Assessing the impact of numerical precision on statistical fine-mapping and developing a comprehensive gene prioritisation tool for kidney function using genome-wide association meta-analyses

Dissertation
zur Erlangung des Doktorgrades
der Biomedizinischen Wissenschaften
(Dr. rer. physiol.)

der
Fakultät für Medizin
der Universität Regensburg

vorgelegt von
Kira Julia Stark geb. Stanzick
aus
Wolfenbüttel

im Jahr
2025

Assessing the impact of numerical precision on statistical fine-mapping and developing a comprehensive gene prioritisation tool for kidney function using genome-wide association meta-analyses

Dissertation
zur Erlangung des Doktorgrades
der Biomedizinischen Wissenschaften
(Dr. rer. physiol.)

der
Fakultät für Medizin
der Universität Regensburg

vorgelegt von
Kira Julia Stark geb. Stanzick
aus
Wolfenbüttel

im Jahr
2025

Dekan:	Prof. Dr. Dirk Hellwig
Betreuerin:	Prof. Dr. Iris Heid
Tag der mündlichen Prüfung:	24.10.2025

Table of Contents

1. Introduction

1.1. The global health burden of kidney disease and the lack of drugs and drug targets to treat it

This thesis has the topic of genetic influences on kidney function in the general population and how these can be measured, analysed, and interpreted. Thus, the first question arising is what the function of the kidney is in the first place.

The paired organ fulfils many tasks in the human body. These include: (1) regulation and maintenance of body fluid volume and osmolarity, (2) excretion of organic water-soluble metabolic end-products, inorganic substances and exogen supplied substances like drugs or vitamins, (3) regulation of blood pressure, (4) regulation of acid-base balance, (5) regulation of extracellular calcium level and (6) erythropoiesis [1]. The impairment of kidney function therefore has a variety of physiological consequences, which in the worst case can lead to death. Nevertheless, the treatment options for reduced kidney function are still very limited.

In epidemiological studies, as used in this work, mainly the second function mentioned above (excretion of different substances) is recorded and analysed. This is done by recording metabolites in blood and urine, such as creatinine, cystatin C, albumin, or urea, from which the filtration capacity of the kidney can be derived qualitatively or quantitatively [2]. Each kidney consists of about 1 million nephrons, which in turn consist of the glomerulus and the renal tubule [3–5]. The glomerulus comprises a tuft of capillaries which are surrounded by an epithelium called Bowman's capsule [3]. As the blood passes through the glomerular capillaries, the serum components are filtered across the glomerular filtration barrier [5]. This structure is formed by the capillary endothelium, the basement membrane and specialized cells of the glomerular epithelium (Podocytes) [3, 6]. Per day, about 180 L of primary filtrate are produced by the glomeruli of which the vast majority is reabsorbed in the tubules [5].

Chronically reduced kidney function or structural abnormalities define chronic kidney disease (CKD) [2]. Severeness of this disease is classified on the basis of glomerular filtration rate (GFR) and the detection or amount of albumin in the urine [2]. In epidemiological studies, the simple definition of CKD as an estimated GFR (eGFR) < 60 ml/1.73m^2 is often used (referring to stages 3-5 in the international KIDIGO classification system), as measurements of the corresponding biomarkers over a

longer period of time are often not available to take the chronification aspect into account [2, 7]. Based on this definition, the prevalence of CKD in Germany is around 2.3%, which corresponds to approximately 2 million people affected [7]. According to a meta-analysis of 33 studies assessing CKD prevalence, the global prevalence of CKD stages 3-5 is around 10.6%, ranging from 1.3% for men in China to 21.3% for women in Nepal [8]. Thus, the estimated number of adults with CKD is around 236 million people worldwide [8]. The consequences and complications of CKD are manifold. The main consequences are cardiovascular disease and kidney failure (end stage renal disease, ESRD), which makes transplantation or dialysis unavoidable for survival. Although only a small proportion of CKD patients develops ESRD, mortality rate increases drastically with higher stages of CKD [9]. To lower the global health burden of CKD a comprehensive understanding of the disease aetiology is required to identify new therapeutic approaches.

Many potential clinical risk factors for CKD are known, such as diabetes, hypertension, autoimmune disease, infectious disease, urolithiasis, lower urinary tract obstructions, hyperuricemia, family history of CKD or kidney injury [10, 11]. Additionally, the sociodemographic factors: age, ethnicity, smoking, body mass index, and exposure to chemicals such as lead and cadmium are associated with higher risk for CKD [10–13]. Nevertheless, individual aetiology remains unclear in most cases and the insufficient knowledge of molecular mechanisms leading to progressive loss of renal function results in a lack of therapeutic targets for drug development [14].

1.2. Introduction of genome-wide association studies (GWAS) and their usefulness to support drug development.

Genome-wide association studies (GWAS) search for associations between naturally occurring variations in the human genome and diseases, phenotypes or biomarkers using the genetic information of thousands or millions of individuals [15]. The 1000 Genomes Project reports around 88 million genetic variations in the human genome, of which 4.1 million to 5.0 million occur in one individuals genome [16]. Most of these alterations are single nucleotide polymorphisms (SNPs) with a frequency above 0.5% [16]. GWAS are a hypothesis free approach testing these millions of SNPs for association with the phenotype in question. However, none of the SNPs showing an

association must be a causal variant but are rather linked to the causative genetic element via a linkage disequilibrium (LD) [17]. The LD describes the probability that two genetic variants are transmitted together from one generation to the other due to proximity on the chromosome or due to selection or population stratification [17].

The first GWAS was published in 2005 and until 2023 over 400,0000 variant x trait associations have been reported [18, 19]. GWAS results can provide insights into the underlying biology of a phenotype, support the estimation of the heritability of a phenotype or prediction of the clinical risk [15]. They can also help with drug development or repurposing [20]. While the normal success rate for drug development is around 10 % the success rate of drug mechanisms with genetic support is estimated to be up to 2.6 times greater than for those without [21–23].

According to the GWAS Catalog, a comprehensive database for GWAS, 80 GWAS and GWAS meta-analysis for the parameter estimated glomerular filtration rate (eGFR) have been published [19]. The two most recent studies include both over one million individuals (partly overlapping data) and each report hundreds of associated loci [24, 25].

1.3. The missing link between GWAS results and biological processes.

Although numerous genetic regions (loci) with great statistical evidence for association have been identified, the prediction of causal variants, genes or biological mechanisms is still a challenging task [26]. One challenge is that the identified genetic loci often consist of many associated variants in high LD [27]. The process of identifying the most likely causal variant(s) within a locus is called fine-mapping and often the first step of understanding an association signal [27–29]. The different fine-mapping approaches vary in their complexity from assuming a single causal variant per signal to modelling multiple causal variants to integrate functional data into the statistical approaches [27, 28]. All fine-mapping approaches rely on LD reference panels to estimate the local LD-structure. Thus, the quality of their results depends on the matching of the reference panel. With increasing size of GWAS and GWAS meta-analyses this becomes a major challenge [30]. Besides the quality of the reference panel, which is influenced by ancestral composition and size, also the quality of the meta-analysis data has an impact on the fine-mapping results [30, 31]. For GWAS meta-analyses often the

software METAL is used [32, 33]. While the calculation is performed with the maximum available accuracy from the input data, the effect estimators and standard errors are stored with an accuracy of four decimal digits using the default settings [34]. For most parameters 4 decimal digits are sufficient to discriminate even small differences between SNP associations. However, due to clinical data included in GWAS meta-analysis data for eGFR, this parameter has been log-transformed to achieve a normal distribution [35]. This is necessary for the statistical analysis but leads to very small effect estimates and standard errors that only differ after the 4th decimal digit [35, 25]. Thus, the first aim of this thesis was to analyse which number of decimal digits are needed for robust fine-mapping in the context of log-transformed eGFR.

After fine-mapping, the most likely causal variants (credible set variants) are usually annotated regarding their effect on genes and proteins to determine the most likely causal gene [36]. For these annotations large publicly available datasets for gene expression, functional predictions of SNPs and overall biological pathways can be used so that comprehensive data is generated [36–41]. However, the results are frequently published as long supplementary tables, which can make accessing and interpreting this data challenging [35, 25, 42]. To improve the understanding of kidney function and prioritise GWAS results more effectively, tools that process and present this evidence in a more accessible format are needed. Therefore, the second aim of this work was to create such a tool making the eGFR GWAS and post-GWAS data easily accessible and user-friendly for a broader scientific community with a particular focus on aspects of kidney function.

Results of both aims are partly published in Stanzick et al. 2023 [43].

2. Methods

2.1. Description of data from UK Biobank, CKDGen, Stanzick et al. 2021

The first part of this thesis rebuilds the meta-analysis of GWAS data for estimated glomerular filtration rate (eGFR) from serum creatinine as published in Stanzick et al 2021 [25] with increased number of decimal digits. Therefore, the GWAS data from the UK Biobank and the CKDGen consortium for European ancestry were used.

The data from CKDGen consortium refers to the publication of Wuttke et al. 2019 [35] and includes summary statistics for eGFR from 567,460 Europeans. The summary statistics are the result of a meta-analysis of mostly population-based studies, but also implied family-based studies, clinical trials, and case-control studies [35]. Each study used an additive genetic model to perform regression on SNP dosage, after adjusting for sex, age, and study-specific features like study site, genetic principal components, or relatedness [35]. Imputation of non-measured SNPs was performed based in the Haplotype Reference Consortium (HRC, v1.1) or the 1000 Genomes project [35]. Due to the inclusion of clinical patients in some studies a natural logarithm was applied to the outcome to reduce the skewness of the eGFR distribution [35]. The association results with 6 decimal digits for effect estimates and 7 decimal digits for standard errors are publicly available at https://CKDGen.imbi.uni-freiburg.de (further referred to as 6/7 decimal digits dataset). The CKDGen GWAS meta-analysis results with 8 decimal digits were directly received from the CKDGen consortium.

The GWAS summary statistics for eGFR based data from the UK Biobank were published in my master thesis and Stanzick et al. 2021 [25]. The main aspects are briefly summarized below:

The UK Biobank study includes around 500,000 individuals from across the United Kingdom aged between 40 and 69 at the time of recruitment [44]. Genetic data was imputed to HRC and the UK10K haplotype reference panel, which resulted in 93,095,623 autosomal SNPs, short indels and large structural variants in 487,442 individuals [44]. After quality control, data from 436,581 individuals with self-reported European ancestry was available for linear mixed model GWAS conducted by Thomas Winkler with the fastGWA tool [45]. To achieve best compatibility with CKDGen data the natural logarithm was applied to eGFR values before GWAS. Age, age^2, age x sex,

age^2 x sex and 20 principal components were included as covariates in the association analysis [25].

The downloaded CKDGen data was already quality controlled at individual study and meta-analysis level. Per study, variants with low imputation quality (Info <0.6) and rare variants (minor allele count, MAC < 10) were excluded and genomic control (GC) correction was applied to all studies with GC inflation factor > 1 [35]. On meta-analysis level, SNPs available in less than 50% of all studies and with MAC<400 were excluded [35]. To harmonize the quality control procedure, the same criteria for imputation quality was applied to UK Biobank GWAS using EasyQC and results were stored with 6/7 and 8/8 decimal digits for effect estimate and standard error [46]. Additionally, UK Biobank GWAS as well as CKDGen data was reduced to common SNPs with minor allele frequency (MAF) > 0.1% which are included in both datasets. As the CKDGen dataset with 8 decimal digits was more extensive than the publicly available version, the SNPs were reduced to the SNPs included in the 6/7 decimal digits dataset after quality control. This guarantied the analysis of the same SNPs just with different number of decimal digits.

In the publication of Stanzick et al., 2021 the association statistics of CKDGen and UK Biobank were meta-analysed with fixed-effect inverse-variance weighted meta-analysis with the software METAL using standard settings (storing effect estimates and standard errors with 4 decimal digits) [25, 34]. Afterwards, SNPs were filtered for MAC > 400, resulting in 8,844,847 SNPs from 1,004,040 included individuals [25]. This dataset is used as comparative data set in this work.

2.2. GWAS meta-analysis between UK Biobank and CKDGen with 6/7, or 8/8 decimal digits for effect and standard error

In Stanzick et al., 2023, analogously to the meta-analysis published in Stanzick et al., 2021, fixed-effect inverse-variance weighted meta-analysis of the above described CKDGen and UK Biobank association results was performed using METAL [34, 25, 43]. In contrast to the original analysis, the two different datasets from CKDGen with 6/7 or 8/8 decimal digits for effect estimate and standard error were meta-analysed with UK Biobank with the same respective number of decimal digits, and the two meta-

analysis association results were stored with 6/7, or 8/8 decimal digits for effect estimates and standard error [43].

2.3. Step-wise conditioning with GCTA

Stanzick et al. 2021 presented 424 loci associated with eGFR at $p < 5x10^{-8}$ identified in an all-ancestry GWAS meta-analysis [25]. The lead variants of these loci (SNPs with the smallest p-value) were used as starting points for approximate conditional analysis inside each locus to identify independent signals. This was performed with the European ancestry GWAS results (with 4 decimal digits) and a LD reference panel based on 20K randomly selected unrelated Europeans from the UK Biobank using the software GCTA [47, 25]. As result of the stepwise conditioning process 634 independent signal were described in the publication [25]. For further analyses additional fully conditioned models were built by conditioning one signal in a locus to all other independent signals (represented by signal index variants) in the same locus [25].

As described in Stanzick et al. 2023, two approaches for conditioning were applied to the meta-analysis results with more decimal digits in this work to analyse the effect of the decimal digits on the conditioning results [43]. First, the conditioning was performed with the fixed list of 634 index SNPs to check whether these could also have been identified as index SNPs independent of the number of decimal digits [43]. And secondly, a new stepwise conditioning analysis was started beginning with the 424 locus lead variants to identify new or other independent signals [43]. For all conditional analyses GCTA and the same LD reference panel from 20K UK Biobank participants was used [43].

2.3.1. Using the previously published 634 index SNPs for conditioning

For each of the 424 loci the corresponding signal index variants were selected. In a first step, conditioning was applied in the same order as index variants were found in the original analysis by Stanzick et al. [25, 43]. For this purpose, each locus was first conditioned on the locus lead variant (first signal index variant) and the conditioned p-value of the next (second) signal index variant in the locus was noted as the identifying

p-value for the second signal index variant. In the second step, conditioning was performed on the lead variant and the second index variant to obtain the identifying p-value of the next index variant. This was repeated for all signal index variants in a locus. Lastly, the fully conditioned models were built as described above. This procedure was done for the 6/7 decimal digits and the 8/8 decimal digits analysis.

2.3.2. Naively identifying independent signals

For both GWAS meta-analysis datasets, a new stepwise conditioning analysis was performed [43]. The 424 loci lead variants and loci definitions from the all-ancestry analysis of Stanzick et al. 2021 were used [25]. For each locus, the eGFR association results were first conditioned on the lead variant. If the conditioned association statistic contained a variant with genome-wide significant p-value, this variant was considered as an index variant of a secondary independent association signal at its locus. In the next step, the original association statistics were conditioned on the lead variant and the index variant. If another variant showed a genome-wide significant p-value in the conditioned association statistics, it was also included in the list of index variants for this locus and the conditioning was repeated with all index variants (including the lead variant). This process was repeated until no more variants with genome-wide significant p-value were found in the conditioned association statistics. At the end, as in the analysis with 634 index variants, the fully conditioned models per locus were built [43].

2.4. Fine-mapping using the Wakefield method

To identify the most likely causal variants per association signal fine-mapping with the Wakefield method was applied, as done in Stanzick et al., 2021 [48, 25]. For this, the fully conditioned association statistics were used for each signal and approximate Bayes Factors (ABFs) and posterior probability of association (PPA) were calculated for each variant. As the prior variance of eGFR $W= 0.005^2$ was used as in Stanzick et al., 2021 [25]. 99% credible sets for each association signal were generated by adding the variants with the highest PPA to the set until a cumulative PPA >99% was reached. This was done based on the conditioning results using the 634 signal index variants

as well as for the new stepwise conditioning results for the 6/7 decimal digits and for the 8/8 decimal digits dataset.

2.5. Comparison of previous fine-mapping signals and credible set variants with analysis results of this work using more decimal digits

As published in Stanzick et al. 2023 [43], the comparison with the original data from Stanzick et al. 2021 [25] was split in two parts:

2.5.1. Comparison with the re-build 634 signals (Comparison of identifying p-values, fully conditioned p-values, credible set sizes and individual SNP PPA)

First, the results of the re-build 634 signals were compared. Since these were exactly the same signals in each data set, all analysis results could be directly compared. The identifying p-values of the index SNPs were compared to analyse whether the 634 signals could have been identified in the data sets with more decimal digits. Additionally, fine-mapping results were compared by looking at credible set sizes per signal and at individual SNP PPA.

2.5.2. Comparison with the newly identified signals

Signal index variants identified with the stepwise conditioning approach in the 6/7 decimal digits and 8/8 decimal digits datasets were not exactly the same as the 634 variants from Stanzick et al. 2021 [25, 43]. Therefore, signals were mapped to each other prior to comparison.

For mapping of signals, correlations (R^2) between the index variants described in Stanzick et al. 2021 [25] and the newly identified index variants in the same locus were calculated using the same reference panel of 20K individuals from UK Biobank as for the conditioning. Signals with the same signal index variant were stated as identical between the three different datasets (original analysis in Stanzick et al. 2021 [25], 6/7 decimal digits, and 8/8 decimal digits) even if the signal index variants of other signals in the locus were different. Correlations above 0.9 will further be named as “highly

correlated" signals, and correlations between 0.5 and 0.9 will be referred to as "correlated" signals. Signals without correlation to a previously described signal will be referred to as "not comparable".

After mapping the signals of the published analysis to the newly identified signals, credible set size and individual SNP PPA and ABFs of identical, highly correlated, and correlated signals were compared.

2.6. Pre-KidneyGPS data processing

The application described in this thesis is called KidneyGPS and was published in Stanzick et al. 2023 [43]. It uses the results of GWAS meta-analyses for different kidney related biomarkers and outcomes as a basis. Various external, freely available datasets are used for the data backbone of KidneyGPS to annotate SNPs and genes. A simplified overview of how the data is processed and incorporated in KidneyGPS is shown in **Figure 1**. The downloaded raw datasets are first transformed by merging data files from one or two sources. In a second transformation the annotation data is combined with the GWAS data. In the final step, all datasets are prepared in such a way that they can be processed easily and quickly by KidneyGPS.

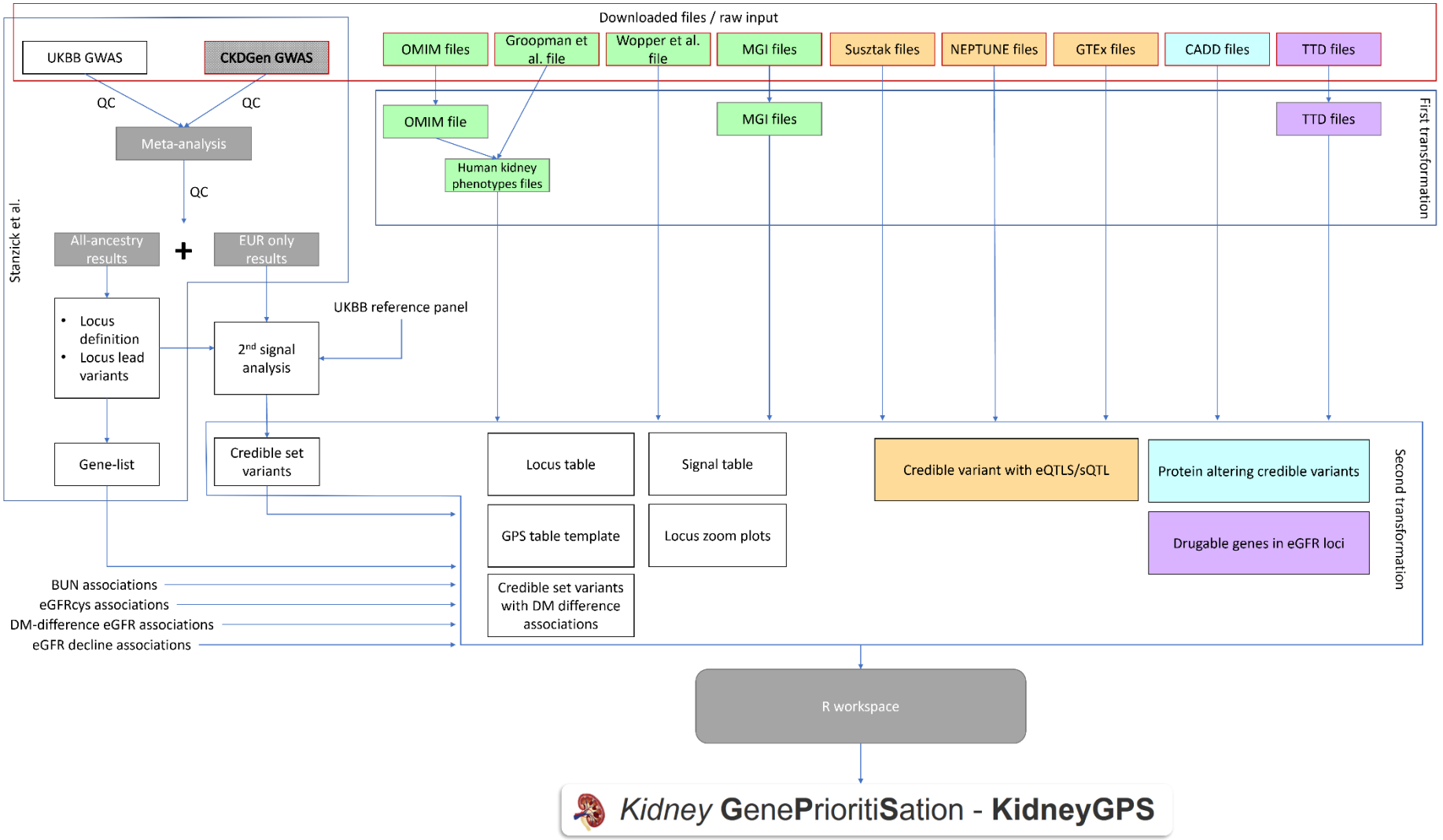


Figure 1: Data-processing steps needed to incorporate GWAS and post-GWAS data in KidneyGPS. GWAS data-processing is described in Stanzick et al. 2021 and was applied to the dataset with 6/7 decimal digits as published in Stanzick et al. 2023 [25, 43]. The downloaded annotation data was processed in a first transformation step that combines annotation data files from one or two sources. The second transformation step combines the GWAS data with the annotation data to generate the data backbone for KidneyGPS.

