2164-8-175

Chia, D. W., Yoder, T. J., Reiter, W. D., & Gibson, S. I. (2000). Fumaric acid: An overlooked form of fixed carbon in Arabidopsis and other plant species. *Planta*, *211*(5), 743–751. https://doi.org/10.1007/s004250000345

Choi, Y. A., Kim, S. G., & Kwon, Y. M. (1999). The plastidic glutamine synthetase activity is directly modulated by means of redox change at two unique cysteine residues. *Plant Science*, *149*(2), 175–182. https://doi.org/10.1016/S0168-9452(99)00163-6

Clerc, M. (2012). *Standard Particle Swarm Optimisation From 2006 to 2011*. 1. http://dx.doi.org/10.1007/978-3-540-71605-1_12

Cousins, A. B., Pracharoenwattana, I., Zhou, W., Smith, S. M., & Badger, M. R. (2008). Peroxisomal malate dehydrogenase is not essential for photorespiration in arabidopsis but its absence causes an increase in the stoichiometry of photorespiratory CO_2 release. *Planta*, *148*, 786–795. https://doi.org/10.1104/pp.108.122622

Dabu, X., Li, S., Cai, Z., Ge, T., & Hai, M. (2019). The effect of potassium on photosynthetic acclimation in cucumber during CO_2 enrichment. *Photosynthetica*, *57*(2), 640–645. https://doi.org/10.32615/ps.2019.073

Douce, R., Bourguignon, J., Neuburger, M., & Rébeillé, F. (2001). The glycine decarboxylase system: A fascinating complex. *Trends in Plant Science*, *6*(4), 167–176. https://doi.org/10.1016/S1360-1385(01)01892-1

Drake, B. G., Gonzàlez-Meler, M. A., & Long, S. P. (1997). MORE EFFICIENT PLANTS: A Consequence of Rising Atmospheric CO_2? *Annual Review of Plant Physiology and Plant Molecular Biology*, *48*(1), 609–639. https://doi.org/10.1146/annurev.arplant.48.1.609

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *Sixth International Symposium on Micro Machine and Human Science*, 39–43. https://ieeexplore.ieee.org/abstract/document/494215.?casa_token=VRHbIOq0xY0AAAAA:tigoKrFPGIOWOZPL3HUCxeJDuwpHdMr7AdrNcyfXSzfY9zdeQ3AAVzx9vd-b63ZQ8Q1ZwFq8E5okfcE

Eddelbuettel, D, & Francois, R. (2011). Rcpp: Seamless R and C plus plus Integration. *Journal of Statistical Software*, *40*(8), 1–18.

Eddelbuettel, Dirk. (2021). *tidyCpp: Tidy C++ Header-Only Definitions for Parts of the C API of R*. https://cran.r-project.org/package=tidyCpp

Eddelbuettel, Dirk, & Balamuta, J. J. (2018). Extending R with C++: A Brief Introduction to Rcpp. *American Statistician*, *72*(1), 28–36. https://doi.org/10.1080/00031305.2017.1375990

Florian, A., Araújo, W. L., & Fernie, A. R. (2013). New insights into photorespiration obtained from metabolomics. *Plant Biology*, *15*(4), 656–666. https://doi.org/10.1111/j.1438-8677.2012.00704.x

Gámez, A. L., Vicente, R., Sanchez-Bragado, R., Jauregui, I., Morcuende, R., Goicoechea, N., & Aranjuelo, I. (2020). Differential Flag Leaf and Ear Photosynthetic Performance Under Elevated (CO_2) Conditions During Grain Filling Period in Durum Wheat. *Frontiers in Plant Science*, *11*(December), 1–12. https://doi.org/10.3389/fpls.2020.587958

Gauthier, P. P. G., Bligny, R., Gout, E., Mahé, A., Nogués, S., Hodges, M., & Tcherkez, G. G. B. (2010). In folio isotopic tracing demonstrates that nitrogen assimilation into glutamate is mostly independent from current CO_2 assimilation in illuminated leaves of Brassica napus. *New Phytologist*, *185*(4), 988–999. https://doi.org/10.1111/j.1469-8137.2009.03130.x

Gomes Silveira, J. A., De Almeida Viégas, R., Almeida Da Rocha, I. M., De Oliveira Monteiro Moreira, A. C., De Azevedo Moreira, R. D., & Abreu Oliveira, J. T. (2003). Proline accumulation and glutamine synthetase activity are increased by salt-induced proteolysis in cashew leaves. *Journal of Plant Physiology*, *160*(2), 115–123. https://doi.org/10.1078/0176-1617-00890

Gong, Y. J., & Zhang, J. (2013). Small-world particle swarm optimization with topology adaptation. *GECCO 2013 - Proceedings of the 2013 Genetic and Evolutionary Computation Conference*, *July*, 25–31. https://doi.org/10.1145/2463372.2463381

Griffin, K. L., & Seemann, J. R. (1996). Plants, CO_2 and photosynthesis in the 21st century. *Chemistry and Biology*, *3*(4), 245–254. https://doi.org/10.1016/S1074-5521(96)90104-0

Härdtlein, J., Pflaum, C., Linke, A., & Wolters, C. H. (2010). Advanced expression templates programming. *Computing and Visualization in Science*, *13*(2), 59–68. https://doi.org/10.1007/s00791-009-0128-2

Hatfield, J. L., & Prueger, J. H. (2015). Temperature extremes: Effect on plant growth and development. *Weather and Climate Extremes*, *10*, 4–10. https://doi.org/10.1016/j.wace.2015.08.001

He, Y., Ma, W. J., & Zhang, J. P. (2016). The Parameters Selection of PSO Algorithm influencing On performance of Fault Diagnosis. *MATEC Web of Conferences*, *63*(2016), 02019. https://doi.org/10.1051/matecconf/20166302019

Heldt, H.-W., & Piechulla, B. (2015). *Pflanzenbiochemie*. https://doi.org/10.1007/978-3-662-44398-9

Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., & Woodward, C. S. (2005). SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software*, *31*(3), 363–396. https://doi.org/10.1145/1089014.1089020

Kennedy, J. (1997). The particle swarm : Social adaptation of knowledge. *IEEE*, 303–308.

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *IEEE*, 1942–1948. https://doi.org/10.1002/9780470612163

Kizildeniz, T., Pascual, I., Irigoyen, J. J., & Morales, F. (2021). Future CO_2, warming and water deficit impact white and red Tempranillo grapevine: Photosynthetic acclimation to elevated CO_2 and biomass allocation. *Physiologia Plantarum*, *172*(3), 1779–1794. https://doi.org/10.1111/ppl.13388

Koffler, B. E., Bloem, E., Zellnig, G., & Zechmann, B. (2013). High resolution imaging of subcellular glutathione concentrations by quantitative immunoelectron microscopy in different leaf areas of Arabidopsis. *Micron*, *45*, 119–128. https://doi.org/10.1016/j.micron.2012.11.006

Krämer, K., Krämer J., Heyer A., of Stuttgart, U., of Biomaterials, I., & file AUTHORS, B. S. (2021). *paropt: Parameter Optimizing of ODE-Systems*. https://cran.r-project.org/package=paropt

Krämer, K., Kepp, G., Brock, J., Stutz, S., & Heyer, A. G. (2022a). Acclimation to elevated CO_2 affects the C / N balance by reducing de novo N-assimilation. *Physiologia Plantarum*, *August 2021*, 1–13. https://doi.org/10.1111/ppl.13615

Krämer, K., Brock, J., & Heyer, A. G. (2022b). Interaction of Nitrate Assimilation and Photorespiration at Elevated CO2. *Frontiers in Plant Science*, *13*(July). https://doi.org/10.3389/fpls.2022.897924

Krämer, K. (2022c). ast2ast: Translates an r Function to a c++ Function.

https://CRAN.R-project.org/package=ast2ast.