2.6.1. Raw data sets

The downloaded data sets can be divided into 4 groups based on the underlying type of data:

(1) Some genes are known to cause kidney phenotypes, either due to hereditary diseases in humans or due to genetically modified mice. Two lists of genes (potentially) causing kidney phenotypes in human were extracted from the publications of Groopman et al. and Wopperer et al. [49, 50]. Additionally, data from the databases 'Online Mendelian inheritance in men' (OMIM) and 'Mouse Genome Informatics' (MGI) were downloaded [51, 52]. OMIM files were downloaded via API requests (see appendix), which queried for all phenotype entries subordinate to the clinical synopsis class "kidney" (done August 7th, 2020). Additionally, a MIM number reference file was downloaded from the OMIM website (March 9th, 2020) which was later used in KidneyGPS to create static links to the respective OMIM gene entries. Download from MGI was not restricted to kidney specific entries but contained all available phenotype entries and corresponding genes (http://www.informatics.jax.org/ downloads/reports/index.html; last visited March 6th, 2020; files: MPheno_OBO.ontology, gene_association.mgi.gz, HMD_HumanPhenotype.rpt, MGI_PhenoGenoMP.rpt).

(2) Some SNPs, the so called eQTLs (expression quantitative trait loci), have impact on a gene's expression. These effects are found with association studies between SNPs and tissue specific gene expression. Two databases provide kidney tissue specific eQTL data: NephQTL (NEPTUNE) and Susztaklab [53, 24]. Both distinguish between glomerular tissue and tubulo-interstitial tissue. The third source of eQTL data used in this work is the GTEx database [37]. eQTL data for 44 tissues including kidney cortex was downloaded. In addition, GTEx also provides information weather a SNP has effect on gene-splicing in different tissues. This splice QTL (sQTL) data was also downloaded.

(3) The effects of SNPs on protein-level are predicted, among others, by the Ensemble Variant Effect Predictor (VEP) [54]. The combined annotation dependent depletion (CADD) database uses these predicted effects to calculate PHRED Scores, which can be used to evaluate a SNPs deleteriousness [38]. This indicates how likely a SNP is to persist in the general population as a result of evolutionary selection. SNPs with a

high deleteriousness have such a strong predicted effect on the protein that they have a low probability of surviving selection (negative selection). Datasets from the CADD database were used as they included both: the predicted effects from VEP as well as the PHRED Scores.

(4) Data files from the Therapeutic Target Database (TTD) were downloaded to map genes to approved drugs and drugs under development (https://db.idrblab.net/ttd/full-data-download October 5th, 2022) [55].

In addition, two datasets with reference gene names were downloaded to translate platform specific gene IDs to official HGNC (HUGO Gene Nomenclature Committee) gene names and Uniprot protein names. The first file was downloaded from https://www.genenames.org/download/custom/ (August 18th, 2022) and contains HGNC ID, Gene names, alias symbols, RefSeq IDs, Ensemble gene IDs, NCBI Gene IDs and Pubmed IDs. The second file was downloaded from https://www.uniprot.org/ (July 7th, 2022) and contains Uniprot Gene-IDs, Protein names and Gene names including alias.

2.6.2. Per dataset/data source processing

The datasets of OMIM, Groopman et al. and MGI had to be processed before combining them with the GWAS data. This process included script-based processing as well as manual curation. The data from OMIM and Groopman et al. were combined to a dataset of genes causing kidney phenotypes in human. Therefore, the OMIM API requests were combined by Thomas Winkler and then a script was used to exclude entries, which only contained the phenotypes “Normal kidneys {UMLS C1864785}”, “Normal renal ultrasound at ages 4 and 7 (in 2 family) {UMLS C4229721}”, “No kidney disease {UMLS C1838421}”, “No renal disease {UMLS C1857414}”, “Normal renal function {SNOMEDCT:81141003} {UMLS C0232805}”, “Normal renal function {SNOMEDCT:81141003} {UMLS C0232805}”, “No kidney disease {UMLS C1838421}”, or “No renal findings {UMLS C4230312}”. After merging the OMIM data with the 625 genes from Groopman et al. a script removed duplicates that were included in both datasets. Manual curation was needed for entries only differing at one word, e.g. “hereditary” vs “familial”.

To generate a list of genes causing kidney phenotypes in mice several processing steps were applied to the MGI data files. First, the list of phenotypes contained in the file MPheno_OBO.ontology was reduced to those phenotypes subordinate to “abnormal kidney morphology MP:0002135” or “abnormal kidney physiology MP:0002136”. In parallel, a table with a 1:1 mapping of gene IDs and phenotype IDs was created from file MGI_PhenoGenoMP.rpt. Both gene IDs and phenotype IDs can occur more than once. Mouse gene names, phenotype names and human gene names were then added using the downloaded HMD_HumanPhenotype.rpt file. This complete list of phenotypes, causative genes and human gene names was reduced to kidney phenotypes using the first created phenotype file.

2.6.3. Combining annotation data with GWAS data

The basis of KidneyGPS are the fine-mapping results presented in chapter 3.1.2 of this thesis. All variant based annotation data (eQTLs, sQTLs, CADD) were reduced to the list of 99% credible set variants. In addition, eQTL effect alleles were switched to the eGFR lowering allele and gene names were translated to the HGNC gene names. The NephQTL data was reduced to significant eQTLs (False discovery rate<0.05), while data from the Susztaklab only contained such significant eQTLs. CADD data was further restricted to variants with a PHRED Score ≥ 15 (which represents the top 3.2% most deleterious variants) or to variants with predicted effect of “stop-gained”, “stop-lost” or “non-synonymous” (highest scoring consequences with consequence score “ConsScore” 8 or 7). For all variants only the gene-variant combination with the highest “ConsScore” was retained.

The genes included in KidneyGPS are obtained from the eGFR associated loci, published previously [25]. All gene-based annotation data (human or mice kidney phenotypes, drug targets) were reduced to these genes after harmonization with HGNC gene names.

2.6.4. Co-localization analysis

Another form of combing GWAS data with eQTL data is the co-localization analysis. As done in Stanzick et al. 2021 [25] the eGFR association signals identified in this

works were analysed regarding their overlap with the eQTL signals from the NephQTL data using the method described by Giambartolomei et al. and the R package gtx [56]. Prior variances were set identically to Stanzick et al. 2021 [25] with 0.005^2 for eGFR association and 0.55^2 and 0.53^2 for tubulo-interstitial and glomerular tissue expression variance.

2.6.5. GWAS data of other phenotypes

KidneyGPS also integrates the GWAS meta-analysis data from Stanzick et al. 2021 [25] for eGFR estimated from serum cystatin C (eGFRcys, N= 460,826) and blood urea nitrogen (BUN, N= 852,678). In addition, eGFR associations stratified by diabetes mellitus (DM) status were obtained from the publication of Winkler et al. 2022 [57]. These were used to annotate the credible variants described above with the stratified association statistics. Together with the eGFR decline effects reported by Gorski et al. 2022, the stratified association results were also used to create a list of eGFR signals, that show different eGFR association effects in people with diabetes or in people without diabetes or have an effect on eGFR decline [58].

2.6.6. Regional association plots

Regional association plots were created for all loci with Locuszoom software [59]. The unconditioned eGFR results and the respective signal index variants from conditional analysis were used. Plots were transferred to https://homepages.uni-regensburg.de/~wit59712/gps/ by Thomas Winkler to make them accessible. A list of links for all 424 plots was created to integrate them in KidneyGPS.

2.6.7. Preparing R-Workspace for KidneyGPS: Run_before_app.r

To reduce loading times, KidneyGPS runs with a R-Workspace that contains all relevant data and is already optimized for the searches or filtering of KindeyGPS. The code and description of the tables created and used later by KidneyGPS are included in the appendix of this work. In summary, the run_before_app.r script does the following: (1) It adds locus IDs and signal IDs of eGFR signals to annotation data tables

to make them searchable by these features. (2) The human gene information from OMIM and Groopman et al. was extended by the data of Wopperer et al. (3) Links to the OMIM website were created using the respective MIM numbers. (4) Credible set variant information (allele frequency, region ID, signal ID and PPA) was added to all variant based annotation data (eQTLs, sQTLs, protein-altering variants). (5) The eQTL data of Susztaklab and NephQTL was combined. (6) The column order and column names of various tables were changed. (7) Links to the NCBI website were created for each variant genome-wide significant associated with eGFR. (8) Association statistics were trimmed to two telling digits. (9) Some tables were mirrored so that the original table is easy to search while the mirror table contains special features for visual display (e.g. links to other websites, a column containing the data source, or descriptive columns like “upregulating” or “downregulating” for eQTL effects above or below 0). (10) A large gene prioritisation table (GPS table) was created containing all annotation information for each gene in each signal as well as the information of the other GWAS data (eGFRcys, BUN, DM stratified, and eGFR decline). (11) The same table as in (10) was created two more times, restricting the annotation data to variants with at least 10% PPA or 50% PPA.

3. Results

3.1. Impact of decimal digits in GWAS results on the number of association signals, credible set variants and prioritisation.

Second-signal analysis and fine-mapping are important post-GWAS analyses used to narrow down the association signals to likely causal variants. Second signal analysis tools like GCTA search for additional independent signals in genetic loci identified with GWAS by performing approximate conditional analyses. For this, the GWAS summary statistics as well as a reference panel are used. To evaluate the impact of decimal digits of the summary statistics on second signal identification and fine-mapping, two main analyses were performed: (1) The analysis of Stanzick et al. 2021 [25] was recreated with the same secondary signals but with summary statistics containing 6/7 or 8/8 decimal digits for effect estimate and standard error. The respective results were compared to the 4 decimal digits analyses from Stanzick et al. 2021 [25]. (2) The approximate conditional analysis on the summary statistics was redone with more decimal digits without predefining the secondary signals. Parts of the following results are published in Stanzick et al. 2023 [43] and marked by citation accordingly.

3.1.1. Re-creation of previous fine-mapping analysis with more decimal digits

3.1.1.1. Stability of signals

The original analysis with 4 decimal digits for effect estimate and standard error in the GWAS meta-analysis for eGFRcrea by Stanzick et al. identified 210 additional signals in the 424 GWAS loci [25]. To investigate the robustness of these 634 signals, the identifying p-values for the 634 signals were compared between the 4/4 (Stanzick et al. 2021 [25]), 6/7 and 8/8 decimal digits analyses. The identifying p-values of the first signals were the same regardless of the number of decimal digits while the identifying p-values of higher order signals differed. This is because the identifying p-value of a first signal in a locus is the original p-value from the meta-analysis, while the identifying p-value of the second signal index SNP is the original p-value conditioned on the first index SNP, and the identifying p-value of the third signal index SNP is the original p-value conditioned on the first and second signal index SNP and so on. **Figure 2** compares the identifying p-values of the original analysis with 4 decimal digits with the 6/7 decimal digits analysis. SNPs on the identity line are the 424 first signal index

variants. 143 of the additional 210 index SNPs have an identifying p-value below the genome-wide significant threshold of $5x10^{-8}$ [43]. In reverse, 67 index SNPs would have not been identified as such in the analysis with more decimal digits [43].

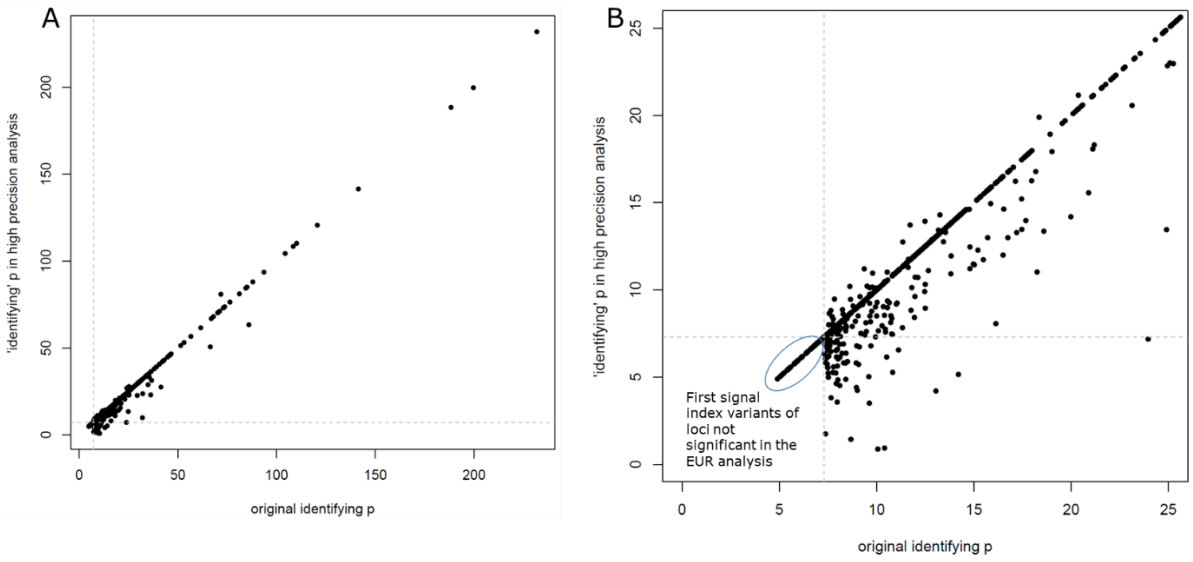


Figure 2: Figures from Stanzick et al. 2023 Supplementary Figure 1 panel A+B [43]: Comparison of the identifying p-values of the 634 signal index variants from the original 4 decimal digits analysis and the 6/7 decimal digits analysis. GWAMA results in EUR were conditioned with GCTA using the same 634 signal index SNPs and the 20K reference panel from UKBB as input. P-values are shown on a log10 scale. A: Overview of all 634 signal index variant identifying p-values. B: Zoom of panel A. Dots marked with the blue oval are index SNPs of a first signal in loci that are only significant in the all-ancestries analysis but not in EUR alone.

Locus regions for conditioning were defined by the all-ancestry meta-analysis from Stanzick et al. 2021 [25] while conditioning was performed with the European-only GWAS results. Thus, not all loci necessarily contain a genome-wide significant SNP in the European-only analysis (unconditioned). Therefore, the first signal index variant for these loci is just the SNP with smallest p-value. However, additional signal index variants must meet the criterion of genome-wide significance after conditioning to represent an additional independent signal. In total, 51 first signal index variants were not genome-wide significant. Nevertheless, these signals are considered as true signals even with their higher p-value in the European-only analysis. The largest p-value of these 51 SNPs was $1.265x10^{-5}$. Therefore, it seems acceptable if other signal index variants in the 6/7 decimal digits analysis only have a p-value smaller than $1.265x10^{-5}$. This reduces the number of potentially suspicious signals identified in the 4/4 decimal digits analysis to 15 signals which are described in detail in the following.

As displayed in **Table 1**, 11 of the 15 suspicious signals are the last one of their loci which means that their identifying p-value is identical to the fully conditioned p-value.

For the signals k1.3, k21.3, k37.4 this is not the case as these are the second last (k1.3 and k21.3) or the third last (k37.4) signal of their loci. The additional conditioning on the index variant of signal k1.4 which is also not genome-wide significant in the 6/7 decimal digits analysis (**Table 1**) partly leads to a recovery of signal k1.3, which has a fully conditioned p-value of 4.6×10^{-6}. This is still not genome-wide significant but acceptable under the above defined criteria. The same effect is visible for signal k37.4 which has a fully conditioned p-value of 5.31×10^{-6}. For the signal index variant of k21.3 the additional conditioning on the last index variant of its locus does not make a great difference. The fully conditioned p-value of k21.3 is 0.037 which is even higher than the identifying p-value.

The increase from 6/7 decimal digits to 8/8 decimal digits had nearly no effect on the identifying p-values (**Figure 3**) indicating no further effect on signal stability [43].

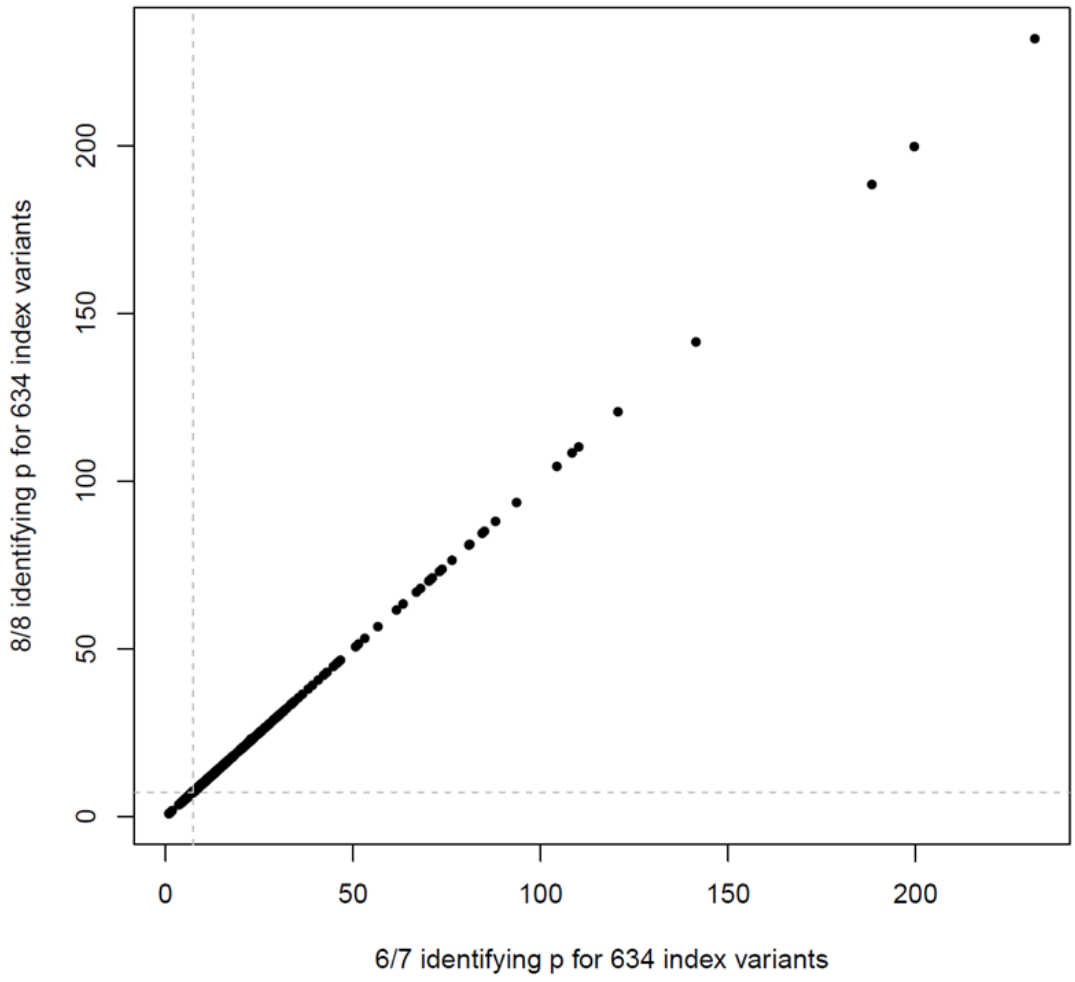


Figure 3: Figure from Stanzick et al. 2023 Supplementary Figure 1 Panel C [43]: Comparison of identifying p-values of the 6/7 and 8/8 decimal digits analyses when rebuilding the 634 signals from the 4/4 decimal digits analysis. Dashed lines mark the genome-wide significance threshold.

Table 1: Association statistics of 15 loci that contain a suspicious signal from the identification step, the fully conditioned model used for fine-mapping and the original GWAS meta-analysis (unconditioned). For each locus, statistics for all signal index variants are displayed. All columns described as '4/4'' refer to the analysis with 4 decimal digits [25] and all columns with '6/7' to the analysis with 6 and 7 decimal digits for effect and standard error (SE). The "N" columns provide the original sample size for the unconditioned statistics and the first signal index SNP per locus in the identification step. For the other signal index variants in the identification step and for all variants in the fully conditioned model, the presented N is N_{eff} which is calculated by GCTA. Marked in red are the p-values (identifying or fully conditioned) which are below $5x10^{-5}$ which defines a suspicious signal.

Locus signal	Signal index SNP	P-value 4/4	P-value 6/7	Effect 4/4	Effect 6/7	SE 4/4	SE 6/7	$N_{(eff)}$ low	$N_{(eff)}$ high
Identification									
n10.1	rs10142359	7.93700e-15	7.93700e-15	0.00210000	0.00210500	0.000300000	0.000271000	1004000	1004000
n10.2	rs17122159	7.51677e-12	2.76687e-07	-0.00205429	-0.00178419	0.000300006	0.000347208	1453910	1017280
n10.3	rs72719670	1.18495e-08	2.32495e-05	-0.0017106	-0.00143524	0.000300006	0.000339207	1274400	934240
Fully conditioned									
n10.1	rs10142359	0.000243772	1.51623e-05	0.00117247	0.00110064	0.000300008	0.000271008	883386	1014550
n10.2	rs17122159	1.27597e-12	6.08087e-08	-0.00212911	-0.00188062	0.000300006	0.000347208	1453910	1017280
n10.3	rs72719670	1.18495e-08	2.32495e-05	-0.0017106	-0.00143524	0.000300006	0.000339207	1274400	934240
unconditioned									
n10.1	rs10142359	7.937e-15	7.937e-15	0.0021	0.002105	3e-04	0.0003472	1004000	1004000
n10.2	rs17122159	1.237e-11	1.237e-11	-0.0024	-0.002353	3e-04	0.0002710	1004040	1004040
n10.3	rs72719670	2.072e-10	2.072e-10	-0.0022	-0.002156	3e-04	0.0003392	948348	948348
Identification									
k3.1	rs28817415	4.31000e-189	4.31000e-189	-0.00800000	-0.007951000	0.000300000	0.000271100	1003980	1003980
k3.2	rs6839100	1.09422e-24	6.59319e-08	0.00307783	0.001757040	0.000300055	0.000325263	1246410	994064
k3.3	rs6839401	8.64070e-11	0.132779	-0.00194681	-0.000468887	0.000300016	0.000311919	1168190	1012840
Fully conditioned									
k3.1	rs28817415	6.31686e-15	2.87791e-18	-0.00234017	-0.00236394	0.000300119	0.000271214	891758	1023370
k3.2	rs6839100	1.12581e-26	6.56692e-05	0.00320778	0.00129827	0.000300055	0.000325263	1246410	994064
k3.3	rs6839401	8.6407e-11	0.132779	-0.00194681	-0.000468887	0.000300016	0.000311919	1168190	1012840
unconditioned									
k3.1	rs28817415	4.310e-189	4.310e-189	-0.0080	-0.007951	3e-04	0.0002711	1003980	1003980
k3.2	rs6839100	1.774e-85	1.774e-85	0.0064	0.006372	3e-04	0.0003252	961734	961734
k3.3	rs6839401	2.669e-28	2.669e-28	-0.0034	-0.003441	3e-04	0.0003119	1004010	1004010

Locus signal	Signal index SNP	P-value 4/4	P-value 6/7	Effect 4/4	Effect 6/7	SE 4/4	SE 6/7	$N_{(eff)}$ low	$N_{(eff)}$ high
Identification									
k1.1	rs1145084	1.52000e-232	1.52000e-232	-0.00910000	-0.00906800	0.000300000	0.000278500	1004040	1004040
k1.2	rs68087211	6.41580e-33	1.81967e-24	-0.00478199	-0.00410010	0.000400134	0.000401642	1049880	976545
k1.3	rs117332547	8.90278e-14	6.18698e-05	-0.00298278	-0.00169893	0.000400038	0.000424142	1177850	981947
k1.4	rs28877685	9.59149e-09	9.08744e-07	-0.00172135	-0.0014319	0.000300003	0.000291603	1028610	1020320
Fully conditioned									
k1.1	rs1145084	4.08701e-15	3.34139e-20	-0.00235682	-0.00256559	0.000300147	0.000278644	939474	1021580
k1.2	rs68087211	1.13136e-38	4.26728e-23	-0.00520413	-0.00397526	0.000400134	0.000401642	1049880	976545
k1.3	rs117332547	7.54135e-16	4.63583e-06	-0.00322487	-0.00194284	0.000400038	0.000424142	1177850	981947
k1.4	rs28877685	9.59149e-09	9.08744e-07	-0.00172135	-0.0014319	0.000300003	0.000291603	1028610	1020320
unconditioned									
k1.1	rs1145084	1.520e-232	1.520e-232	-0.0091	-0.009068	3e-04	0.0002785	1004040	1004040
k1.2	rs68087211	5.800e-153	5.800e-153	-0.0106	-0.010579	4e-04	0.0004015	961734	961734
k1.3	rs117332547	8.460e-45	8.460e-45	-0.0060	-0.005956	4e-04	0.0004241	961734	961734
k1.4	rs28877685	4.032e-06	4.032e-06	0.0013	0.001344	3e-04	0.0002916	1004040	1004040

Identification									
k6.1	rs1047891	2.29000e-121	2.2900e-121	-0.00700000	-0.006952000	0.000300000	0.000296800	960783	960783
k6.2	rs72944180	3.90712e-11	0.115730	-0.00198256	-0.000543224	0.000300044	0.000345353	1413710	1000050
Fully conditioned									
k6.1	rs1047891	6.08525e-30	5.79162e-30	-0.00341109	-0.00337603	0.000300079	0.000296883	1025750	982122
k6.2	rs72944180	3.90712e-11	0.115730	-0.00198256	-0.000543224	0.000300044	0.000345353	1413710	1000050
unconditioned									
k6.1	rs1047891	2.290e-121	2.290e-121	-0.0070	-0.006952	3e-04	0.0002968	960783	960783
k6.2	rs72944180	2.346e-69	2.346e-69	-0.0061	-0.006078	3e-04	0.0003453	1003870	1003870

Locus signal	Signal index SNP	P-value 4/4	P-value 6/7	Effect 4/4	Effect 6/7	SE 4/4	SE 6/7	$N_{(eff)}$ low	$N_{(eff)}$ high
Identification									
k11.1	rs2279463	2.46000e-94	2.46000e-94	0.00820000	0.008195000	0.000400000	0.000397700	998409	998409
k11.2	rs316010	3.40750e-67	2.08439e-51	-0.00692846	-0.006717690	0.000400060	0.000445371	1303060	985346
k11.3	rs3798167	1.89325e-37	4.24561e-32	-0.00383675	-0.004014430	0.000300008	0.000340410	1354030	985623
k11.4	rs1867351	1.23776e-35	1.36358e-29	0.00373801	0.003777680	0.000300007	0.000334408	1351600	1019480
k11.5	rs12523781	2.16860e-13	7.80771e-12	0.00293520	0.002950980	0.000400002	0.000431302	1017670	820327
k11.6	rs9347368	6.65781e-14	3.80181e-14	0.00374730	0.003659770	0.000500010	0.000483610	1014960	1016800
k11.7	rs141805598	1.04853e-12	1.87952e-11	0.01211090	0.011652800	0.001700020	0.001735320	961420	864732
k11.8	rs118180095	3.08765e-08	6.16642e-08	0.00996558	0.009765690	0.001800010	0.001803810	744742	695015
k11.9	rs3123629	5.29052e-09	2.39026e-09	0.00175133	0.001767360	0.000300000	0.000296100	1049680	1009820
k11.10	rs2619264	4.22548e-08	0.0178217	0.00164438	0.000767668	0.000300005	0.000324006	1253740	1007350
Fully conditioned									
k11.1	rs2279463	0.000929659	0.00203261	0.00132466	0.00122733	0.00040008	0.000397784	1054060	999269
k11.2	rs316010	1.5268e-38	1.2496e-13	-0.00519399	-0.00330083	0.00040006	0.000445371	1303060	985346
k11.3	rs3798167	1.56674e-49	3.15401e-39	-0.00443876	-0.00446048	0.000300008	0.00034041	1354030	985623
k11.4	rs1867351	1.63979e-34	9.94306e-28	0.00367567	0.00364954	0.000300007	0.000334408	1351600	1019480
k11.5	rs12523781	2.11623e-18	8.29656e-16	0.00350038	0.00347187	0.000400002	0.000431302	1017670	820327
k11.6	rs9347368	5.84405e-11	3.28743e-11	0.0032739	0.00320783	0.00050001	0.00048361	1014960	1016800
k11.7	rs141805598	6.60411e-12	1.4383e-10	0.0116723	0.0111265	0.00170002	0.00173532	961420	864732
k11.8	rs118180095	2.81941e-09	2.97173e-09	0.0106953	0.0107023	0.00180001	0.00180381	744742	695015
k11.9	rs3123629	1.88343e-10	1.71774e-10	0.00191117	0.0018905	3e-04	0.0002961	1049680	1009820
k11.10	rs2619264	4.22548e-08	0.0178217	0.00164438	0.000767668	0.000300005	0.000324006	1253740	1007350
unconditioned									
k11.1	rs2279463	2.460e-94	2.460e-94	0.0082	0.008195	4e-04	0.0003977	998409	998409
k11.2	rs316010	9.956e-71	9.956e-71	-0.0079	-0.007918	4e-04	0.0004453	1004000	1004000
k11.3	rs3798167	1.852e-14	1.852e-14	-0.0026	-0.002608	3e-04	0.0003404	998409	998409
k11.4	rs1867351	1.860e-12	1.860e-12	0.0024	0.002356	3e-04	0.0003344	1004020	1004020
k11.5	rs12523781	1.993e-03	1.993e-03	0.0013	0.001333	4e-04	0.0004313	664336	664336
k11.6	rs9347368	4.030e-11	4.030e-11	0.0032	0.003193	0.0005	0.0004836	1003960	1003960
k11.7	rs141805598	8.965e-07	8.965e-07	0.0085	0.008526	0.0017	0.0017353	935382	935382

k11.8	rs118180095	4.218e-04	4.218e-04	0.0064	0.006360	0.0018	0.0018038	851305	851305
k11.9	rs3123629	5.993e-02	5.993e-02	-0.0006	-0.000557	0.0003	0.0002961	995766	995766
k11.10	rs2619264	9.586e-10	9.586e-10	0.0020	0.001982	0.0003	0.0003240	1004020	1004020

Locus signal	**Signal index SNP**	**P-value 4/4**	**P-value 6/7**	**Effect 4/4**	**Effect 6/7**	**SE 4/4**	**SE 6/7**	**$N_{(eff)}$ low**	**$N_{(eff)}$ high**
Identification									
k21.1	rs13065446	7.42600e-54	7.42600e-54	-0.00450000	-0.004463000	0.000300000	0.000288900	1003960	1003960
k21.2	rs9812062	1.59091e-15	6.05678e-12	0.00239098	0.002279550	0.000300009	0.000331411	1280130	983130
k21.3	rs7630837	2.09608e-09	0.0361724	-0.00179721	-0.000723466	0.000300026	0.000345332	1447750	1024140
k21.4	rs57345461	1.02246e-08	2.50132e-07	0.00229081	0.002319920	0.000400005	0.000449806	1368320	1014110
Fully conditioned									
k21.1	rs13065446	4.46154e-10	1.03931e-10	-0.0018713	-0.00186684	0.000300034	0.000288934	1002250	1012840
k21.2	rs9812062	2.37862e-09	6.95327e-09	0.00179093	0.00191955	0.000300009	0.000331411	1280130	983130
k21.3	rs7630837	2.26183e-09	0.0365632	-0.0017935	-0.000721954	0.000300026	0.000345332	1447750	1024140
k21.4	rs57345461	1.02246e-08	2.50132e-07	0.00229081	0.002319920	0.000400005	0.000449806	1368320	1014110
unconditioned									
k21.1	rs13065446	7.426e-54	7.426e-54	-0.0045	-0.004463	3e-04	0.0002889	1003960	1003960
k21.2	rs9812062	7.683e-16	7.683e-16	0.0027	0.002671	3e-04	0.0003314	960783	960783
k21.3	rs7630837	1.370e-43	1.370e-43	-0.0048	-0.004780	3e-04	0.0003453	1004010	1004010
k21.4	rs57345461	2.057e-07	2.057e-07	0.0023	0.002336	4e-04	0.0004498	1003820	1003820

Identification									
k30.1	rs2076668	6.91000e-43	6.91000e-43	-0.00390000	-0.00385300	0.000300000	0.000280700	961246	961246
k30.2	rs639763	7.86632e-09	3.01915e-05	-0.00173143	-0.00141143	0.000300007	0.000338308	1327390	978263
Fully conditioned									
k30.1	rs2076668	2.99259e-31	7.35278e-35	-0.00348852	-0.00345766	0.000300027	0.000280726	932151	997840
k30.2	rs639763	7.86632e-09	3.01915e-05	-0.00173143	-0.00141143	0.000300007	0.000338308	1327390	978263
unconditioned									
k30.1	rs2076668	6.910e-43	6.910e-43	-0.0039	-0.003853	3e-04	0.0002807	961246	961246
k30.2	rs639763	3.612e-12	3.612e-12	-0.0024	-0.002352	3e-04	0.0003383	961246	961246

Locus signal	Signal index SNP	P-value 4/4	P-value 6/7	Effect 4/4	Effect 6/7	SE 4/4	SE 6/7	$N_{(eff)}$ low	$N_{(eff)}$ high
Identification									
k31.1	rs6127099	3.83200e-52	3.83200e-52	-0.00480000	-0.00476600	0.000300000	0.000313700	960780	960780
k31.2	rs2762943	7.62608e-22	8.44447e-19	0.00480267	0.00481948	0.000500025	0.000544329	1073210	848722
k31.3	rs2585442	7.55338e-13	1.94012e-11	0.00215083	0.00215553	0.000300017	0.000321220	1183190	967301
k31.4	rs17217119	9.79880e-10	5.73272e-05	-0.00183395	-0.00136491	0.000300025	0.000339229	1363290	999393
Fully conditioned									
k31.1	rs6127099	1.71636e-08	4.87615e-08	-0.00169172	-0.00171168	0.000300035	0.000313738	1095370	938837
k31.2	rs2762943	1.11735e-23	1.95399e-19	0.00501559	0.00490757	0.000500025	0.000544329	1073210	848722
k31.3	rs2585442	3.18516e-09	7.81746e-09	0.00177663	0.00185419	0.000300017	0.00032122	1183190	967301
k31.4	rs17217119	9.79880e-10	5.73272e-05	-0.00183395	-0.00136491	0.000300025	0.000339229	1363290	999393
unconditioned									
k31.1	rs6127099	3.832e-52	3.832e-52	-0.0048	-0.004766	3e-04	0.0003137	960780	960780
k31.2	rs2762943	9.694e-22	9.694e-22	0.0052	0.005214	3e-04	0.0005443	873788	873788
k31.3	rs2585442	2.227e-28	2.227e-28	0.0035	0.003548	3e-04	0.0003212	960783	960783
k31.4	rs17217119	1.993e-39	1.993e-39	-0.0045	-0.004457	3e-04	0.0003392	993708	993708

Identification									
k59.1	rs223308	4.24000e-28	4.24000e-28	-0.00300000	-0.00297000	0.000300000	0.000270300	1004040	1004040
k59.2	rs223389	2.35370e-10	3.10247e-04	0.00190096	0.00122665	0.000300011	0.000340114	1391750	1014870
Fully conditioned									
k59.1	rs223308	4.3714e-09	1.24688e-10	-0.00176095	-0.00173908	0.000300017	0.000270316	883775	1020260
k59.2	rs223389	2.35370e-10	3.10247e-04	0.00190096	0.00122665	0.000300011	0.000340114	1391750	1014870
unconditioned									
k59.1	rs223308	4.240e-28	4.240e-28	-0.0030	-0.002970	3e-04	0.0002703	1004040	1004040
k59.2	rs223389	8.907e-20	8.907e-20	0.0031	0.003096	3e-04	0.0003401	1004020	1004020

Locus signal	Signal index SNP	P-value 4/4	P-value 6/7	Effect 4/4	Effect 6/7	SE 4/4	SE 6/7	$N_{(eff)}$ low	$N_{(eff)}$ high
Identification									
k37.1	rs187355703	6.66800e-40	6.66800e-40	0.01170000	0.01165100	0.000900000	0.000881300	958688	958688
k37.2	rs35440573	5.58146e-26	1.49962e-28	-0.00316253	-0.00315693	0.000300018	0.000284818	956334	994465
k37.3	rs863678	4.26218e-21	7.02439e-22	-0.00282795	-0.00280823	0.000300018	0.000292118	959363	948376
k37.4	rs16863775	1.09346e-09	3.91049e-05	-0.00182860	-0.00143086	0.000300009	0.000347911	1426980	994422
k37.5	rs12472159	3.63722e-08	1.43451e-07	0.00165233	0.0015667	0.000300008	0.000297808	1031120	980669
k37.6	rs1992909	1.79996e-08	8.92917e-09	-0.00168907	-0.00162148	0.000300001	0.000282001	917034	972635
Fully conditioned									
k37.1	rs187355703	3.75931e-25	3.62787e-26	0.00932509	0.00932645	0.000900083	0.000881386	913476	892792
k37.2	rs35440573	9.7991e-23	3.38293e-25	-0.00294438	-0.00295366	0.000300018	0.000284818	956334	994465
k37.3	rs863678	5.5929e-20	6.77244e-21	-0.00274574	-0.00273926	0.000300018	0.000292118	959363	948376
k37.4	rs16863775	1.25143e-10	5.30982e-06	-0.00192995	-0.00143086	0.000300009	0.000347911	1426980	994422
k37.5	rs12472159	4.77843e-09	1.7189e-08	0.00175646	0.00167908	0.000300008	0.000297808	1031120	980669
k37.6	rs1992909	1.79996e-08	8.92917e-09	-0.00168907	-0.00162148	0.000300001	0.000282001	917034	972635
unconditioned									
k37.1	rs187355703	6.668e-40	6.668e-40	0.0117	0.011651	9e-04	0.0008813	958688	958688
k37.2	rs35440573	3.719e-29	3.719e-29	-0.0032	-0.003192	3e-04	0.0002848	961734	961734
k37.3	rs863678	1.672e-27	1.672e-27	-0.0032	-0.003174	3e-04	0.0002921	997455	997455
k37.4	rs16863775	8.499e-16	8.499e-16	-0.0028	-0.002800	3e-04	0.0003479	961734	961734
k37.5	rs12472159	5.081e-13	5.081e-13	0.0022	0.002151	3e-04	0.0002978	961734	961734
k37.6	rs1992909	1.832e-03	1.832e-03	-0.0009	-0.000879	3e-04	0.0002820	960783	960783

Locus signal	Signal index SNP	P-value 4/4	P-value 6/7	Effect 4/4	Effect 6/7	SE 4/4	SE 6/7	$N_{(eff)}$ low	$N_{(eff)}$ high
Identification									
k35.1	rs17256228	9.44500e-39	9.44500e-39	-0.00560000	-0.00563800	0.000400000	0.000433000	1003020	1003020
k35.2	rs78660602	7.42551e-25	1.22421e-27	-0.00514781	-0.00518942	0.000500032	0.000476333	938119	968841
k35.3	rs7707989	5.25552e-11	1.12493e-08	0.00196908	0.00197594	0.000300004	0.000346005	1372570	967041
k35.4	rs1168410	3.68605e-09	1.17366e-08	0.00176935	0.00173162	0.000300007	0.000303607	888737	813265
k35.5	rs35239	3.02662e-08	1.01643e-05	-0.00166200	-0.00150332	0.000300005	0.000340606	1356020	813265
Fully conditioned									
k35.1	rs17256228	1.55607e-41	2.09666e-35	-0.00540056	-0.00537728	0.000400030	0.000433036	1284190	1027050
k35.2	rs78660602	2.15833e-20	8.03456e-23	-0.00462740	-0.00468427	0.000500032	0.000476333	938119	968841
k35.3	rs7707989	8.06919e-11	1.51153e-08	0.00194982	0.00195847	0.000300004	0.000346005	1372570	967041
k35.4	rs1168410	6.75673e-10	2.77341e-09	0.00185155	0.00180478	0.000300007	0.000303607	888737	813265
k35.5	rs35239	3.02662e-08	1.01643e-05	-0.00166200	-0.00150332	0.000300005	0.000340606	1356020	813265
unconditioned									
k35.1	rs17256228	9.445e-39	9.445e-39	-0.0056	-0.005638	4e-04	0.0004330	1003020	1003020
k35.2	rs78660602	2.568e-31	2.568e-31	-0.0055	-0.005544	5e-04	0.0004763	1004040	1004040
k35.3	rs7707989	1.722e-07	1.722e-07	0.0018	0.001808	3e-04	0.0003460	961734	961734
k35.4	rs1168410	8.414e-10	8.414e-10	0.0019	0.001863	3e-04	0.0003036	823439	823439
k35.5	rs35239	1.008e-09	1.008e-09	-0.0021	-0.002080	3e-04	0.0003406	994815	994815

Locus signal	Signal index SNP	P-value 4/4	P-value 6/7	Effect 4/4	Effect 6/7	SE 4/4	SE 6/7	$N_{(eff)}$ low	$N_{(eff)}$ high
Identification									
k84.1	rs3904600	3.96400e-26	3.96400e-26	0.00300000	0.00299900	0.000300000	0.000283600	961734	961734
k84.2	rs13192947	3.33867e-09	4.45845e-09	0.00177422	0.00177044	0.000300003	0.000301803	1077980	998242
k84.3	rs1413700	2.18601e-08	0.000152203	0.00167906	0.0012109	0.000300014	0.000319715	1234390	1018660
Fully conditioned									
k84.1	rs3904600	3.75028e-06	9.05762e-07	0.0013875	0.00182634	0.000300016	0.000283616	946010	992072
k84.2	rs13192947	5.79785e-10	1.43559e-09	0.00185877	0.0012109	0.000300003	0.000301803	1077980	998242
k84.3	rs1413700	2.18601e-08	0.000152203	0.00167906	0.0012109	0.000300014	0.000319715	1234390	1018660
unconditioned									
k84.1	rs3904600	3.964e-26	3.964e-26	0.0030	0.002999	3e-04	0.0002836	961734	961734
k84.2	rs13192947	1.316e-06	1.316e-06	0.0015	0.001460	3e-04	0.0003018	961734	961734
k84.3	rs1413700	4.048e-23	4.048e-23	0.0032	0.003166	3e-04	0.0003197	1003860	1003860