Krämer, K., Krämer, J., Heyer, A., of Stuttgart, U., of Biomaterials, I., & at the University of Stuttgart | file AUTHORS, B. S. (2021). *paropt: Parameter Optimizing of ODE-Systems*. https://cran.r-project.org/package=paropt

Krapp, A., Hofmann, B., Schäfer, C., & Stitt, M. (1993). Regulation of the expression of rbcS and other photosynthetic genes by carbohydrates: a mechanism for the 'sink regulation' of photosynthesis? *The Expression of Negation*, *3*, 817–828. https://doi.org/10.1515/9783110219302

Küstner, L., Nägele, T., & Heyer, A. G. (2019). Mathematical modeling of diurnal patterns of carbon allocation to shoot and root in Arabidopsis thaliana. *Npj Systems Biology and Applications*, *5*(1), 1–11. https://doi.org/10.1038/s41540-018-0080-1

Leegood, R. C., Lea, P. J., Adcock, M. D., & Häusler, R. E. (1995). The regulation and control of photorespiration. *Journal of Experimental Botany*, *46*(special_issue), 1397–1414. https://doi.org/10.1093/jxb/46.special_issue.1397

Li, J., Weraduwage, S. M., Preiser, A. L., Tietz, S., Weise, S. E., Strand, D. D., Froehlich, J. E., Kramer, D. M., Hu, J., & Sharkey, T. D. (2019). A cytosolic bypass and g6p shunt in plants lacking peroxisomal hydroxypyruvate reductase1. *Plant Physiology*, *180*(2), 783–792. https://doi.org/10.1104/pp.19.00256

Li, P., Ainsworth, E. A., Leakey, A. D. B., Ulanov, A., Lozovaya, V., Ort, D. R., & Bohnert, H. J. (2008). Arabidopsis transcript and metabolite profiles: Ecotype-specific responses to open-air elevated CO_2. *Plant, Cell and Environment*, *31*(11), 1673–1687. https://doi.org/10.1111/j.1365-3040.2008.01874.x

Lillo, C., Meyer, C., & Ruoff, P. (2001). The nitrate reductase circadian system. The central clock dogma contra multiple oscillatory feedback loops. *Plant Physiology*, *125*(4), 1554–1557. https://doi.org/10.1104/pp.125.4.1554

Lin, S.-H., Kuo, H.-F., Canivenc, G., Lin, C.-S., Lepetit, M., Hsu, P.-K., Tillard, P., Lin, H.-L., Wang, Y.-Y., Tsai, C.-B., Gojon, A., & Tsay, Y.-F. (2008). Mutation of the Arabidopsis NRT1.5 Nitrate Transporter Causes Defective Root-to-Shoot Nitrate Transport. *The Plant Cell Online*, *20*(9), 2514–2528. https://doi.org/10.1105/tpc.108.060244

Linka, M., & Weber, A. P. M. (2005). Shuffling ammonia between mitochondria and plastids during photorespiration. *Trends in Plant Science*, *10*(10), 461–465. https://doi.org/10.1016/j.tplants.2005.08.002

Lisec, J., Schauer, N., Kopka, J., Willmitzer, L., & Fernie, A. R. (2006). Gas chromatography mass spectrometry-based metabolite profiling in plants. *Nature Protocols*, *1*(1), 387–396. https://doi.org/10.1038/nprot.2006.59

Loudon, K., & Grimm, R. (2018). C++ kurz & gut. In *O'Reilley*.

Ludewig, U., Neuhäuser, B., & Dynowski, M. (2007). Molecular mechanisms of ammonium transport and accumulation in plants. *FEBS Letters*, *581*(12), 2301–2308. https://doi.org/10.1016/j.febslet.2007.03.034

Masclaux-Daubresse, C., Daniel-Vedele, F., Dechorgnat, J., Chardon, F., Gaufichon, L., & Suzuki, A. (2010). Nitrogen uptake, assimilation and remobilization in plants: Challenges for sustainable and productive agriculture. *Annals of Botany*, *105*(7), 1141–1157. https://doi.org/10.1093/aob/mcq028

Meek, T. D., & Villafranca, J. J. (1980). Kinetic Mechanism of Escherichia Coli Glutamine Synthetase. *Biochemistry*, *19*(24), 5513–5519. https://doi.org/10.1021/bi00565a008