Identification									
k85.1	rs11071738	5.9680e-21	5.9680e-21	-0.00260000	-0.00255100	0.000300000	0.000271700	1003700	1003700
k85.2	rs2652815	1.06035e-08	2.67970e-04	-0.00171628	-0.00124716	0.000300007	0.000342209	1384940	997546
Fully conditioned									
k85.1	rs11071738	7.53667e-10	1.96921e-11	-0.00184639	-0.00182272	0.000300013	0.000271712	885365	1011590
k85.2	rs2652815	1.06035e-08	2.67970e-04	-0.00171628	-0.00124716	0.000300007	0.000342209	1384940	997546
unconditioned									
k85.1	rs11071738	5.968e-21	5.968e-21	-0.0026	-0.002551	3e-04	0.0002717	1003700	1003700
k85.2	rs2652815	8.003e-13	8.003e-13	-0.0025	-0.002450	3e-04	0.0003422	1003650	1003650

Locus signal	Signal index SNP	P-value 4/4	P-value 6/7	Effect 4/4	Effect 6/7	SE 4/4	SE 6/7	$N_{(eff)}$ low	$N_{(eff)}$ high
Identification									
k107.1	rs6968865	2.86600e-20	2.86600e-20	-0.00260000	-0.00257700	0.000300000	0.000279400	999266	999266
k107.2	rs58221005	3.84011e-09	1.32891e-05	-0.00176731	-0.00151436	0.000300006	0.000347707	1405810	980799
Fully conditioned									
k107.1	rs6968865	9.62641e-14	1.32392e-15	-0.00223387	-0.00223315	0.000300012	0.000279411	945512	1021590
k107.2	rs58221005	3.84011e-09	1.32891e-05	-0.00176731	-0.00151436	0.000300006	0.000347707	1405810	980799
unconditioned									
k107.1	rs6968865	2.866e-20	2.866e-20	-0.0026	-0.002577	3e-04	0.0002794	999266	999266
k107.2	rs58221005	6.069e-10	6.069e-10	-0.0022	-0.002152	3e-04	0.0003477	961734	961734

Identification									
k154.1	rs6433958	2.52100e-12	2.52100e-12	-0.00190000	-0.001888	0.000300000	0.000269700	1001490	1001490
k154.2	rs62188290	9.77156e-09	1.42798e-05	0.00172042	0.001508	0.000300005	0.000347505	1411310	985772
Fully conditioned									
k154.1	rs6433958	8.68811e-08	2.30133e-09	-0.0016057	-0.00161149	0.000300007	0.000269706	883077	1024000
k154.2	rs62188290	9.77156e-09	1.42798e-05	0.00172042	0.001508	0.000300005	0.000347505	1411310	985772
unconditioned									
k154.1	rs6433958	2.521e-12	2.521e-12	-0.0019	-0.001888	3e-04	0.0002697	1001490	1001490
k154.2	rs62188290	1.937e-08	1.937e-08	0.0020	0.001952	3e-04	0.0003475	958796	958796

3.1.1.2. Stability of credible set sizes

99% credible sets describe a set of variants that with a probability of 99% contain the SNP that is causal for the association signal. Small sets with five or less variants are of particular interest for further investigation as this is a manageable number for laboratory analyses. Thus, it is important to understand how the number of decimal digits of GWAS data influences the credible set size. As shown in **Figure 4** the two analyses with more decimal digits tend to have fewer signals with single and small credible sets, but more signals with large credible set size (>50).

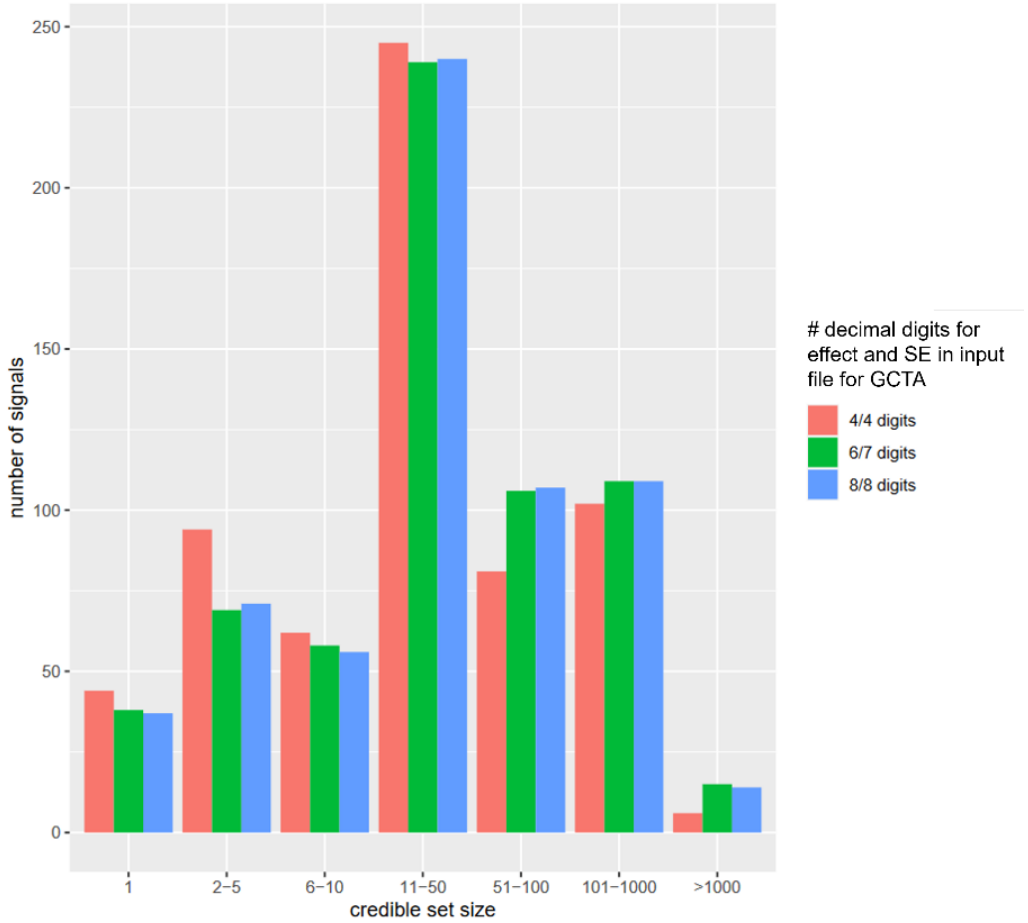


Figure 4: Comparison of the number of signals with a specific 99% credible set size between the three analyses with different number of decimal digits in the input files for fine-mapping.

Of more interest than the general development of credible set sizes is the development of the individual sets, e.g. which originally small sets from the 4/4 decimal digits analysis stay small in the 6/7 or 8/8 decimal digits analyses. Of the138 signals with small credible sets described by Stanzick et al. 2021 [25], 93 are still small in the 6/7 and 8/8 decimal digits analyses and one signal is still small in the 8/8 decimal digits

analysis (5 credible set variants) but not in the 6/7 digits analysis (6 credible set variants) (**Figure 5A**). 3 of the 45 originally small sets which are not small in the analyses with more decimal digits even reach credible set sizes above 1000 (**Figure 6** small panel). The signal with nearly 4000 credible set variants in the higher precision analyses is signal k3.3 which was above stated as a suspicious signal. From the originally 44 signals with single credible set, 29 were reidentified as such by both analyses with more decimal digits (**Figure 5B**). The other 15 became larger signals with 8 signals even having credible set sizes above 5 (**Table 2)**. Beside the reidentified small signals, 14 signals were newly identified as small in the analyses with more decimal digits including 8 single credible sets in both analyses and one single credible set, which is only a single credible set in the 6/7 decimal digits analysis. Overall, the credible set size is more stable between the two analyses with more decimal digits than between the original analysis with 4 decimal digits and the 6/7 decimal digits analysis (**Figure 7**). Nevertheless, two signals would be stated as small or single credible set in only one of the two analyses with more decimal digits.

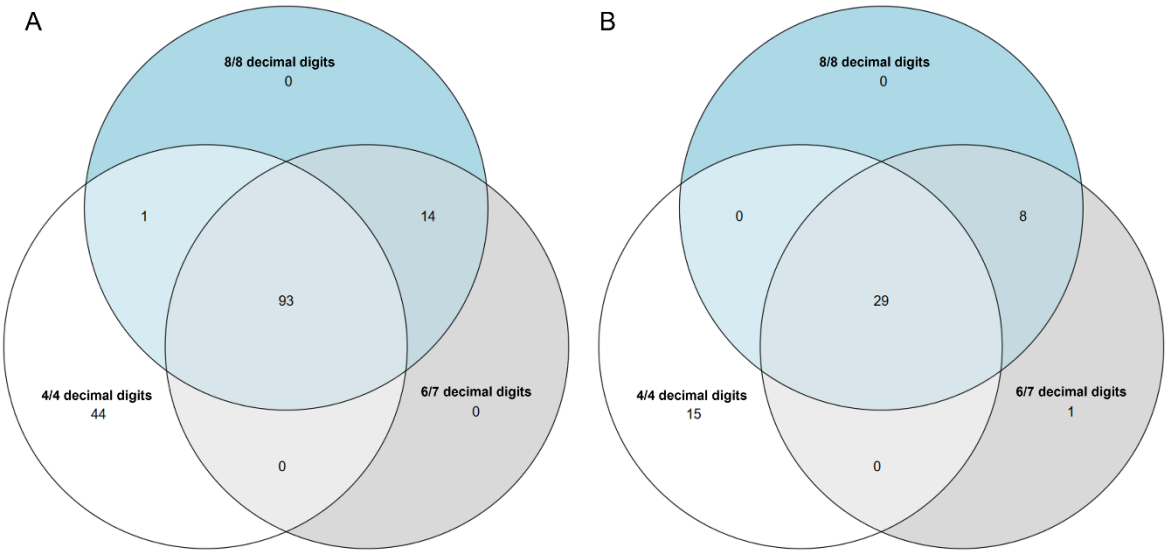


Figure 5: Overlap of signals with small (≤5 variants, A) and single credible sets (B) between the three fine-mapping analyses with different numbers of decimal digits.

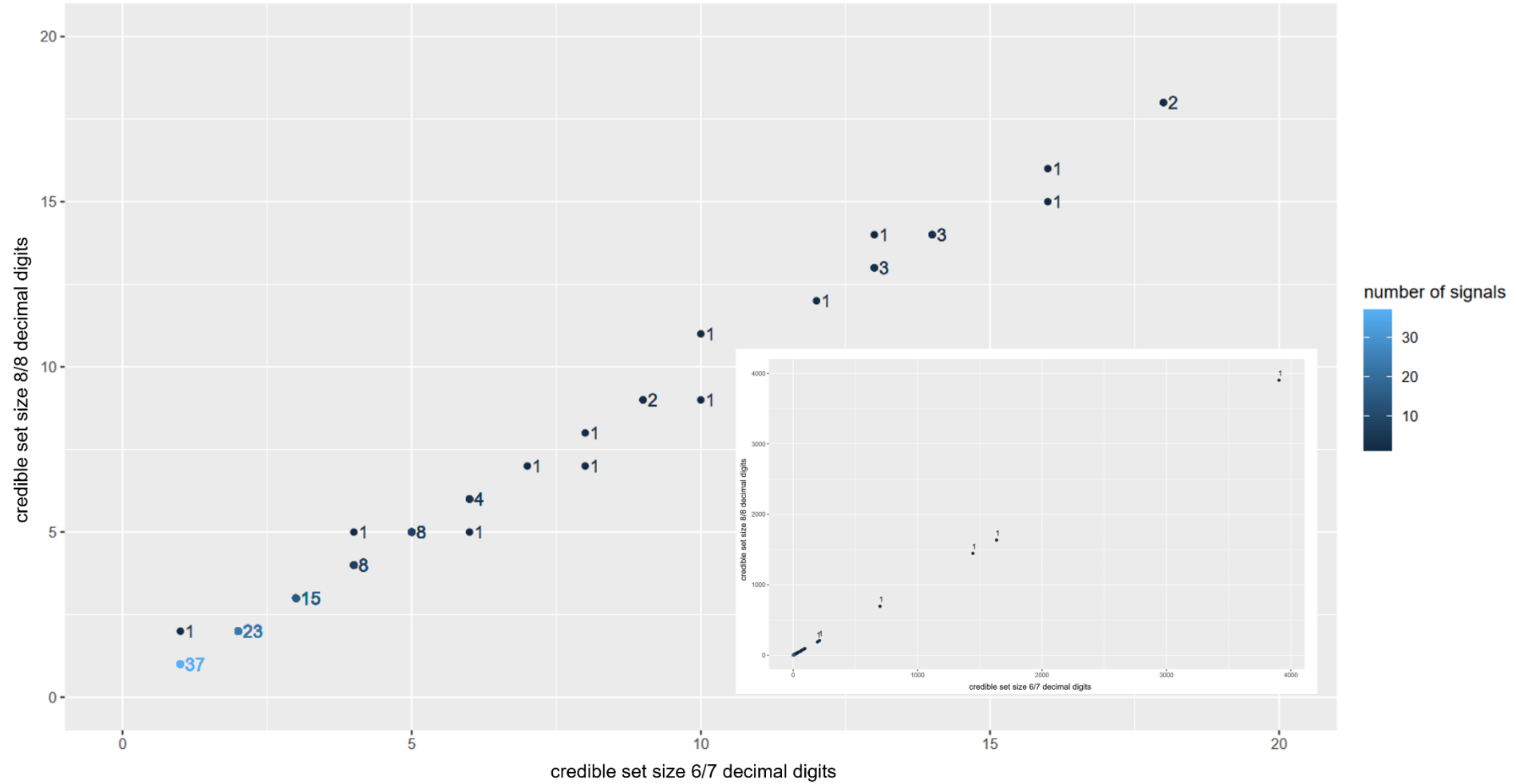


Figure 6: Credible set sizes of the 138 originally small credible sets from the 4/4 decimal digits analysis [25] in the 6/7 decimal digits analysis compared to the set size from the 8/8 decimal digits analysis. Numbers indicate the number of signals with this specific set size. The snippet in the right corner shows the overall picture, while the main graph is reduced to credible set sizes below 20. The numbers and color indicate how many signals with the respective credible set sizes are represented by one dot.

Table 2: Development of credible set size from 4/4 decimal digits [25] to 6/7 decimal digits for all 634 signals. The 634 signals were grouped by their credible set size in the 4/4 and 6/7 decimal digits analysis. The table shows how many signals belong to the same and how many to different credible set size groups. Greenish colour indicates a same or smaller set size in the 6/7 decimal digits analysis while reddish colour indicates a larger set size in the 6/7 decimal digits analysis.

		Credible set size 6/7 decimal digits						
		1	**2-5**	**6-10**	**11-50**	**51-100**	**101-1000**	**>1000**
Credible set size 4/4 decimal digits	**1**	**29**	7	2	5	1	0	0
	2-5	8	**50**	8	18	4	3	3
	6-10	0	5	**32**	21	3	1	0
	11-50	0	8	14	**178**	29	12	4
	51-100	0	1	0	11	**48**	19	2
	101-1000	0	0	0	7	22	**72**	1
	>1000	0	0	0	0	0	2	**4**

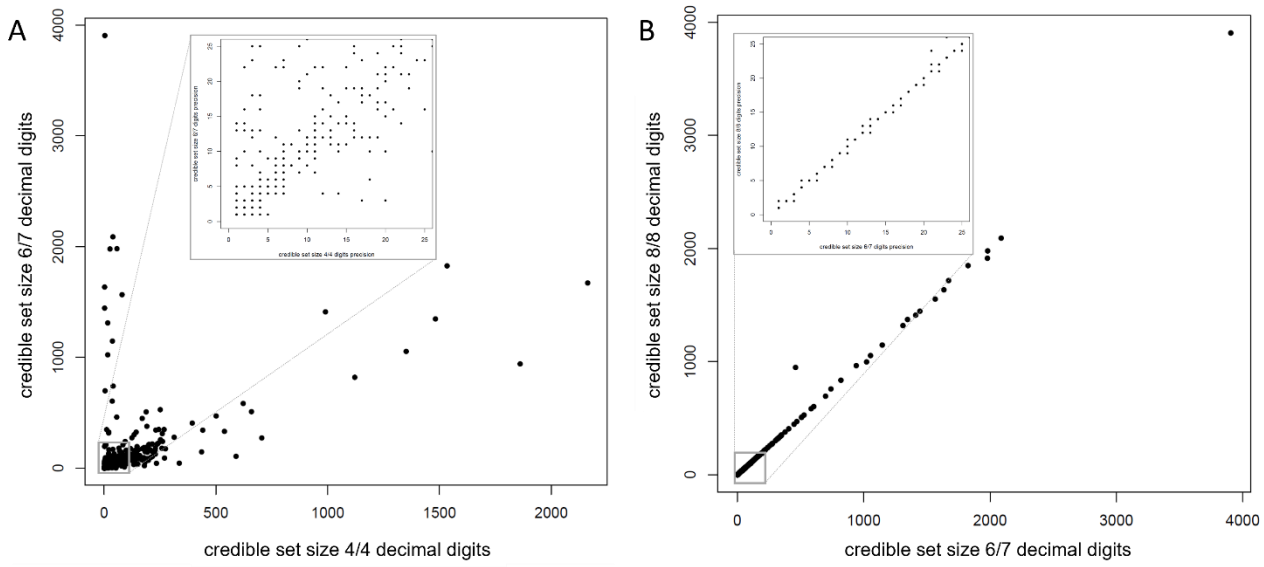


Figure 7: Modified from Stanzick et al. 2023 Supplementary Figure 2 panel A+B [43]: Comparison of credible set sizes for all 634 signals. A: Comparison of the credible set sizes of the 4/4 decimal digits analysis with the 6/7 decimal digits analysis. B: Comparison of the credible set sizes of the 6/7 decimal digits analysis with the 8/8 decimal digits analysis. Small panels are focused on signals with 25 or less credible set variants.

3.1.1.3. Stability of individual SNP PPA

Beside the development of the credible set size, it is interesting if the credible sets still consist of the same SNPs and what happens with the posterior probability of association (PPA) of SNPs which were described to be the causal variant with over 99% PPA by Stanzick et al. 2021 [25]. In total 38,306 SNPs were described by Stanzick et al. 2021 as credible set variants of the 634 signals (based on 4/4 decimal digits analyses) [25]. In the 6/7 decimal digits analysis 58,915 SNPs are included in these 634 credible sets and 58,916 SNPs in the 8/8 decimal digits analysis. In total these three analyses include 68,042 unique SNPs. Of these, 29,332 are shared between all three analyses and 29,319 SNPs are shared between the two analyses with more decimal digits (**Figure 8A**). To compare the individual PPA of a SNP it must be mapped to the same signal. In total, 76,688 unique SNP-signal combinations are included in the 3 analyses with a similar overlap than the individual credible set variant (**Figure 8B**). For the sake of simplicity, only the 44,355 credible set variant-signal pairs from the 4/4 decimal digits analysis are analysed in the following and values from the higher precision analyses were mapped to these by SNP and signal, even if the SNP is not a credible set variant (for this signal) in the higher precision analyses.

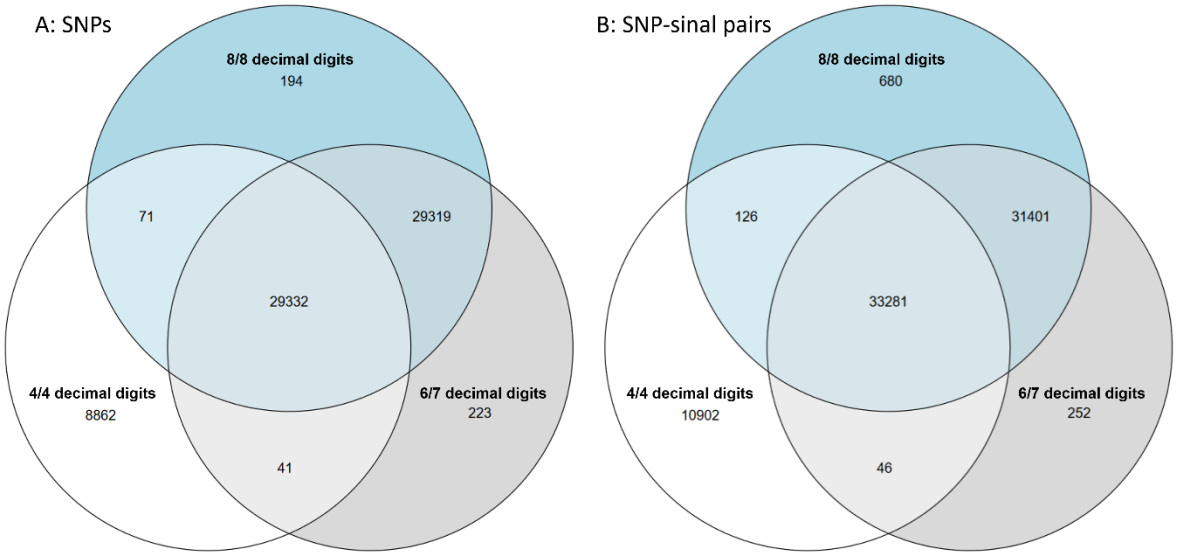


Figure 8: Overlap between the unique credible set variants and credible set variant-signal pairs that are credible set variants (-signal pairs) in at least one of the three analyses. A: Overlap of credible set variants (n=68,042) between the three analyses compared by RSID. B: Overlap of credible set variant-signal pairs (n=76,688) between the three analyses compared by RSID + signal assignment.