Miller, A. J., & Cramer, M. D. (2005). Root nitrogen acquisition and assimilation. In *Plant and Soil* (Vol. 274, Issues 1–2). https://doi.org/10.1007/s11104-004-0965-1

Miller, G., Suzuki, N., Ciftci-Yilmaz, S., & Mittler, R. (2010). Reactive oxygen species homeostasis and signalling during drought and salinity stresses. *Plant, Cell and Environment*, *33*(4), 453–467. https://doi.org/10.1111/j.1365-3040.2009.02041.x

Nägele, T., Henkel, S., Hörmiller, I., Sauter, T., Sawodny, O., Ederer, M., & Heyer, a. G. (2010). Mathematical Modeling of the Central Carbohydrate Metabolism in Arabidopsis Reveals a Substantial Regulatory Influence of Vacuolar Invertase on Whole Plant Carbon Metabolism. *Plant Physiology*, *153*(1), 260–272. https://doi.org/10.1104/pp.110.154443

Noguero, M., & Lacombe, B. (2016). Transporters involved in root nitrate uptake and sensing by Arabidopsis. *Frontiers in Plant Science*, *7*(September), 1–7. https://doi.org/10.3389/fpls.2016.01391

Peterhansel, C., & Maurino, V. G. (2011). Photorespiration redesigned. *Plant Physiology*, *155*(1), 49–55. https://doi.org/10.1104/pp.110.165019

Rachmilevitch, S., Cousins, A. B., & Bloom, A. J. (2004). Nitrate assimilation in plant shoots depends on photorespiration. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(31), 11506–11510. https://doi.org/10.1073/pnas.0404388101

Rackauckas, C. (2018). *A Comparison Between Differential Equation Solver*. 1–13. https://doi.org/10.15200/winn.153459.98975

Rebeille, F., Neuburger, M., & Douce, R. (1994). Interaction between glycine decarboxylase, serine hydroxymethyltransferase and tetrahydrofolate polyglutamates in pea leaf mitochondria. *Biochemical Journal*, *302*(1), 223–228. https://doi.org/10.1042/bj3020223

Renné, P., Dreßen, U., Hebbeker, U., Hille, D., Flügge, U. I., Westhoff, P., & Weber, A. P. M. (2003). The Arabidopsis mutant dct is deficient in the plastidic glutamate/malate translocator DiT2. *Plant Journal*, *35*(3), 316–331. https://doi.org/10.1046/j.1365-313X.2003.01806.x

Sahay, S., Robledo-Arratia, L., Glowacka, K., & Gupta, M. (2021). Root NRT, NiR, AMT, GS, GOGAT and GDH expression levels reveal NO and ABA mediated drought tolerance in Brassica juncea L. *Scientific Reports*, *11*(1), 1–15. https://doi.org/10.1038/s41598-021-86401-0

Sanderson, C., & Curtin, R. (2018). A User-Friendly Hybrid Sparse Matrix Class in C++. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10931 LNCS*, 422–430. https://doi.org/10.1007/978-3-319-96418-8_50

Sengupta, S., Basak, S., & Peters, R. (2018). Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives. *Machine Learning and Knowledge Extraction*, *1*(1), 157–191.

Seyoshi, K., Ishikawa, S., & Abdel-latif, H. I. S. (2010). Nitrate transport in barley. *Nitrogen Assimiliation in Plants*, *June 2014*.

Sharkey, T. D. (1988). Estimating the rate of photorespiration in leaves. *Physiologia Plantarum*, *73*(1), 147–152. https://doi.org/10.1111/j.1399-3054.1988.tb09205.x

Shi, S., Xu, X., Dong, X., Xu, C., Qiu, Y., & He, X. (2021). Photosynthetic Acclimation and Growth Responses to Elevated CO_2 Associate with Leaf Nitrogen and Phosphorus Concentrations in Mulberry (Morus multicaulis Perr .). *Forests (2021) 12(6)*, 1–14.