The individual SNP PPA varies a lot between the 4/4 and the 6/7 decimal digits analysis (**Figure 9A**) while it is nearly stable between the 6/7 and the 8/8 digits analysis (**Figure 9B**). Of the original 44 single credible set SNPs that per definition have a PPA >99%,

29 SNPs still have a PPA>99% in the analyses with more decimal digits, 7 SNPs have a PPA between 50% and 99% and 8 SNPs have a PPA below 50%. 5 of these 8 SNPs with PPA below 50% are part of a larger credible set, while 3 SNPs are not credible set variants for the respective signal in the 6/7 decimal digits analysis (**Table 3**).

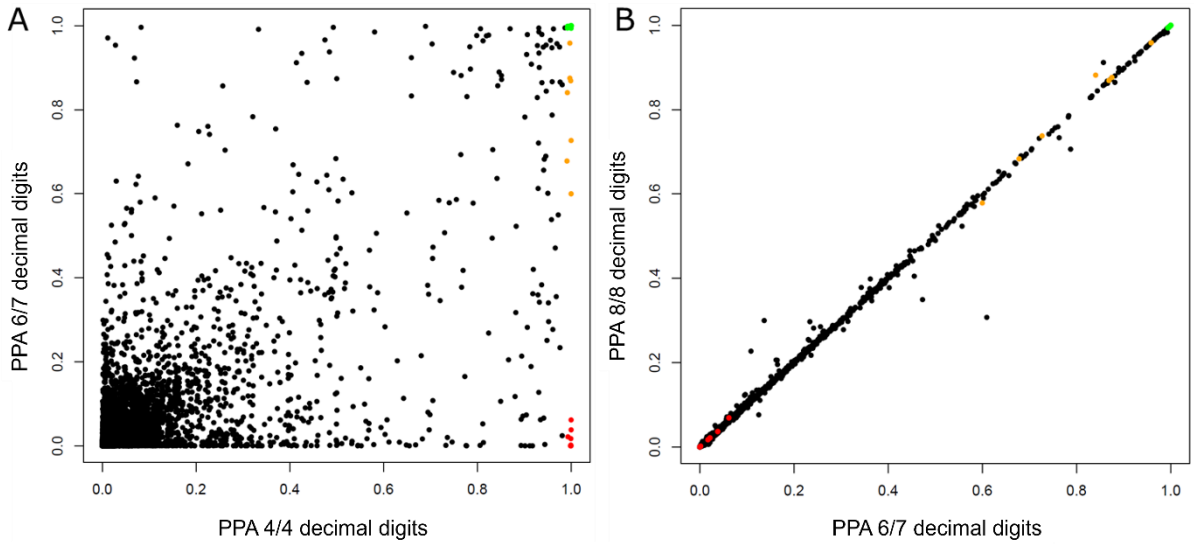


Figure 9: PPA comparison of 44,355 credible set variant-signal pairs identified in the 4/4 decimal digits analysis [25]. A: Comparison between fine-mapping results (PPA) from the 4/4 [25] and the 6/7 decimal digits analysis for 44,355 credible set variant-signal pairs. All coloured dots highlight SNPs with PPA>99% in the 4/4 decimal digits analysis (n=44). Green dots reflect SNPs with PPA>99% (n=29), orange dots SNPs with >50% but <99% (N=7) PPA and red dots SNPs with PPA <50% (n=8) in the 6/7 precision analysis. B: Comparison between PPA from the 6/ and the 8/8 decimal digits analysis for 44,355 credible set variant-signal pairs identified in the 4/4 decimal digits analysis. Colour coding is identical to panel A.

Table 3: Characteristics of 8 credible set variants with a PPA>99% (single credible sets) in the 4/4 decimal digits analysis [25] and PPA <50% in the 6/7 decimal digits analysis. Beta and SE are from the fully conditioned model when other signals in the respective locus exist. The N is the estimated effective sample size from GCTA. The last column shows the credible set size of the respective signal in the 6/7 decimal digits analysis and whether the variant, which had a PPA > 99% in the 4/4 decimal digits analysis is included in this set. The last three SNPs are not credible set variants for the respective signal in the 6/7 decimal digits analysis.

RSID	***Signal***	***# Signals in locus***	***Beta 4/4***	***Beta 6/7***	***SE 4/4***	***SE 6/7***	***N 4/4***	***N 6/7***	***PPA 4/4***	***PPA 6/7***	***Set size 6/7 (included in set)***
rs68087211	k1.2	4	-0.00520413	-0.00397526	0.000400134	0.000401642	1049880	976545	0.9999833	0.06163245	58 (yes)
rs114353587	k66.2	2	-0.00577307	-0.00572006	0.000600025	0.000641728	1111020	910286	0.9931253	0.02148356	4 (yes)
rs3813498	k87.1	1	-0.00310000	-0.00308300	0.000300000	0.000345900	1003090	1003090	0.9993512	0.001657001	42 (yes)
rs6128082	k97.1	2	0.00288973	0.00278634	0.000300011	0.000349514	1442590	996111	0.9999998	0.03796609	9 (yes)
rs2135881	n13.1	1	-0.00230000	-0.00225200	0.000300000	0.000348500	989406	989406	0.9994489	0.01711196	5 (yes)
rs6120738	k30.1	2	0.00366199	0.00363675	0.000300026	0.000320730	951322	780169	0.9985842	1.755114e-06	40 (no)
rs11650989	k7.1	4	0.00372836	0.00331350	0.000300064	0.000339777	1270660	928735	0.9997912	1.052775e-15	2 (no)
rs10994720	k40.1	1	0.00380000	0.00384100	0.000300000	0.000338100	998409	998409	0.9994361	2.166964e-07	4 (no)

3.1.1.4. Subgroup analyses

The original 634 signals from the 4/4 decimal digits analysis [25] can be divided into different subgroups. First, there are the 424 first signals of each locus that can be genome-wide significant (N=373) or not (N=51) in the original meta-analysis in European ancestry (EUR). Second, there are 210 secondary signals (all genome-wide significant in the 4/4 decimal digits analysis) that can be divided into three groups by the identification p-value of their index SNP in the 6/7 decimal digits analysis: $p_{ident} < 5x10^{-8}$ (N=143), p_{ident} between $5x10^{-8}$ and $1.265x10^{-5}$ (N=52), and $p_{ident} > 1.265x10^{-5}$ (N=15). Third, one could look at the fully conditioned p-value of each index SNP in the 4/4 decimal digits analysis and divide between p_{cond} smaller than $5x10^{-8}$ (N=571), p_{cond} between $5x10^{-8}$ and $1.265x10^{-5}$ (N=59) or p_{cond} larger than $1.265x10^{-5}$ (N=4). The following describes how these subgroups behave differently regarding changes in credible set size and individual SNP PPA when increasing the number of decimal digits from 4/4 to 6/7 decimal digits in the GWAS data. Comparison to 8/8 decimal digits is not shown as credible set size and individual SNP PPA are nearly identical to 6/7 decimal digits analysis (see chapter 3.1.1.2 and 3.1.1.3).

Of the 373 first signals that are genome-wide significant in EUR, 68 signals (18.2%) belonged to a larger credible set size group in the 6/7 compared to the 4/4 decimal digits analysis. From the not genome-wide significant first signals only 9.8% (5 from 51) belong to a larger credible set size group in the new analysis (**Table 4**). However, the credible set sizes of signals with a higher index SNP p-value in EUR are on average larger so that they belong a priori to credible set size groups covering a larger spectrum of credible set sizes. All not genome-wide significant signals of the 4/4 decimal digits analysis had a credible set size above 10. Thus, fewer credible set sizes get large enough in the 6/7 decimal digits analysis that they are assigned to another group. Among the genome-wide significant signals, 37.5% of the previously single-variant credible sets and 23.0% of the small credible sets (2-5 credible set variants), belong to a larger set size group in the 6/7 decimal digits analysis.

From the secondary signals, 24.5%, 42.3% and 86.7% belonged to a larger credible set size group for signals with genome wide significant p_{ident}, p_{ident} between $5x10^{-8}$ and $1.265x10^{-5}$, and $p_{ident} > 1.265x10^{-5}$ respectively (**Table 5**). Therefore, an instable p_{ident} between the analyses with 4/4 and 6/7 decimal digits seems to be a good indicator for instable credible set sizes. When looking at the single-variant and small sets, there are

only 3 previously small sets among the signals with $p_{ident} > 1.265x10^{-5}$ which become very large in the high precision analysis. Additionally, there are more previously single-variant and small sets among the signals with p_{ident} between $5x10^{-8}$ and $1x10^{-5}$ which belong to a larger group in the 6/7 digits analysis (70%) than among the signals with genome wide significant p_{ident} (45%). When regarding the fully conditional p-value of the signal index variants from the 4/4 decimal digits analysis, 23.8% of the signals with genome-wide significant p_{cond} belong to a larger credible set size group in the analysis with 6/7 decimal digits, 6.7% of the signals with p_{cond} between $5x10^{-8}$ and $1x10^{-5}$, and 25% of the signals with p_{cond} larger than $1x10^{-5}$ (**Table 6**).

In total, the credible set sizes of the first signals are a more stable: Only 73 from 424 first signals, (17.2%) belong to a larger credible set size group in the 6/7 compared to the 4/4 decimal digits analysis, while 70 from 210 (33.3%) secondary signals changed to a larger credible set size group. However, this effect is strongly driven by the 67 secondary signals with not-genome-wide significant identifying p-value in the 6/7 decimal digits analysis.

For the individual SNP PPA no effect was seen when dividing signals in subgroups (**Supplementary Figure 1, Supplementary Figure 2, Supplementary Figure 3**).

Table 4: Development of credible set size from 4/4 [25] to 6/7 decimal digits for 424 first signals. The 424 signals were divided by the p-value of their index variant in the 4/4 decimal digits analysis in genome-wide significant (gws, N=373) and not genome-wide significant (not gws, N=51). Then the signals were grouped by their credible set size in the 4/4 and the 6/7 decimal digits analysis. The table shows how many signals belong to the same and how many to different credible set size groups. Greenish colour indicates a same or smaller set size in the 6/7 decimal digits analysis while reddish colour indicates a larger set size in the 6/7 decimal digits analysis.

			Credible set size 6/7 decimal digits						
		First signal p-value (from 4/4 decimal digits analysis)	**1**	**2-5**	**6-10**	**11-50**	**51-100**	**101-1000**	**>1000**
Credible set size 4/4 decimal digits	**1**	$< 5 \times 10^{-8}$	15	4	1	4	0	0	0
		$> 5 \times 10^{-8}$	0	0	0	0	0	0	0
	2-5	$< 5 \times 10^{-8}$	6	41	4	10	0	0	0
		$> 5 \times 10^{-8}$	0	0	0	0	0	0	0
	6-10	$< 5 \times 10^{-8}$	0	5	23	14	2	0	0
		$> 5 \times 10^{-8}$	0	0	0	0	0	0	0
	11-50	$< 5 \times 10^{-8}$	0	5	7	117	17	2	0
		$> 5 \times 10^{-8}$	0	0	2	9	2	0	0
	51-100	$< 5 \times 10^{-8}$	0	0	0	7	31	10	0
		$> 5 \times 10^{-8}$	0	0	0	1	5	2	0
	101-1000	$< 5 \times 10^{-8}$	0	0	0	4	13	30	0
		$> 5 \times 10^{-8}$	0	0	0	3	5	21	1
	>1000	$< 5 \times 10^{-8}$	0	0	0	0	0	0	1
		$> 5 \times 10^{-8}$	0	0	0	0	0	0	0

Table 5: Development of credible set size from 4/4 [25] to 6/7 decimal digits for 210 secondary signals. The 210 signals were divided by the identifying p-value of their index variant in the 6/7 decimal digits analysis in genome-wide significant (N=143), below 1.265x10^{-5} (N=52), and above 1.265 x 10^{-5} (suspicious. N=15). Then the signals were grouped by their credible set size in the 4/4 and the 6/7 decimal digits analysis. The table shows how many signals belong to the same and how many to different credible set size groups. Greenish colour indicates a same or smaller set size in the 6/7 decimal digits analysis while reddish colour indicates a larger set size in the 6/7 decimal digits analysis.

			Credible set size 6/7 decimal digits						
		Identifying p-value (from 6/7 decimal digits analysis)	**1**	**2-5**	**6-10**	**11-50**	**51-100**	**101-1000**	**>1000**
Credible set size 4/4 decimal digits	**1**	< 5 x 10^{-8}	13	3	0	0	1	0	0
		1.265 x 10^{-5} to 5 x 10^{-8}	1	0	1	1	0	0	0
		> 1.265 x 10^{-5}	0	0	0	0	0	0	0
	2-5	< 5 x 10^{-8}	1	8	4	8	1	1	0
		1.265 x 10^{-5} to 5 x 10^{-8}	1	1	0	0	3	1	1
		> 1.265 x 10^{-5}	0	0	0	0	0	1	2
	6-10	< 5 x 10^{-8}	0	0	8	6	1	0	0
		1.265 x 10^{-5} to 5 x 10^{-8}	0	0	1	1	0	1	0
		> 1.265 x 10^{-5}	0	0	0	0	0	0	0
	11-50	< 5 x 10^{-8}	0	3	3	45	6	1	0
		1.265 x 10^{-5} to 5 x 10^{-8}	0	0	1	7	4	5	1
		> 1.265 x 10^{-5}	0	0	1	0	0	4	3
	51-100	< 5 x 10^{-8}	0	1	0	3	5	3	0
		1.265 x 10^{-5} to 5 x 10^{-8}	0	0	0	0	7	3	0
		> 1.265 x 10^{-5}	0	0	0	0	0	1	2
	101-1000	< 5 x 10^{-8}	0	0	0	0	2	14	0
		1.265 x 10^{-5} to 5 x 10^{-8}	0	0	0	0	2	6	0
		> 1.265 x 10^{-5}	0	0	0	0	0	1	0
	>1000	< 5 x 10^{-8}	0	0	0	0	0	1	1
		1.265 x 10^{-5} to 5 x 10^{-8}	0	0	0	0	0	1	2
		> 1.265 x 10^{-5}	0	0	0	0	0	0	0

Table 6: Development of credible set size from 4/4 [25] to 6/7 decimal digits for all 634 signals. The 634 signals were divided by the fully conditioned p-value of their index variant in the 4/4 decimal digits analysis in genome-wide significant (gws N=571), below $1x10^{-5}$ ($1x10^{-5}$ N=59), and above $1x10^{-5}$ (susp. N=4). Then the signals were grouped by their credible set size in the 4/4 and the 6/7 decimal digits analysis. The table shows how many signals belong to the same and how many to different credible set size groups. Greenish colour indicates a same or smaller set size in the 6/7 decimal digits analysis while reddish colour indicates a larger set size in the 6/7 decimal digits analysis.

			Credible set size 6/7 decimal places						
		Fully conditioned P-value (from 4/4 decimal digits analysis)	**1**	**2-5**	**6-10**	**11-50**	**51-100**	**101-1000**	**>1000**
Credible set size 4/4 decimal places	**1**	< 5×10^{-8}	29	7	2	5	1	0	0
		1.265×10^{-5} to 5×10^{-8}	0	0	0	0	0	0	0
		> 1.265×10^{-5}	0	0	0	0	0	0	0
	2-5	< 5×10^{-8}	8	50	8	18	4	3	3
		1.265×10^{-5} to 5×10^{-8}	0	0	0	0	0	0	0
		> 1.265×10^{-5}	0	0	0	0	0	0	0
	6-10	< 5×10^{-8}	0	5	31	21	3	1	0
		1.265×10^{-5} to 5×10^{-8}	0	0	1	0	0	0	0
		> 1.265×10^{-5}	0	0	0	0	0	0	0
	11-50	< 5×10^{-8}	0	7	12	167	27	11	4
		1.265×10^{-5} to 5×10^{-8}	0	1	2	10	2	1	0
		> 1.265×10^{-5}	0	0	0	1	0	0	0
	51-100	< 5×10^{-8}	0	1	0	10	42	16	2
		1.265×10^{-5} to 5×10^{-8}	0	0	0	1	5	2	0
		> 1.265×10^{-5}	0	0	0	0	1	1	0
	101-1000	< 5×10^{-8}	0	0	0	4	14	49	0
		1.265×10^{-5} to 5×10^{-8}	0	0	0	3	8	22	1
		> 1.265×10^{-5}	0	0	0	0	0	1	0
	>1000	< 5×10^{-8}	0	0	0	0	0	2	4
		1.265×10^{-5} to 5×10^{-8}	0	0	0	0	0	0	0
		> 1.265×10^{-5}	0	0	0	0	0	0	0

3.1.2. Second signal analysis with GCTA without predefining signal index variants

3.1.2.1. Stability of signals

The original second signal analysis by Stanzick et al. 2021 identified 634 independent signals within the 424 eGFR associated loci [25]. The naïve conditioning with GCTA with the 6/7 or the 8/8 decimal digits GWAS datasets as input identified only 594 signals in the same 424 loci [43]. All but one signal index variants of those 594 signals were identical or highly correlated between the two analyses with more decimal places. The only not comparable signal is the second signal in locus k10 which has a R^2 of 0.42. As the conditional results of the 6/7 and 8/8 decimal digits analyses are, except for this one signal, totally comparable, the further comparisons have only been made between the original and the 6/7 decimal digits analysis.

Of the 210 secondary signals in the analysis with 4/4 decimal digits [25] and the 170 secondary signals in the analysis with 6/7 decimal digits [43], 61 signals had an identical signal index SNP, 35 signals were highly correlated (R^2>0.9), and 22 signals were correlated (R^2 between 0.5 and 0.9). The other 92 signals from the 4/4 decimal digits analysis had no correlated signal in the 6/7 decimal digits analysis, as well as 52 signals vice versa (**Figure 10**).

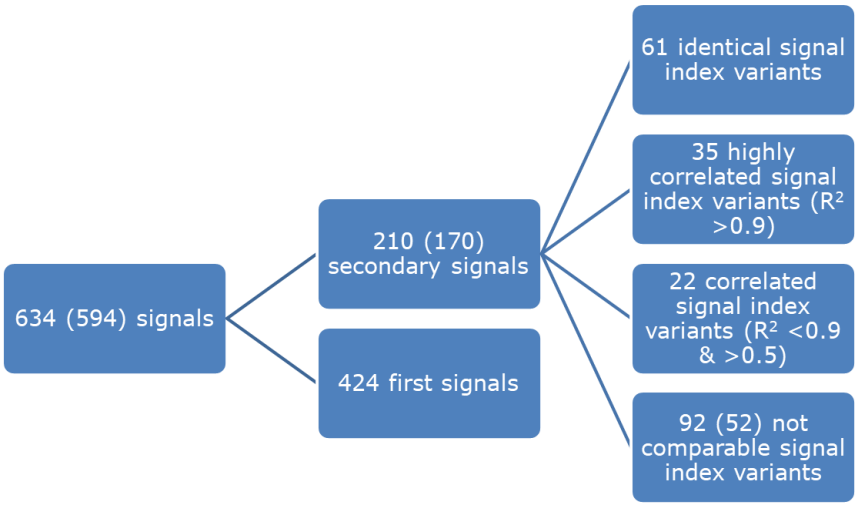


Figure 10: Comparison of signal index variants between 4/4 [25] and 6/7 [43] decimal digits analyses, with 634 and 594 signals, respectively. The signals were divided into first and secondary signals. The first signals are identical due to the identical unconditioned p-values. The index variants of the secondary signals were compared regarding the R^2 between the two analyses. R^2 was calculated based on the same reference panel used for conditioning. Numbers in braces refer to the number of signals in the 6/7 decimal digits analysis.

In contrast to the 13 signals in the 4/4 decimal digits analysis, that were not significant in the fully conditioned model, all signals in the 6/7 decimal digits analysis were

significant ($P< 5*10^{-8}$) in the fully conditioned model (except for first signals, which are not genome-wide significant in EUR in the unconditioned analysis).

3.1.2.2. Stability of credible set sizes

The total number of credible set variants in the 4/4 decimal digits analysis was 38,306 [25] and 35,885 for the 6/7 decimal digits analysis [43]. **Figure 11** shows how the sizes of smaller credible sets (below 50 SNPs in both analyses) of the 542 signals that were comparable between the two analyses compare to each other. Although there are some signals with larger differences, overall, the set sizes are comparable between the two analyses (r = 0.87, **Figure 11**). Of the original 44 single credible set variant signals and 94 signals with small sets, 40 and 79 signals are included in the comparable signals. Among the 40 comparable single variant signals from the 4/4 decimal digits analysis, 25 signals still have only one credible set variant while 15 signals have larger credible sets in the 6/7 decimal digits analysis. Among the 79 small sets from the 4/4 decimal digits analysis, 10 signals become single variant signals, 46 signals stay in the group of small credible sets and 23 signals have larger credible sets in the analysis with more decimal digits. Additionally, 18 signals have small credible sets in the 6/7 analysis which were having a larger credible set in the 4/4 decimal digits analysis (**Table 7**).

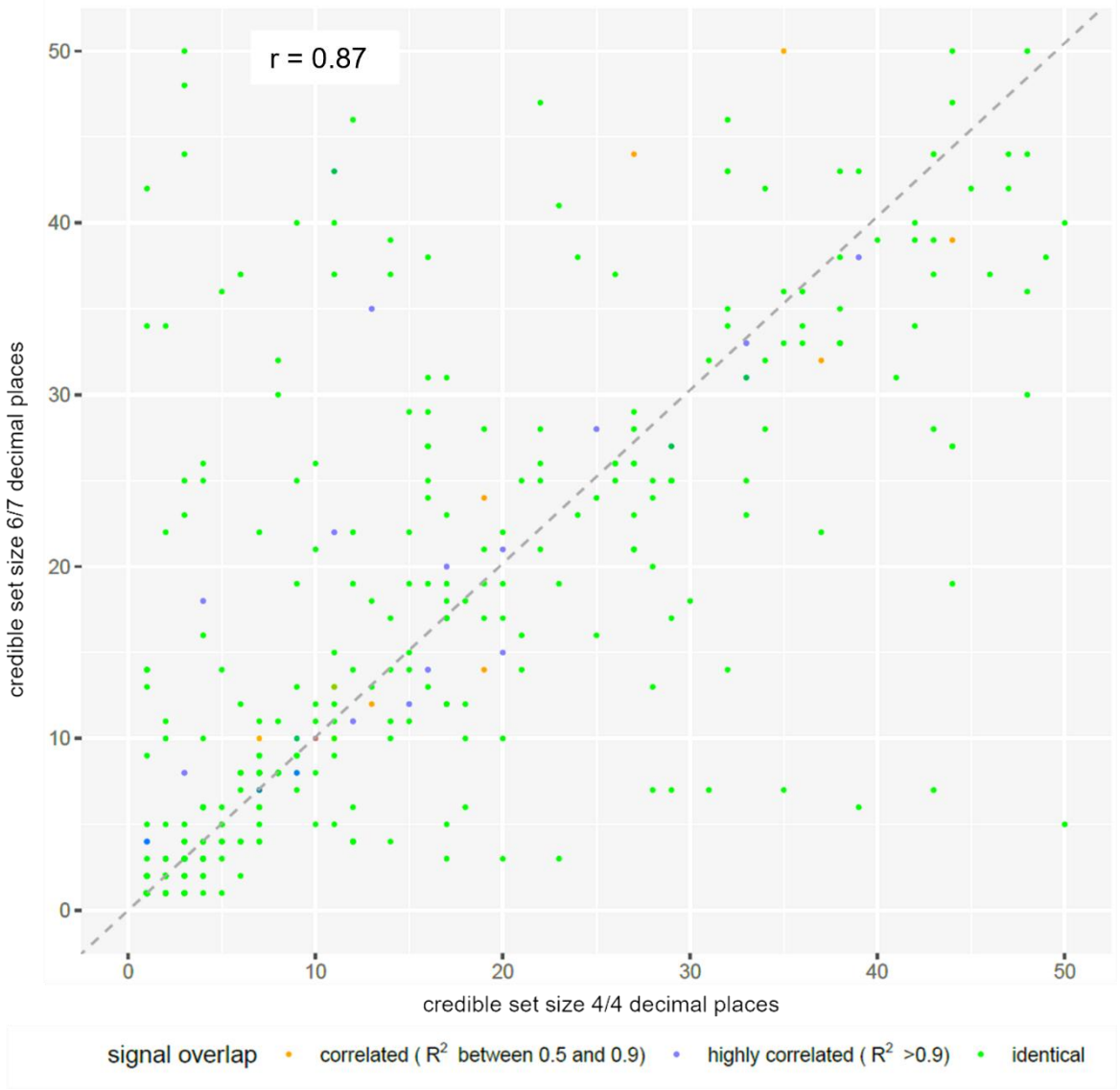


Figure 11: Comparison of credible set sizes between the 4/4 [25] and the 6/7 [43] decimal digits analyses of 542 comparable signals. Signals were mapped to each other between the two analyses by their signal index SNP. The colour indicates the type of correlation between two signal index SNPs. The graph is restricted to smaller credible set sizes with less than 50 credible set variants in both analyses. The dashed line is the least squares line with intercept=0 and slope=1.0093. The Pearson's correlation coefficient r is 0.87.

Table 7: Development of credible set size from 4/4 [25] to 6/7 [43] decimal digits for 542 comparable signals, which either have the same or correlated signal index SNPs. The signals were grouped by their credible set size in the 4/4 and the 6/7 decimal digits analysis. The table shows how many signals belong to the same and how many to different credible set size groups. Greenish colour indicates a same or smaller set size in the 6/7 decimal places analysis while reddish colour indicates a larger set size in the 6/7 decimal digits analysis.

		Credible set size 6/7 decimal digits						
		1	**2-5**	**6-10**	**11-50**	**51-100**	**101-1000**	**>1000**
Credible set size 4/4 decimal digits	**1**	25	9	1	5	0	0	0
	2-5	10	46	7	14	1	1	0
	6-10	0	7	29	15	3	0	0
	11-50	0	10	14	155	23	4	0
	51-100	0	1	0	15	40	11	0
	101-1000	0	0	2	10	16	65	1
	>1000	0	0	0	0	0	0	2

3.1.2.3. Stability of individual SNP PPA

23,902 credible set variants are overlapping between the two analyses, meaning they are credible set variants for the same (correlated) signal. Comparison of individual SNP PPA in the two analyses shows that there is only a moderate correlation and PPAs in the 6/7 decimal digits analysis tend to be smaller than in the 4/4 decimal digits analysis (**Figure 12**). Of the 40 SNPs with PPA>99% from the 4/4 decimal digits analysis, 25 SNPs still have a PPA >99% in the 6/7 decimal digits analysis (**Figure 13**). Additional 8 SNPs have a PPA >50% so that they are still the most likely causal SNP in their signal. 3 SNPs are not a credible set variant for the respective signal anymore.

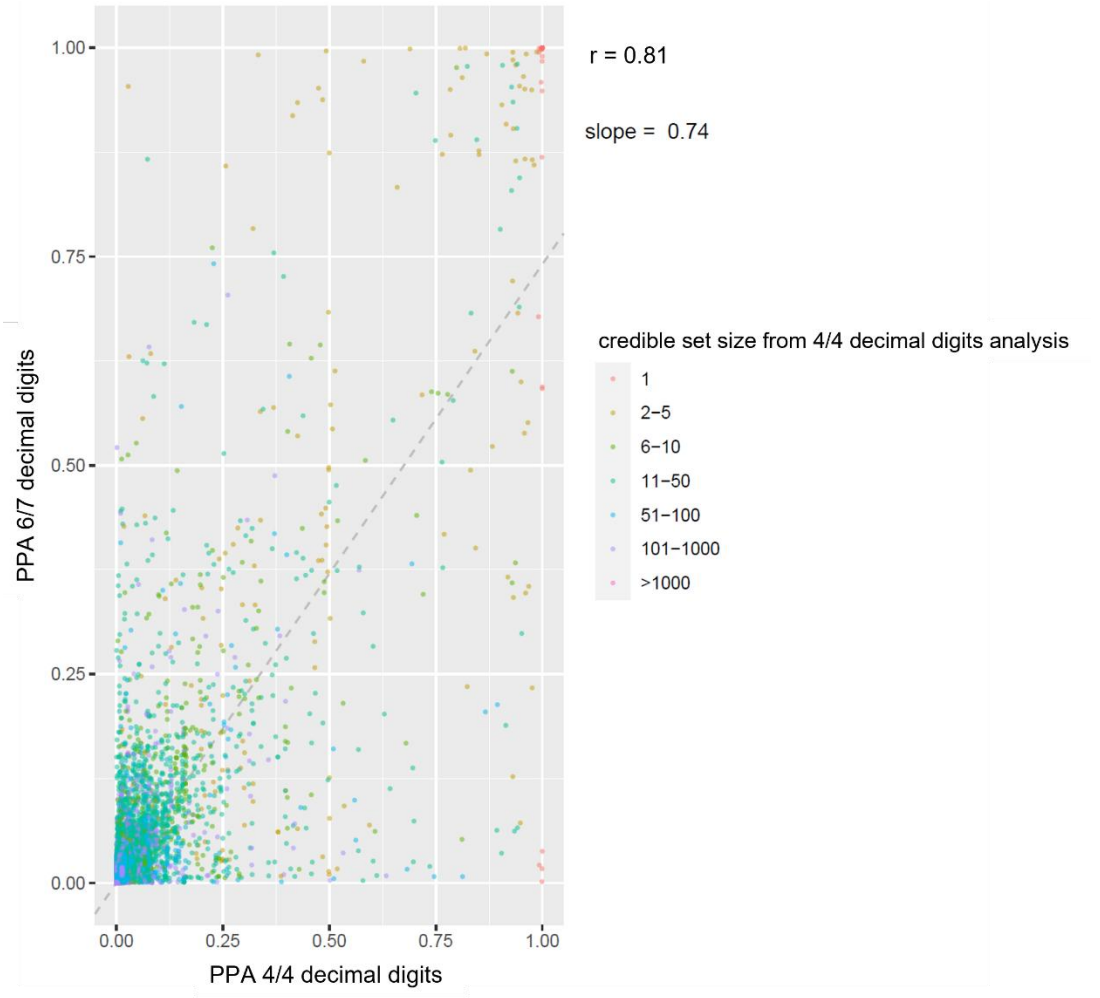


Figure 12: PPA comparison of 23,902 credible set variants that are credible sets variants of one of the 542 comparable signals in both (4/4- and 6/7-decimal digits) analyses. Colour indicates the credible set size of the corresponding signal in the 4/4 decimal digits analysis. The dashed line is the least squares line with intercept=0 and slope=0.74. The Pearson's correlation coefficient r is 0.81.

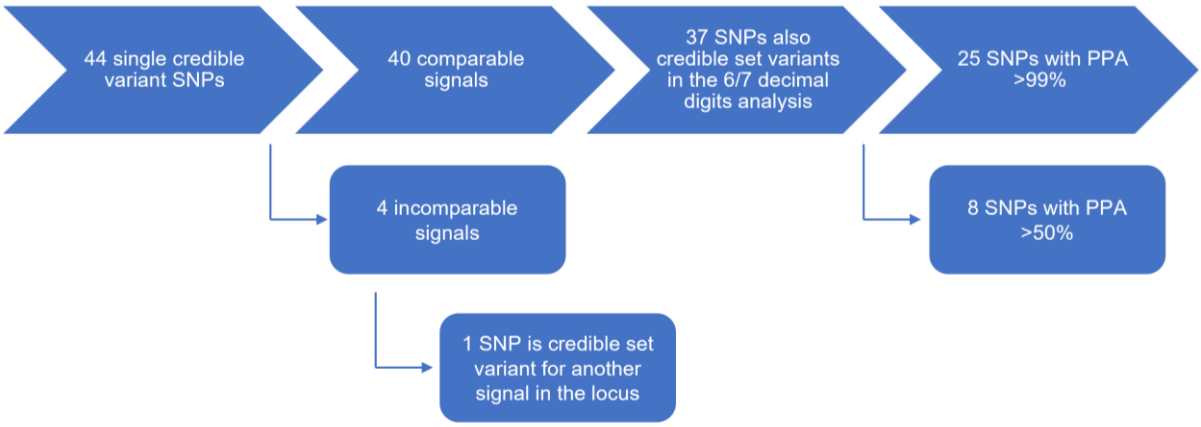


Figure 13: Behaviour of the 44 single credible set variant signals from the 4/4 decimal digits analysis in the 6/7 decimal digits analysis.

The individual SNP PPA is a comparative measurement comparing the approximate bayes factor (ABF) of a SNP to the sum of all ABFs in the respective signal. Thus, smaller changes in the ABFs of other SNPs in a signal might strongly affect the individual SNPs PPA. Therefore, also the SNP ABFs (which mostly reflect the fully conditioned p-values especially in the 6/7 decimal digits analysis, **Figure 14)** from the 4/4 decimal digits analysis were compared to the 6/7 decimal digits analysis. With a Pearson's correlation coefficient r of 0.95 this absolute measurement is more stable across different numbers of decimal digits than the PPA (**Figure 15**).

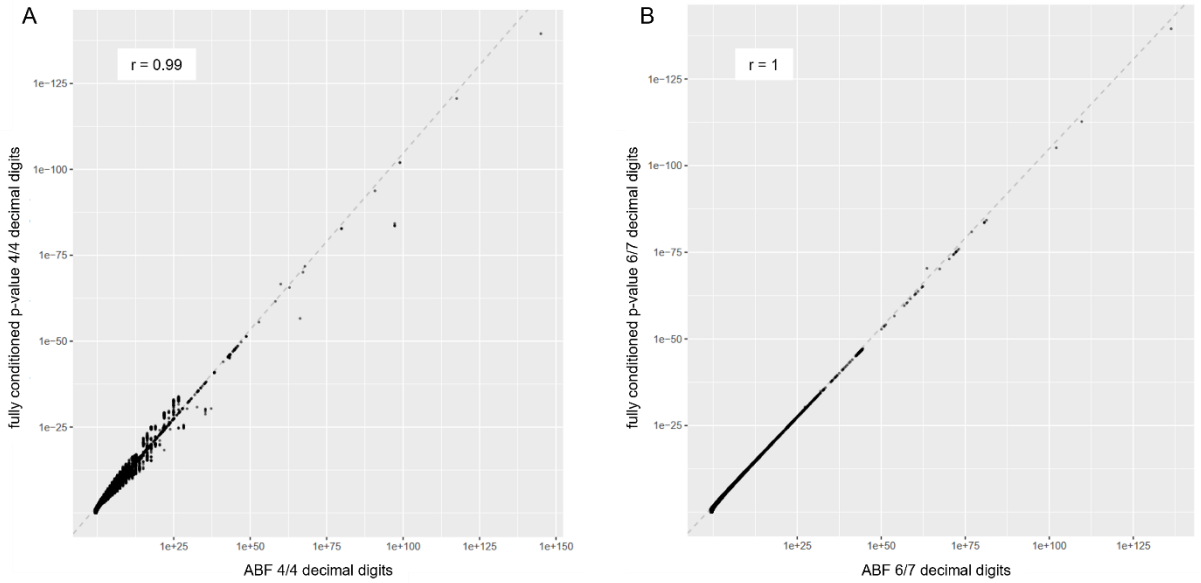


Figure 14: Comparison of fully conditioned p-value and ABF in the 4/4- and the 6/7- decimal digits analysis. of 23,902 credible set variants that are credible sets variants of one of the 542 comparable signals in both (the 4/4- and 6/7-decimal digits) analyses. Panel A shows the comparison in the 4/4 decimal digits analysis and panel B in the 6/7 decimal digits analysis. The dashed lines are the least squares lines and r is the respective Pearson's correlation coefficient.

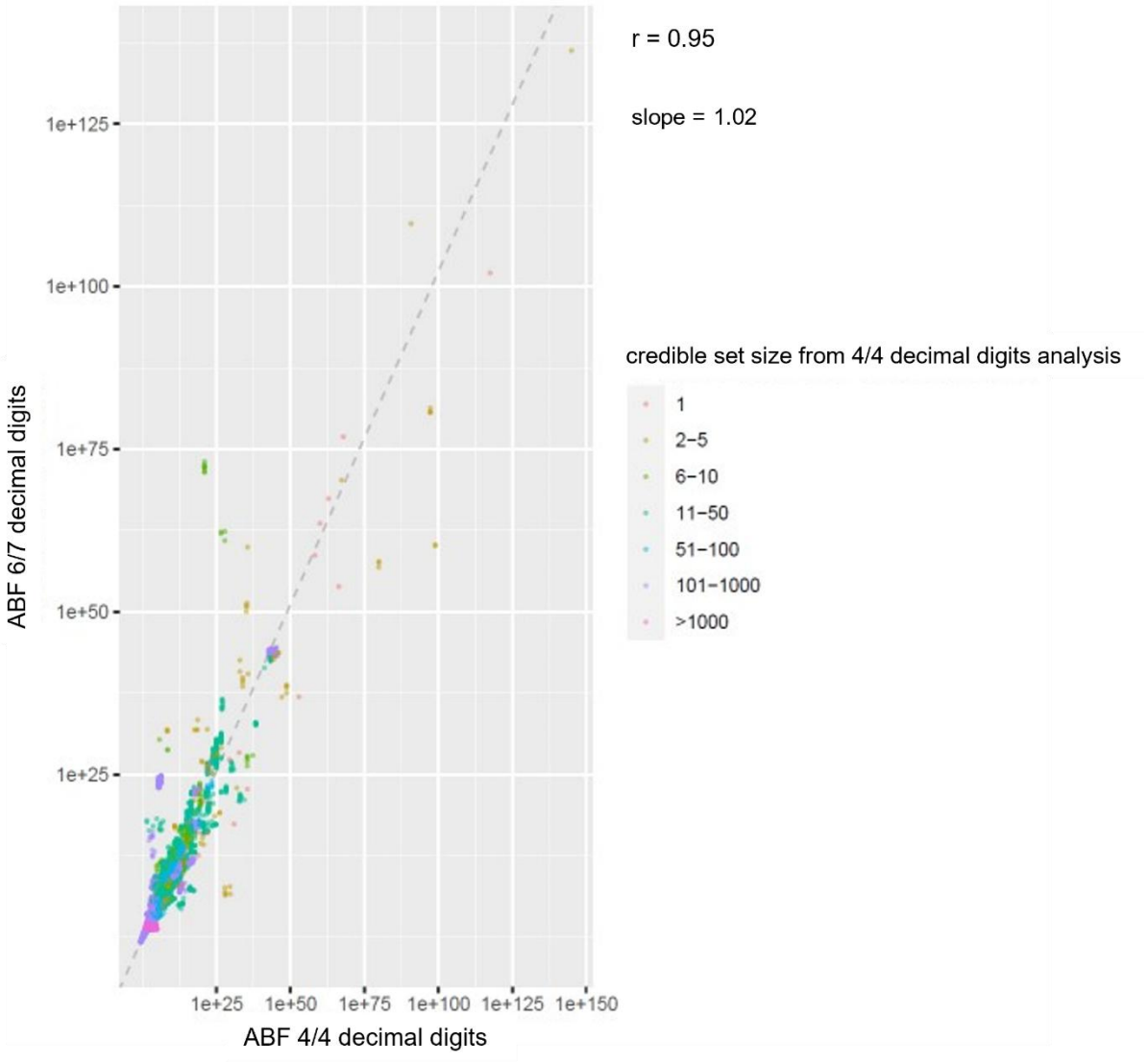


Figure 15: ABF comparison of 23,902 credible set variants that are credible sets variants of one of the 542 comparable signals in both (the 4/4- and 6/7-decimal digits) analyses. Colour indicates the credible set size of the corresponding signal in the 4/4 decimal digits analysis. The dashed line is the least squares line with intercept=0 and slope=1.02. The Pearson's correlation coefficient r is 0.95.

3.2. KidneyGPS: a tool to annotate variants and genes for kidney function to allow for prioritisation of GWAS results as well as to make this information easily accessible to a broader scientific community

KidneyGPS is a web-application programmed as a R-Shiny App and the version 2.3.1 is available at https://kidneygps.ur.de/gps/. The application was presented in the publication of Stanzick et al. 2023 in BMC Bioinformatics [43]. Therefore, the next chapters will include numbers and aspects from the publication. These will be marked by citations.

3.2.1. Pre-KidneyGPS data processing results

The data backbone of KidneyGPS is the R-Workspace described in chapter 2.6.7 of this thesis. Thus, comprehensive evidence for all 5906 genes located in the 424 GWAS loci associated with eGFR and the 35,885 variants of 99% credible sets of the 594 independent association signals (based on the 6/7 decimal digits analysis) is included. As described in Stanzick et al. 2023, credible set variants of small sets with 5 or less variants and variants with a high PPA over 50% show strong statistical support of being causal for the association. Of the 594 signals, 172 signals with 335 variants fulfil one of these two criteria (see Stanzick et al. 2023, Figure 1) [43].

The GWAS data from Stanzick et al. 2021 for eGFR estimated from serum cystatin C (eGFRcys) and blood urea nitrogen (BUN) supports 491 of the 594 signals for eGFR [25, 43]. This is because the corresponding locus lead variant of the eGFR locus is nominally significant associated with eGFRcys with same effect direction and/or BUN with opposite effect direction [25, 43]. 8 eGFR signal index variants were also associated with eGFR decline or highly correlated to an eGFR decline associated variant [43, 58]. Based on the data of Winkler et al. 2022, one eGFR signal index variant showed only significant eGFR association in persons with diabetes mellitus (DM) while 5 other signal index variants had stronger effects in persons with DM despite being associated with eGFR also in persons without DM and one signal index variant was only associated in persons without DM [43, 57].

In total, 13,716 credible set variants were mapped to any of 940 genes as being protein-relevant or eQTL/sQTL in kidney tissue [43]. 23 of those variants had a PPA >99% and 47 variants a PPA between 50% and 99% (Table 1 of Stanzick et al 2023)

[43]. While those statistical supported variants were nearly equally mapped to a gene by being protein-relevant or eQTL in kidney tissue, variants with less than 50% PPA or in small sets were 10-times more often mapped by the eQTL data (Table 1 of Stanzick et al 2023.) [43]. The eQTLs of statistical supported variants were found slightly more often in tubulo-interstitial tissue (PPA>99%: 10 vs 7 variants; PPA 50-99%: 22 vs 18 variants; other small set variants: 54 vs 38 variants, Stanzick et al. 2023 Table 1), while the other eQTLs were found nearly equally in tubulo-interstitial tissue and glomerular tissue (10,472 and 10,562, respectively) [43]. Independent of that, 381 of the 5906 genes had a known kidney phenotype in mouse or human or were drug targets for kidney diseases [43].

3.2.2. Presentation of app.r (R-script for running KidneyGPS)

KidneyGPS runs when the app.r script is executed. This is located in a folder with the *run_before_app_server.r* script described in chapter 2.6.7 and two additional folders: “/data” and “/www” (**Figure 16**). The “/data” folder contains the two files *packages.r*, which is a script to load all required packages, and “workspace.Rdata”, which is the R-workspace created by the *run_before_app_server.r* script. Both files are loaded and the *packages.r* script is executed in the app.r script before the Shiny logic starts. Shiny itself requires the definition of “ui” (User Interface) and “server”. Within the definition of “ui” the “style.css” and “kidney-g8a69f0709_1920.png” are loaded from the “/www” folder.