Siegenthaler, U., & Oeschger, H. (1978). Predicting future atmospheric carbon dioxide levels. *Science*, *199*(4327), 388–395. https://doi.org/10.1126/science.199.4327.388

Slot, M., & Kitajima, K. (2015). General patterns of acclimation of leaf respiration to elevated temperatures across biomes and plant types. *Oecologia*, *177*(3), 885–900. https://doi.org/10.1007/s00442-014-3159-4

Smith, N. G., & Dukes, J. S. (2013). Plant respiration and photosynthesis in global-scale models: Incorporating acclimation to temperature and CO_2. *Global Change Biology*, *19*(1), 45–63. https://doi.org/10.1111/j.1365-2486.2012.02797.x

South, P. F., Cavanagh, A. P., Liu, H. W., & Ort, D. R. (2019). Synthetic glycolate metabolism pathways stimulate crop growth and productivity in the field. *Science*, *363*(6422), 0–10. https://doi.org/10.1126/science.aat9077

Strodtkötter, I., Padmasree, K., Dinakar, C., Speth, B., Niazi, P. S., Wojtera, J., Voss, I., Do, P. T., Nunes-Nesi, A., Fernie, A. R., Linke, V., Raghavendra, A. S., & Scheibe, R. (2009). Induction of the AOX1D isoform of alternative oxidase in A. thaliana T-DNA insertion lines lacking isoform AOX1A is insufficient to optimize photosynthesis when treated with antimycin a. *Molecular Plant*, *2*(2), 284–297. https://doi.org/10.1093/mp/ssn089

Sunil, B., Saini, D., Bapatla, R. B., Aswani, V., & Raghavendra, A. S. (2019). Photorespiration is complemented by cyclic electron flow and the alternative oxidase pathway to optimize photosynthesis and protect against abiotic stress.

Photosynthesis Research, *139*(1–3), 67–79. https://doi.org/10.1007/s11120-018-0577-x

Syukuro, M., & Richard T., W. (1980). On the Distribution of Climate Change Resulting from an Increase in CO_2 Content of the Atmosphere. *Journal of the Atmospheric Sciences*, *37*(1), 99–118.

Syukuro, M., & T.Wetherald, R. (1974). The effects of Doubling the CO_2 Concentration on the Climate of a General Circulation Model. *Journal of the Atmospheric Sciences*, *32*(1).

Szulejko, J. E., Kumar, P., Deep, A., & Kim, K. H. (2017). Global warming projections to 2100 using simple CO_2 greenhouse gas modeling and comments on CO_2 climate sensitivity factor. *Atmospheric Pollution Research*, *8*(1), 136–140. https://doi.org/10.1016/j.apr.2016.08.002

Team, R. C. (2017). R language definition. *ACM SIGDA Newsletter*. https://doi.org/10.1145/1061414.1061416

Timm, S., Florian, A., Jahnke, K., Nunes-Nesi, A., Fernie, A. R., & Bauwe, H. (2011). The hydroxypyruvate-reducing system in arabidopsis: Multiple enzymes for the same end. *Plant Physiology*, *155*(2), 694–705. https://doi.org/10.1104/pp.110.166538

Timm, S., Nunes-Nesi, A., Florian, A., Eisenhut, M., Morgenthal, K., Wirtz, M., Hell, R., Weckwerth, W., Hagemann, M., Fernie, A. R., & Bauwe, H. (2021). Metabolite profiling in arabidopsis thaliana with moderately impaired photorespiration reveals novel metabolic links and compensatory mechanisms of photorespiration. *Metabolites*, *11*(6). https://doi.org/10.3390/metabo11060391

Timm, S., Nunes-Nesi, A., Pärnik, T., Morgenthal, K., Wienkoop, S., Keerberg, O., Weckwerth, W., Kleczkowski, L. A., Fernie, A. R., & Bauwe, H. (2008). A cytosolic pathway for the conversion of hydroxypyruvate to glycerate during photorespiration in Arabidopsis. *Plant Cell*, *20*(10), 2848–2859. https://doi.org/10.1105/tpc.108.062265

Titus, J. G., & Narayanan, V. (1996). The risk of sea level rise: A delphic Monte Carlo analysis in which twenty researchers specify subjective probability distributions for model coefficients within their respective areas of expertise. In *Climatic Change* (Vol. 33, Issue 2). https://doi.org/10.1007/bf00140246

Tsay, Y. F., Chiu, C. C., Tsai, C. B., Ho, C. H., & Hsu, P. K. (2007). Nitrate transporters and peptide transporters. *FEBS Letters*, *581*(12), 2290–2300. https://doi.org/10.1016/j.febslet.2007.04.047

Tsay, Y. F., Schroeder, J. I., Feldmann, K. A., & Crawford, N. M. (1993). The Herbicide Sensitivity Gene C M .1 of Arabidopsis Encodes a Nitrate-Inducible Nitrate Transporter. *Cell Press*, *72*(1), 705–713. https://doi.org/10.6119/JMST-014-0206-1

Tungate, K. D., Israel, D. W., Watson, D. M., & Rufty, T. W. (2007). Potential changes in weed competitiveness in an agroecological system with elevated temperatures. *Environmental and Experimental Botany*, *60*(1), 42–49. https://doi.org/10.1016/j.envexpbot.2006.06.001

Twigg, C. (2003). Catmull-Rom splines. *Computer*, *41*(6), 4–6.

Voss, I., Sunil, B., Scheibe, R., & Raghavendra, A. S. (2013). Emerging concept for the role of photorespiration as an important part of abiotic stress response. *Plant Biology*, *15*(4), 713–722. https://doi.org/10.1111/j.1438-8677.2012.00710.x

Walton, N. J., & Butt, V. S. (1981). Metabolism and decarboxylation of glycollate and serine in leaf peroxisomes. *Planta*, *153*(3), 225–231. https://doi.org/10.1007/BF00383891

Wickham, H. (2015). Advanced R. In *Journal of Japan Institute of Electronics Packaging* (Vol. 14, Issue 5). https://doi.org/10.5104/jiep.14.338

Wingler, A., Lea, P. J., Quick, W. P., & Leegood, R. C. (2000). Photorespiration: Metabolic pathways and their role in stress protection. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *355*(1402), 1517–1529. https://doi.org/10.1098/rstb.2000.0712

Wong, S. C. (1990). Elevated atmospheric partial pressure of CO_2 and plant growth - II. Non-structural carbohydrate content in cotton plants and its effect on growth parameters. *Photosynthesis Research*, *23*(2), 171–180. https://doi.org/10.1007/BF00035008

Zambrano-Bigiarini, M., Clerc, M., & Rojas, R. (2013). Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements. *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, 2337–2344. https://doi.org/10.1109/CEC.2013.6557848

Zhang, Z. S., Liu, M. J., Scheibe, R., Selinski, J., Zhang, L. T., Yang, C., Meng, X. L., & Gao, H. Y. (2017). Contribution of the Alternative Respiratory Pathway to PSII Photoprotection in C3 and C4 Plants. *Molecular Plant*, *10*(1), 131–142. https://doi.org/10.1016/j.molp.2016.10.004

KONRAD KRÄMER

Doktorand

konrad_kraemeryahoo.de +49 1573 7960497 Fasanenhofstrasse 74b, 70565 Stuttgart
konrad1991 konradkraemer91

WERDEGANG

Promotion im Bereich Pflanzenphysiologie/Systembiologie

Universität Stuttgart

01/2018-12/2021 Stuttgart

- Publikation der Software *paropt* (Konrad Krämer, 2021)
- Metabolomanalysen (mittels GC-MS/MS)
- Simulationen metabolischer Netzwerke
- Software-Entwicklung
- Seminargestaltung-/ und Begleitung *Dynamische Modellierung*

Master in Technischer Biologie (1.0)

Universität Stuttgart

03/2016-12/2017 Stuttgart/Potsdam

- HiWi, Institut für Pflanzenbiotechnologie, Universität Stuttgart
- Masterarbeit (1.0), Max Planck Institut für molekulare Pflanzenphysiologie in der Arbeitsgruppe *Zentraler Metabolismus*, Potsdam

Englisch-Sprachkurs (CAE C1)