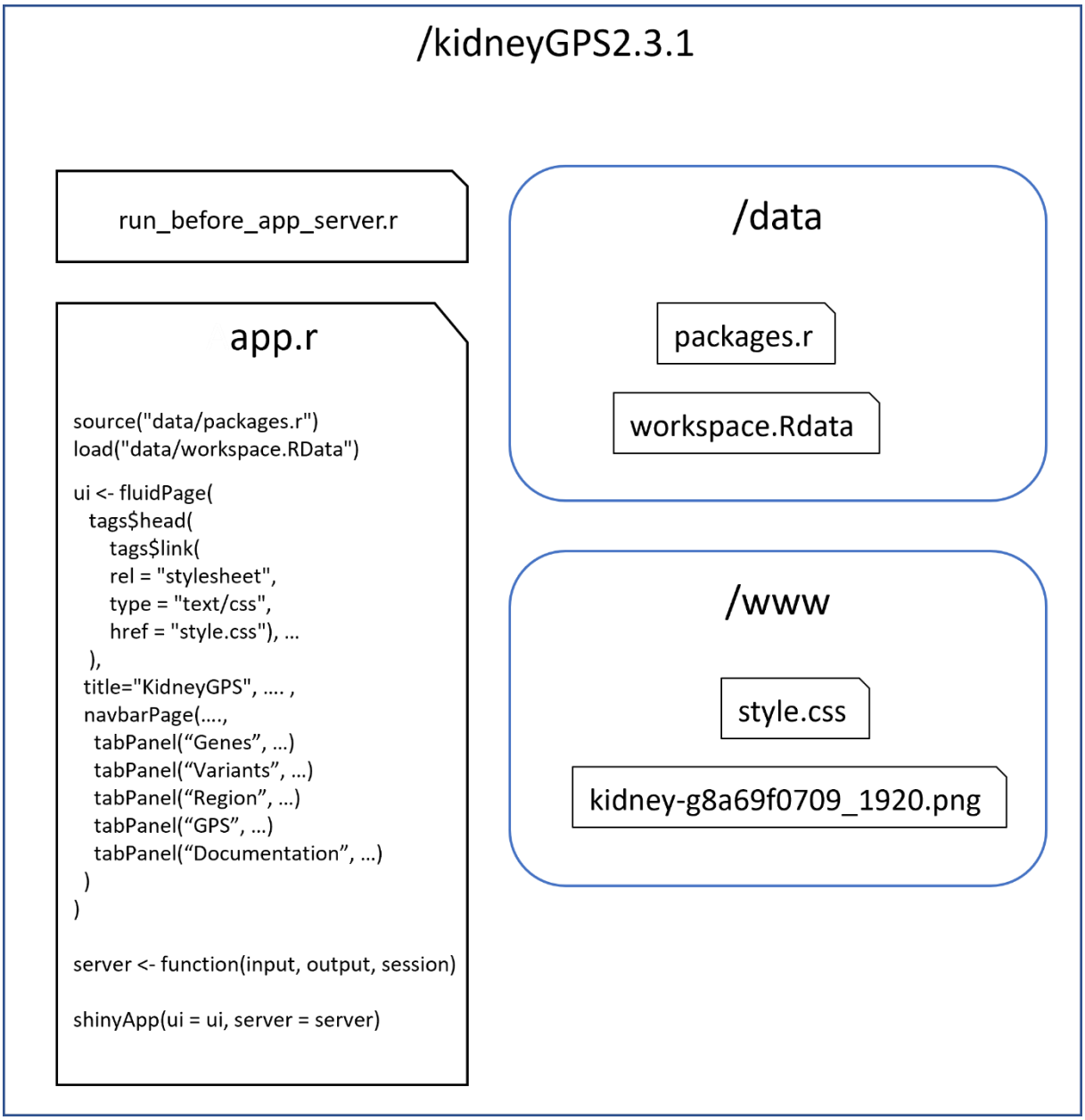


Figure 16: Folder structure required for KidneyGPS. Parts with black lining describe R-scripts or other necessary resources, parts with blue linings describe required folder structures.

3.2.2.1. app.r: General structure and important elements of user interface and server part

The whole code of the app.r script to run KidneyGPS is part of the appendix of this work. As for all R-Shiny applications the code is separated in the definition of the user interface (ui) which defines the whole appearance of the application, and the server which defines all functions needed to make the app reactive to user interaction.

User interface

The user interface of KidneyGPS is designed as a fluid page, which means, that the layout consists of rows which include columns. All elements within one row appear on the same hight if the browser window has adequate width. The components are scaled in real-time to fill all available browser width. Before starting the page layout, global elements are defined within the 'tags$head' function and with the definition of the title. Global styling information is loaded from the CSS script (described in chapter 3.2.2.3) and the reaction to the "Enter" key is defined, which also works for the whole app. The layout itself contains basically three main rows: The header, the descriptive paragraph and a "navbarPage" (**Figure 17**). The navbarPage contains the 5 'tabPanels' "Genes", "Variants", "Region", "GPS" and "Documentation & Help", which can be selected with the tabs. Each panel content than has its own layout (whole source code shown in appendix).

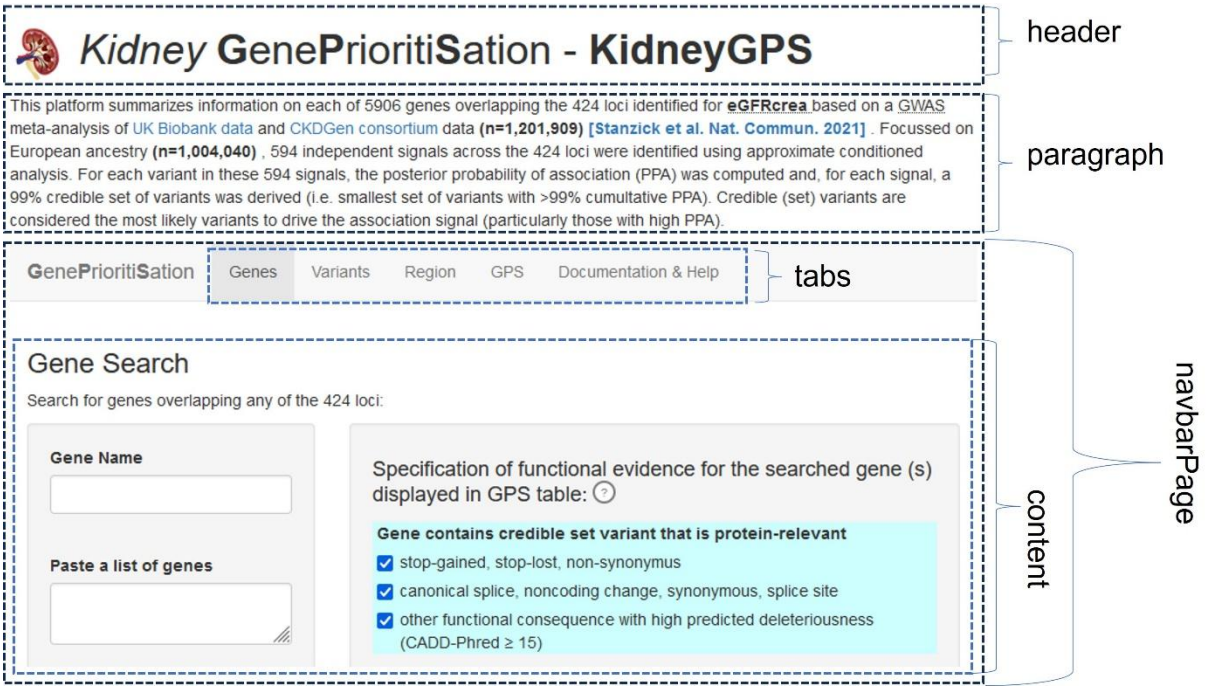


Figure 17: KidneyGPS Layout. The user interface of KidneyGPS is divided in three main parts: header, introducing paragraph and the RShiny element 'navbarPage'. This last element contains different tabs, which each have a different content.

All layouts have in common that they consist of static object, like paragraphs of text, input variables, which are all objects that can take user input, and output variables, which display the output generated by the server functions. All input variables are listed in **Table 8**. Their location in the code relates to the location on the webpage. The kind of widget defines the appearance of the input variable and which kind of input values it can store. These can be an uploaded file, text or numbers entered by the user, text

or numbers referring to selection options, or Boolean values. All input values can be accessed via input$”Variable name” in the server function.

Table 8: List of input variables defined in the UI part of KidneyGPS. The variable name is used in the server part to access these elements. The column “Kind of Input/Widget” contains the RShiny nomenclature for the respective type of input element, except for the first row which describes the variable for the reaction on the “Enter” key. The columns “Context” and “On which tab” allow to localize the input element on the KidneyGPS webpage. The column “Stored value” presents either all possible values stored in the respective input element or gives the data type.

Variable name/ID	**Kind of Input/Widget**	**context**	**On which tab**	**Stored value**
keyPressed	Reacts on “Enter”	-	Every Page	Random number
Genename	textInput	Gene name	Gene	Entered string
gene_batch	textAreaInput	Gene names	Gene	Entered string
file2	fileInput	File with gene names	Gene	File
reset_gene	actionButton	File reset	Gene	Increasing integers
sep_gene	radioButtons	File separator	Gene	“,” “;” “\”
gene_go	actionButton	Start search	Gene	Increasing integers
question_GPS_gene	actionButton	Question column selection	Gene	Increasing integers
columns1	checkboxGroupInput	Protein altering column selection	Gene	11,12,13
columns2	checkboxGroupInput	eQTL/sQTL column selection	Gene	14, 15, 16, 17, 18, 19, 22
columns3	checkboxGroupInput	Kidney phenotypes column selection	Gene	20, 21
columns4	checkboxGroupInput	drugability column selection	Gene	25, 26
question_details_gene	actionButton	Question additional information	Gene	Increasing integers
detail_evidence_locus_gene	checkboxGroupInput	Additional locus information	Gene	“summary”, “cred_var”, “dm_stat”
gene.locuszoom	checkboxGroupInput	Locuszoom	Gene	“locus_zoom”
ppa	numericInput	PPA restriction	Gene	Entered value between 0 and 1
question_SNP	actionButton	Overall SNP search question	SNP	Increasing integers
snp	textInput	Single SNP search	SNP	Entered string
SNP_batch	textAreaInput	Multiple SNP search	SNP	Entered string
file1	fileInput	SNP file	SNP	uploaded file
reset_variant	actionButton	Reset uploaded file	SNP	Increasing integers
sep_snp	radioButtons	SNP file separator	SNP	“,” , “;” , “\t”

Variable name/ID	Kind of Input/Widget	context	On which tab	Stored value
snp_go	actionButton	Start SNP search	SNP	Increasing integers
variant-search-options	checkboxGroupInput	Select gws SNPs and/or credible SNPs	SNP	“gws”, “credvar”
detail_evidence_variant	checkboxGroupInput	SNP annotations	SNP	"CADD", "NEPTUNE_glo", "NEPTUNE_tub", "GTEx_eQTL", "GTEx_sQTL", "GTEx_wo_kidney_eQTL", "GTEx_wo_kidney_sQTL", "dm_stat"
snp.locuszoom	checkboxGroupInput	SNP locuszoom	SNP	“locus_zoom”
Chromosome	selectInput	Chromosome for region	Region	1 to 22 (single value)
pos_start	numericInput	Start position	Region	Entered integer
pos_end	numericInput	End position	Region	Entered integer
region_go	actionButton	Start region search	Region	Increasing integers
signalppafilter	checkboxInput	Restrict signals by PPA criterium	GPS	TRUE/FALSE
signalhigh	radioButtons	Select PPA	GPS	0.99, 0.5, 0.1
signalsmall	checkboxInput	Restrict to small signals	GPS	TRUE/FALSE
signalnofilter	checkboxInput	Show all signals	GPS	TRUE/FALSE
signalcysbun	radioButtons	Restrict to validated loci	GPS	TRUE/FALSE
signalDM	radioButtons	Restrict to DMdiff signals	GPS	TRUE/FALSE
signaldecline	radioButtons	Restrict to decline signals	GPS	TRUE/FALSE
genemapprot	checkboxGroupInput	genes with protein altering	GPS	11,12,13
genemapeqtl	checkboxGroupInput	Genes with eQTL	GPS	14, 15, 16, 17, 18, 19
mapppafilter	checkboxInput	Mapping with PPA filter	GPS	TRUE/FALSE
maphigh	radioButtons	Select PPA	GPS	0.99, 0.5, 0.1
mapsmall	checkboxInput	Map small signals	GPS	TRUE/FALSE
nofiltermapppa	checkboxInput	No additional mapping filter	GPS	TRUE/FALSE
mapnofilter	checkboxInput	No mapping	GPS	TRUE/FALSE
genelistpheno	checkboxGroupInput	Genes with phenotype	GPS	“mouse”, “human”
genelistdrug	checkboxGroupInput	Drugable genes	GPS	“kidney”, “other”
genelistnofilter	checkboxInput	No pheno/drug filter	GPS	TRUE/FALSE
gpsgo	actionButton	Start prioritisation	GPS	Increasing integers

The output variables follow the same logic as the input variables. They are defined in the "ui" part of the app.r script at the place where the output should appear on the webpage. Variable names must be unique. The output itself is generated within the "server" function. There are mainly three kinds of output used in KidneyGPS: (1) textOutput: This kind of output allows to render text in form of paragraphs, (2) DT::dataTableOutput: This allows to render data-tables, (3) uiOutput: This kind of output allows to define a user-interface section within the server part depending on other variables; it is mostly used to enable or disable specific input-options or to generate locus-zoom plot links. All variables are listed in **Table 9**. To attach content to the output variables these are assessed in the server via output$"variable_name".

Table 9: List of output variables defined in the UI part of KidneyGPS. The "Kind of Output" column refers to the RShiny nomenclature for the respective type of output. The "On which tab" column shows on which tab of KidneyGPS the output is shown and the "content" column briefly describes the output.

Variable Name	**Kind of Output**	**On which tab**	**content**
testforme	textOutput	Gene	Check if entered genes are HGNC gene names
namecheck_gene	textOutput	Gene	Check if genes are in GPS
GPSrows	DT::dataTableOutput	Gene	GPS excerpt of searched genes
CADD_gene_text	textOutput	Gene	Description of CADD results
CADD_gene	DT::dataTableOutput	Gene	CADD results
glo_gene_text	textOutput	Gene	Description of eQTL glomerulus results
glo_gene	DT::dataTableOutput	Gene	eQTL glomerulus results
tub_gene_text	textOutput	Gene	Description of eQTL tubulus results
tub_gene	DT::dataTableOutput	Gene	eQTL tubulus results
eqtl_gene_text	textOutput	Gene	Description of GTEx eQTL kidney results
eqtl_gene	DT::dataTableOutput	Gene	GTEx eQTL kidney results
sqtl_gene_text	textOutput	Gene	Description of GTEx sQTL kidney results
sqtl_gene	DT::dataTableOutput	Gene	GTEx sQTL kidney results
eqtl_wo_kidney_gene_text	textOutput	Gene	Description of GTEx eQTL other tissue results
eqtl_wo_kidney_gene	DT::dataTableOutput	Gene	GTEx eQTL other tissue results
sqtl_wo_kidney_gene_text	textOutput	Gene	Description of GTEx sQTL other tissue results
sqtl_wo_kidney_gene	DT::dataTableOutput	Gene	GTEx sQTL other tissues results
mgi_gene_text	textOutput	Gene	Description of MGI results
mgi_gene	DT::dataTableOutput	Gene	MGI results
omim_gene_text	textOutput	Gene	Description of human phenotypes results
omim_gene	DT::dataTableOutput	Gene	Human phenotypes results
drug_gene_kid_text	textOutput	Gene	Description of kidney disease drugs results
drug_gene_kid	DT::dataTableOutput	Gene	kidney disease drugs results
drug_gene_oth_text	textOutput	Gene	Description of other disease drugs results
drug_gene_oth	DT::dataTableOutput	Gene	other disease drugs results

Variable Name	Kind of Output	On which tab	content
summary_text	textOutput	Gene	Not used
summary	DT::dataTableOutput	Gene	Signal table including searched genes
cred_var_gene	DT::dataTableOutput	Gene	Full list of credible set variants of locus containing searched gene
dm_stat_gene	DT::dataTableOutput	Gene	Association statistics separated by DM status for credible set variants
LocusZoom_gene_text	textOutput	Gene	Not used
plot_download_links_gene	uiOutput	Gene	Link to locus-zoom plot
detail.evidence.snp	uiOutput	SNP	Search options when credible set variants are selected (CADD, eQTL…)
snpinputcheck	textOutput	SNP	Not used
namecheck	textOutput	SNP	Is SNP gws associated with eGFR
all_var	DT::dataTableOutput	SNP	Association statistics all ancestries for gws SNPs
namecheck2	textOutput	SNP	Is SNP a credible set variant
cred_var	DT::dataTableOutput	SNP	Association statistics EUR ancestry conditioned and unconditioned for credible set variants
CADD_text	textOutput	SNP	Description of CADD results
CADD	DT::dataTableOutput	SNP	CADD results
glo_text	textOutput	SNP	Description of eQTL glomerulus results
glo	DT::dataTableOutput	SNP	eQTL glomerulus results
tub_text	textOutput	SNP	Description of eQTL tubulus results
tub	DT::dataTableOutput	SNP	eQTL tubulus results
eqtl_text	textOutput	SNP	Description of GTEx eQTL kidney results
eqtl	DT::dataTableOutput	SNP	GTEx eQTL kidney results
sqtl_text	textOutput	SNP	Description of GTEx sQTL kidney results
sqtl	DT::dataTableOutput	SNP	GTEx sQTL kidney results
eqtl_wo_kidney_text	textOutput	SNP	Description of GTEx eQTL other tissue results
eqtl_wo_kidney	DT::dataTableOutput	SNP	GTEx eQTL other tissue results
sqtl_wo_kidney_text	textOutput	SNP	Description of GTEx sQTL other tissue results
sqtl_wo_kidney	DT::dataTableOutput	SNP	GTEx sQTL other tissues results
dm_stat_snp	DT::dataTableOutput	SNP	Association statistics separated by DM status for credible set variants
LocusZoom_SNP_text	textOutput	SNP	Not used
plot_download_links_snp	uiOutput	SNP	Link to Locus-zoom plot
valid_gene_test	textOutput	Region	Test if entered region makes sense
ov_loci	DT::dataTableOutput	Region	Region overlapping signals
GPS_region_search	DT::dataTableOutput	Region	GPS of region overlapping loci
signalppafilterdetail	uiOutput	GPS	PPA signal filter options section
detail.genemap.ppa	uiOutput	GPS	Variant restriction option section in gene-map section
mapppafilterdetail	uiOutput	GPS	PPA filter options for gene-map
GPS_datadescription	verbatimTextOutput	GPS	Used as test option
GPS	DT::dataTableOutput	GPS	GPS based on filtering
GPS_Genes1	textOutput	GPS	Number of genes in filtered GPS

In contrast to the input and output variables, the static user interface items do not have to be labelled. However, they can be labelled to allow to change their visibility in the server part. Besides of the kidney image in the header of the webpage, all these objects are divisions that contain other objects like paragraphs, headers, input-values or output-values. A full list is given in **Table 10**.

Table 10: List of labelled static user interface items. Multiple static UI items are labelled with an "id" to access them in the server part of KidneyGPS.

Kind of object	**id**	**description**
tags$img	kidney_image	Kidney image in header
div	gene-upload-section	Gene file upload section
div	gene_batch_upload_options	Options for gene file upload section
div	gene_go_div	Gene go button
div	Gene_div	Overall gene search output division
div	GPS_gene	Gene search GPS output division
div	div_cadd_gene	Gene search CADD output division
div	div_glo_gene	Gene search glomerulus eQTL output division
div	div_tub_gene	Gene search tubulus eQTL output division
div	div_GTEx_eqtl_gene	Gene search GTEx kidney eQTL output division
div	div_GTEx_sqtl_gene	Gene search GTEx kidney sQTL output division
div	div_GTEx_wo_kidney_eqtl_gene	Gene search GTEx other tissue eQTL output division
div	div_GTEx_wo_kidney_sqtl_gene	Gene search GTEx other tissue sQTL output division
div	div_mgi_gene	Gene search MGI output division
div	div_omim_gene	Gene search human phenotype output division
div	div_drug_gene_kid	Gene search kidney disease drugs output division
div	div_drug_gene_oth	Gene search other diseases drugs output division
div	div_summary	Gene search region table output division
div	div_cred_var_gene	Gene search credible variants output division
div	div_dm_stat_gene	Gene search eGFR by DM status output division
div	div_locus_zoom_gene	Gene search locus-zoom output division
div	snp-upload-section	SNP file upload section
div	SNP_batch_upload_options	Options for SNP file upload section
div	snp_go_div	SNP go button
div	SNP_div	Overall SNP search output
div	gws_div	All ancestries associtaion statistics
div	cred_var_div	European ancestry association statistics conditioned and unconditioned
div	div_cadd_snp	SNP search CADD output division
div	div_glo_snp	SNP search glomerulus eQTL output division
div	div_tub_snp	SNP search tubulus eQTL output division
div	div_GTEx_eqtl_snp	SNP search GTEx kidney eQTL output division
div	div_GTEx_sqtl_snp	SNP search GTEx kidney sQTL output division
div	div_GTEx_eqtl_wo_kidney_snp	SNP search GTEx other tissue eQTL output division
div	div_GTEx_sqtl_wo_kidney_snp	SNP search GTEx other tissue sQTL output division
div	div_dm_stat_snp	SNP search eGFR by DM status output division
div	div_locus_zoom_snp	SNP search locus-zoom output division
div	region_go_div	Go button region search
div	region_div	Region search overall output

Kind of object	id	description
bsCollapse	regionCollapse	Collapsible region search output, contains to panles
div	gpscontainersignal	Signal filter panel on GPS page
div	gpsppafilter	PPA filter options in signal filter
div	nofiltersignal	No filter button for signal filter
div	furthersignalfilter	Additional signal filter options
div	gpscontainermap	Variant-to-gene mapping panel on GPS page
div	ppafiltergenemap	PPA filter options for variant-gene mapping
div	nofiltermap	No filter button for variant-gene mapping
div	gpscontainerfilter	Gene-to-phenotype mapping section
div	nofiltergenelist	No filter button for gene-phenotype mapping

Server

The server part of the app.r script consists of various R functions that enable the application to react on the user input to create specific output that is displayed via the output variables. Besides the values stored in the input variables, the server functions have access to the variables created by the run_before_app_server.r script and are stored in the R-Workspace. The logic of R-Shiny applications is, to execute each function each time a reactive value inside the function changes. The input variables are such reactive values, while the variables from the R-Workspace are static. Additional reactive values can be generated to store intermediate results. **Table 11** contains a full list of these additional reactive values. Reactive behaviour of the application can be suppressed by functions like “eventReactive” or “observeEvent” which will only execute if one specific value changes. This is done throughout the app, so that the actualisation of searches will only be executed when the respective button is clicked. All changes on search options made before clicking the button are stored in the additional reactive values.