EF School

01/2016-03/2016 Dublin

Bachelor in Technischer Biologie (1.9)

Universität Stuttgart

09/2012-12/2015 Stuttgart

- HiWi, Institut für technische Biochemie, Universität Stuttgart
- Bachelorarbeit (1.0), Institut für Biochemie, Universität Stuttgart
- Praktikum, Julius Kühn Institut für Pflanzenschutz, Siebeldingen

Praktikum

Eurofins

05/2012-07/2012 Niefern

Ausbildung zum BTA

Naturwissenschaftliches Technikum Dr. Künkele

03/2010-06/2012 Landau in der Pfalz

Allgemeine Hochschulreife (3.0)

Trifels Gymnasium Annweiler

03/2010 Annweiler am Trifels

ÜBER MICH

Hochmotivierter Biotechnologe mit 5 jähriger Erfahrung in Metabolomanalysen, enzymatischen Assays und wissenschaftlicher Datenanalyse (QTL mapping, GWAS, Simulationen, Optimierung, Statistik, RMarkdown, Visualisierung von Daten und wissenschaftlichem Programmieren in R, C++ und Fortran).

STÄRKEN

Analytisches Denken	●●●●●
Problemlösen	●●●●●
Begeisterung & Neugier	●●●●●

FÄHIGKEITEN

Metabolomanalysen	●●●●●
GC-MS/MS	●●●●○
Enzymanalysen	●●●○○

Statistik	●●●◐○
Data-Science	●●●●○

C++	●●●◐○
R	●●●●●
Fortran	●●◐○○

SPRACHEN

Englisch	●●●●○
Deutsch	●●●●●

PUBLIKATIONEN

Konrad Krämer, J. K., Arnd Heyer. (2021). *Paropt: Parameter optimizing of ode-systems* [R package version 0.2.1]. https://CRAN.R-project.org/package=paropt

ANHANG

Abitur-Zeugnis
BTA-Zeugnis
Bachelor-Zeugnis & Master-Zeugnis
CAE Zertifikat

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die Arbeit mit dem Titel *Interaktion von erhöhten* CO_2 *Konzentrationen und Stickstoff Metabolismus in Pflanzen* selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel genutzt habe. Alle wörtlich oder inhaltlich übernommenen Stellen habe ich als solche gekennzeichnet. Diese Dissertation wurde zu keiner Zeit in derselben oderähnlichen Version bei einem anderen Prüfungsamt eingereicht.

Ort, Datum

Unterschrift

Acknowledgement

First of all I would like to thank Mr. Heyer for giving me the opportunity to conduct my thesis at the department for Plant-Biotechnology. Beyond that I would like to thank for the inspiring discussions, the support in the laboratory and doing all the handicraft in order to grow plants at eCO_2. Furthermore, I'm grateful for giving me the possibility to explore all the research fields I was interested in. Especially, I would like to thank Mr. Heyer for supporting me during software development. In addition I'm really grateful that I had the opportunity to gather experience in educational work. Moreover, I would like to thank the Plant-Biotechnology team. First I would like to thank Imke for many discussions about plant metabolism. Simon and Gabi I would thank for all the help in the laboratory especially with the GC-MS/MS.

Furthermore, I would like to thank the greenhouse team for the excellent care of my plants. Thank you very much Annika and Nadja! I would also add that I'm really grateful for all the help with the hydroponic cultures.

In addition I would like to thank all the students who worked in the Plant-Biotechnology group during my doctoral thesis. Special thanks to Antonia it was always fun to write R code with her. Furthermore, I would like to thank Cedric and Rufus for their excellent work on different N sources and the interaction with PR. Moreover, I would like to thank Tobias, for testing my code and his enthusiasm about programming.

I would like to thank Jan for all his support in so many ways, our stimulating discussions in the course of my thesis and all the fun we have together.

Moreover, I would like to thank my family for all the support during my entire life. I'm really glad that you were always there.

Finally let me express a special thanks to my girlfriend Jana. Thank you for your supporting manner and all your patience with me. You encouraged and cared for me during the difficult phases of the thesis. I'm really grateful that you are on my side!