Table 11: Additional reactive values created in the server part of KidneyGPS to store intermediate results.

Variable	description	Stored values
go$gps_correct	Check if all required GPS filter options are selected	TRUE/FALSE
gps_gps_rows$signalfilter	Rows of GPS which meet the selected signal filter options	Vector with row indices
gps_gps_rows$mapfilter	Rows of GPS which meet the selected variant-to-gene mapping filter options	Vector with row indices
gps_gps_rows$genelistfilter	Rows of GPS which meet the selected gene-to-phenotype mapping options filter options	Vector with row indices
map()	Selected variant-to-gene mapping options	Vector with respective column indices
gps_version_all_show()	Filtered GPS with gps_gps_rows$signalfilter	Data frame of filtered GPS_show

Variable	description	Stored values
gps_version_10_show()	Filtered GPS (only variants with PPA ≥10%) with gps_gps_rows$signalfilter	Data frame of filtered GPS_10_show
gps_version_50_show()	Filtered GPS (only variants with PPA ≥50%) with gps_gps_rows$signalfilter	Data frame of filtered GPS_50_show
ppa_filter_gps()	Summary of the selected additional variant restriction in variant-to-gene mapping	Character ("no_filter", "small", "small_0.99", "small_0.5", "small_0.1", “small”)
gps_version_all_show_map()	gps_version_all_show() filtered with variant-to-gene options	Data frame
gps_version_10_show_map()	gps_version_10_show() filtered with variant-to-gene options	Data frame
gps_version_50_show_map()	gps_version_50_show() filtered with variant-to-gene options	Data frame
gps_version_mapped_show()	Combination of gps_version_all_show_map(), gps_version_10_show_map(), gps_version_50_show_map()	Data frame
gps_version_mapped_pheno_show()	gps_version_mapped_show() filtered with gene-to-phenotype options	Data frame
go_correct()	Was the “Go prioritize!” button clicked?	TRUE/FALSE
gps_show_final()	Displayed version of GPS (filtered GPS when go_correct() ==TRUE, else unfiltered version)	Data frame
go$region	Check if entered positions are not zero and if “Go” is clicked or the enter key was pressed	Integer (increasing)
region_start()	Store entered region start after starting search	Should be integer
region_start()	Store entered region end after starting search	Should be integer
valid_reg()	Check if region is valid	TRUE/FALSE
chr()	Store selected chromosome	Integer
ov_loci()	region_table_show reduced to overlapping loci	Data frame
overlap_loci()	Are there overlapping loci	TRUE/FALSE
GPS_region_search()	Part of GPS_show with overlapping loci	Data frame
inputtype$snp	Which option to enter input was last clicked?	String ("single", “list”, “upload”)
go$snp	Was the go button clicked or enter key was pressed	Integer (increasing)
single_snp()	Value of single SNP input	string
input_snp_batch_list()	Values of batch SNP input	vector
input_snp_batch_upload$input	Values from SNP file upload	vector
snp()	Stores value(s) from single_snp() or input_snp_batch_list() or input_snp_batch_upload$input depending on inputtype$snp	vector
inputtype$snp_final	Stores value of inputtype$snp after search was started	string
cred_vars()	Searched SNPs which are credible variants	vector
valid()	Is cred_vars() not empty?	TRUE/FALSE

Variable	description	Stored values
detail_evidence_variant()	SNP search options	Vector with strings
gws_snps()	Searched SNPs which are genome-wide significant	vector
gws()	Is gws_snps() not empty?	TRUE/FALSE
Locus_ID_snp()	Locus IDs of Searched SNPs	vector
snp_plots()	Links to Locus Zoom plots in HTML format	vector
inputtype$gene	Which option to enter input was last clicked?	String ("single", “list”, “upload”)
go$gene	Was the go button clicked or enter key was pressed	Integer (increasing)
single_gene()	Value of single gene input	string
input_gene_batch_list()	Values of batch gene input	vector
input_gene_batch_upload$input	Values from gene file upload	vector
gene()	Stores value(s) from single_gene() or input_gene_batch_list() or input_gene_batch_upload$input depending on inputtype$gene	vector
inputtype$gene_final	Stores value of inputtype$gene after search was started	string
corrected_genenames()	Indices of searched genes, whose names can be translated in HGNC gene names	Vector with indices
correct_genenames()	Subset of searched genes with corrected gene names and which are included in reference gene list	vector
genes_in_gps()	Subset of searched genes which are in GPS	vector
valid_gene()	Is genes_in_gps() not empty	TRUE/FALSE
columns()	Summary vector of gene search options	Vector (column indices)
detail_evidence_locus_gene()	Stores value of input$detail_evidence_locus_gene after search was started	vector
ppa()	Stores value of input$ppa after search was started	numeric
incorrect_genenames()	Number of searched genes which are not included in correct_genenames()	integer
cadd_gene()	Subset of CADD table with searched genes	Data frame
glo_gene()	Subset of glo table with searched genes	Data frame
tub_gene()	Subset of tub table with searched genes	Data frame
eqtl_gene()	Subset of GTEx_eQTL table with searched genes	Data frame
eqtl_wo_kidney_gene()	Subset of gtex_without_kidney_show table with searched genes	Data frame
sqtl_gene()	Subset of GTEx_sQTL table with searched genes	Data frame
sqtl_wo_kidney_gene()	Subset of sqtl_without_kidney_show table with searched genes	Data frame
mgi_gene()	Subset of MGI_showtable with searched genes	Data frame
omim_gene()	Subset of OMIM_MIM table with searched genes	Data frame

Variable	description	Stored values
drug_gene_kid()	Subset of drugability_show table with searched genes and kidney disease indication	Data frame
drug_gene_oth()	Subset of drugability_show table with searched genes and without kidney disease indication	Data frame
GPS_gene()	GPS of searched genes with selected properties	Data frame
rowsGPS()	Number of rows in GPS_gene()	interger
Locus_ID()	Locus IDs of searched genes	vector
gene_plots()	Links to Locus Zoom plots in HTML format	vector

3.2.2.2. Detailed description of important functionalities

Main functions for the users of KidneyGPS are the gene-, variant- and region search, and the gene-prioritisation. These require various small process that are included in the KidneyGPS code. The most important ones are explained below.

The gene and variant search have a similar structure. Both allow three types of input: A single gene name (variant ID), a list of gene names (variant IDs) or a file including gene names (variant IDs). By clicking on one of the three available input fields (e.g., for gene-search input widgets: “Genename”, “gene_batch” and file2) the respective input type will be stored in a reactive value (e.g. for gene-search: “inputtype$gene”; **Source code 1**).

```
inputtype <- reactiveValues('gene'=c())
    onclick("Genename", (inputtype$gene=c("single")), add=T)
    onclick("gene_batch", (inputtype$gene=c("list")), add=T)
    onclick("gene-upload-section",    (inputtype$gene=c("upload")),
add=T)
```

Source code 1: Storage of last clicked input type of the gene-search in the reactive value “`inputtype$gene`”. Values of “`inputtype$gene`” can be the strings “single”, “list” and “upload”.

Once any input is made the “go” button gets visible, which is done by the “ShowElement” function from the package shinyjs, that can make an element visible. The “go” buttons are included in divisions which are per default invisible and are made visible when the respective input widgets have input and therefore are `TRUE` (see **Source code 2**). The search is started when the “go” button is clicked or the “Enter” key is pressed.

```
observe(

if(isTruthy(input$file2)|isTruthy(input$Genename)|isTruthy(input$ge
ne_batch)){
      shinyjs::showElement(id="gene_go_div")
    }
  )
```

Source code 2: The division "gene_go_div" on the gene-search tab gets visible when any of the three input-option has input.

The search results for genes and variants are slightly different. If a searched gene is included in the KidneyGPS data, i.e., it is located within an eGFR associated locus, an excerpt of the gene prioritisation table (GPS table) is displayed, followed by tables with the underlaying annotation information (Stanzick et al. 2023, [43]). Additional locus-based information is available if the corresponding options are selected by the user. All variant based annotations for the gene can be restricted to variants above a certain posterior probability of association (PPA) threshold (Stanzick et al. 2023, [43]). Programmatically, this is achieved by first querying the annotation data for the searched gene and variants that fulfil the PPA criterion. The results of these searches are stored in reactive values, which are then used to generate the GPS for the specific gene. The GPS section displayed is therefore generated anew with every search.

The GPS tab shows by default the whole GPS table including 12,895 entries for all 5906 genes among 594 signals. Customized prioritisation is achieved by selecting respective filter options. These options are structured in three sections: (1) Signal-filter, (2) Variant-to-gene mapping and (3) Gene-to-phenotype mapping (**Figure 18**). The first section contains filter-options that refer to signal-properties, like credible set size, validation by association with additional biomarkers or association with eGFR sub-phenotypes. The second section includes options, how a credible set variant can be mapped to a gene. The third section refers to gene properties like known kidney phenotypes or drug target information. Applying those filters will restrict the GPS table to genes, that are mapped by the selected properties.

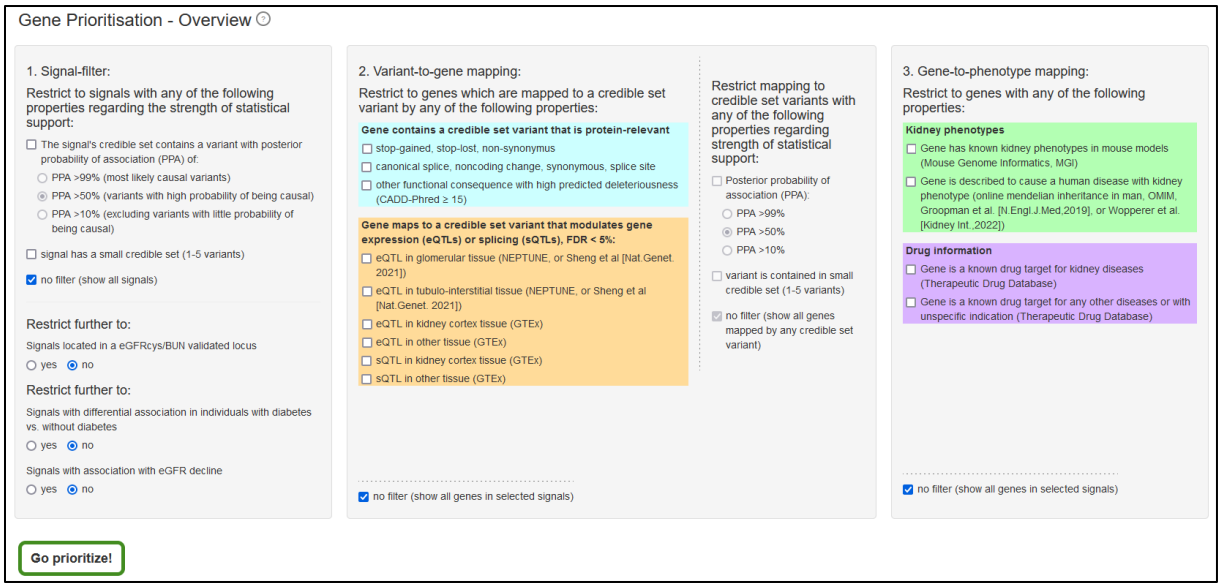


Figure 18: Snapshot of KidneyGPS GPS-tab with the three filter options sections. The first section allows filtering on signal level, the second section on gene and variant level and the third on gene level. The filtering is applied when the "Go prioritize!" button is pressed.

Some options are disabled as long as the superordinate option is not selected. This is achieved by defining the subordinate options in the UI part of KidneyGPS as UIOutput and only defining them more precisely in the server part based on the parent option (**Source code 3, Source code 4**).

```
checkboxInput(inputId = "signalppafilter",
label ="The signal's credible set contains a variant with a postior
probability of association (PPA) of:", value = FALSE),
div(id="gpsppafilter",
uiOutput("signalppafilterdetail")
    )
```

Source code 3: Part of the UI definition of the GPS tab. The first signal-filter option is defined a `checkboxInput` while the corresponding PPA options are only defined as `uiOutput "signalppafilterdetail"`. This works as a placeholder for a changeable UI-part that is defined in the server section.

```
observe({
    if(input$signalppafilter){
      output$signalppafilterdetail <- renderUI(
      radioButtons(inputId = "signalhigh",
                   choiceNames = c("PPA >99% (most likely causal
variants)", "PPA >50% (variants with high probability of being
causal)", "PPA >10% (excluding variants with little probability of
being causal)"),
                    choiceValues = c(0.99, 0.5, 0.1),
                    label =NULL,
                    selected = 0.5)
      )
    }else{
      output$signalppafilterdetail <- renderUI(
        disabled(radioButtons(inputId = "signalhigh",
                   choiceNames = c("PPA >99% (most likely causal
```

```
variants)", "PPA >50% (variants with high probability of being
causal)", "PPA >10% (excluding variants with little probability of
being causal)"),
                    choiceValues = c(0.99, 0.5, 0.1),
                    label =NULL,
                    selected = 0.5))
                  )
    }
})
```

Source code 4: Server definition of "signalppafilterdetail". If the "signalppafilter" option is selected, the UI section "signalppafilterdetail" is rendered as normal radio buttons with the Id "signalhigh", otherwise it is rendered as disabled radio buttons making them greyed out and not selectable for the user.

I have chosen this solution to avoid implausible selections. The same reason applies to the automatic selection and deselection of filter options when non-combinable filters are selected, e.g. the "signal has a small credible set" filter and the "no filter" option in the "Signal-filter" section. This is implemented using the "updateCheckboxinput" function (**Source code 5**).

```
observeEvent(input$signalppafilter,{
    if(!(input$signalppafilter | input$signalsmall)){
      updateCheckboxInput(session, "signalnofilter", value=TRUE,
label = "no filter (show all signals)")
    }
    req(input$signalppafilter)
    updateCheckboxInput(session, "signalnofilter", value=FALSE,
label = "no filter (show all signals)")
  })
```

Source code 5: Example implementation of automatic selection and deselection of the "signalnofilter" option. When the option "The signal's credible set contains a variant with posterior probability of association (PPA) of" referring to "input$signalppafilter" is changed by the user this is registered by the app via the "observeEvent" function. It is then checked whether option "signalppafilter" or option "signalsmall" is still selected. If not, option "signalnofilter" is automatically selected. If option "signalppafilter" is selected (and not deselected), option "signalnofilter" is deselected.

Selection of the filter options has no effect on the GPS table unless the "Go prioritize!" button is clicked. The first progress triggered by this is a consistency check of the filter options. While the "no filter" options of all three sections are automatically selected if the other filter options are deselected by the user, it is also possible for the user to deselect the "no filter" option while no other filter options are selected. Since the intention to deselect the "no filter" option is not clear, the other filter options are not automatically selected. Instead, a reactive value is implemented in the KidneyGPS code, which only becomes TRUE if one of the filter options or the "no filter" option is selected. If the reactive value is FALSE and the "Go prioritize!" button pressed, a pop-up dialog appears informing the user that there are missing selections. If the consistency check was successful, the selected filter options are applied to the GPS table by processing the following steps: (1) The rows of the GPS table meeting the

signal filter criteria are stored in a reactive value. (2) The variant-to-gene mapping options regarding the annotation are stored as the columns of the GPS table containing the respective information. (3) All three GPS table versions stored in the R workspace (full GPS version, version only containing information of credible set variants with PPA >10% and version only containing information of credible set variants with PPA >50%) are filtered to the rows identified in step one. (4) The three versions of step 3 are further filtered on rows whose entries in any of the columns of step 2 are non-zero. In addition, the PPA and small set options of the second filter section (variant-to-gene mapping) are applied. If only credible set variants with PPA > 10 % or > 50 % are of interest, the full GPS version is set to `NULL`, while only the respective GPS version is filtered on the rows with non-zero columns. In the case that only small sets or variants with PPA >99% or a combination of both options are of interest the GPS_10 and GPS_50 version are set to `NULL` and the full GPS version is filtered. In any other filter combination, the full GPS version and one of the other two versions are filtered to rows with non-zero columns. (5) In case of two filtered versions from step 4, these are combined and duplicates are deleted. (6) If Gene-to-phenotype mapping options are selected, the GPS table is filtered to rows with non-zero entries for any of the selected options. If only one drug information option is selected, the respective entry in the GPS table must match the selected option, e.g. if the option “gene is a known drug target for kidney diseases” is selected only entries with "yes for kidney disease" will be counted as non-zero. (7) The final GPS table version is displayed and the number of unique gene- and signal-names counted and provided to the user.

Documentation and help

The Documentation & Help tab of KindeyGPS provides the user with information on release history, data sources, privacy and security statements, citation and contact information, and user guides for the three search pages and the GPS tab. A step-by-step guide for the usage of the filter options of the GPS tab is provided as a pdf file which also includes three prioritisation examples.

3.2.2.3. Styling KidneyGPS with Cascading Style Sheets (CSS)

While RShinys function “renderDataTable” renders basic tables, most table headers in KidneyGPS are created manually to allow double-row headers and to display

additional information when hovering over them. This is done by following the HTML logic of headers that allows to define multiple table rows inside the table header and to specify the size of each cell, including the spanning of multiple rows or columns. Text displayed when a table header cell is hovered is also defined in the table headers (**Source code 6**). The respective cells have two classes assigned with one being a unique class the CSS sheet uses to control the visibility of the respective extra text. The other class is “hover” to mark all objects that have this feature. The additional information is labelled as paragraph or span inside the cell and has also a class assigned. When the respective header-cell is hovered, the additional information gets visible inside an extra field outside of the original cell. The exact position is defined in the CSS sheet (**Source code 7**).

Additional styling information provided by the CSS sheet (see appendix) include the colouring of objects with the classes “CADD”, “eqtl”, “pheno” or “drug” and specific table header cells, a changed positioning of the input objects on the GPS-tab when a small screen (below 768px) is used, and the whole appearance of the question buttons with the class “question”.

```
sketchcredvar = htmltools::withTags(table(
    class = 'display',
    thead(
      tr(
        th(rowspan=2, 'RSID*'),
        th(rowspan=2,'Locus ID'),
        th(rowspan=2,class="Signalidwithcomment hover",'Signal ID',
           span(class="Signalidcomment", "ID of independent signals
                within a locus")),
        th(rowspan=2,class="PPA hover", 'PPA',
           p(class="PPAcomment",
           "Posterior probability of association of the SNP
           (probability of driving the association of the
            respective
            signal)")),
        th(rowspan=2,'N'),
        th(rowspan=2,'Effect allele'),
        th(rowspan=2,'Other allele'),
        th(rowspan=2,'EAF'),
        th(colspan=3, 'Association with log(eGFRcrea) conditioned
        on
           other signal index variants'),
        th(colspan=3, 'Association with log(eGFRcrea)
        unconditioned'),
        th(rowspan=2,'Chr'),
        th(rowspan=2, class="pos hover", 'Pos',
           span(class="poscomment","SNP position from GRCh37")),
      ),
      tr(
        lapply(c('beta', 'StdErr', 'P-value','beta', 'StdErr',
                 'P-value'), th)
      )
    )
 ))
```

Source code 6: Definition of the table header of the credible set variants table on the "Variants" tab. The header is stored in the object "`sketchcredvar`". The header contains two rows and 16 columns. Ten cells span two rows (`rowspan=2`) while two cells of the first row each span 3 columns (`colspan=3`). The second row is built by using the "`lapply`"-function to apply the "`th`"-definition to all cells. Class definitions are used to make cells available for styling with CSS including the hover behaviour.

```
.Geneofficial, .Locusidwithcomment, .Signalidwithcomment,
.validationwithcomment, .DMwithcomment,
.declinewithcomment,.eQTLglowithcomment,.eQTLtubwithcomment, .PPA,
.REF, .ALT, .CADDscore, .pos, .snpsearch, .maxPPA{

  position: relative;
  text-decoration: underline grey dotted;
}

.Genecomment, .Locusidcomment, .Signalidcomment,
.validationcomment, .DMcomment, .declinecomment, .PPAcomment,
.REFcomment, .ALTcomment, .CADDscorecomment, .poscomment,
.snpsearchcomment, .maxPPAcomment{
  display:none;
  position:absolute;
  z-index:1000000;
  border:1px;
  background-color:white;
  border-style:solid;
  border-width:1px;
  border-color:grey;
  padding:4px 4px;
  border-radius: 6px;
  font-weight: normal;
  color:black;
  left: 105%;
  transform: translate(0, -60%);
  text-align: center;
}

.Signalidwithcomment:hover span.Signalidcomment{
  display:block;
  width: 150px;
  min-width: 100%;
  max-height: 95%;
}
```

Source code 7: Parts of the CSS sheet used to style KidneyGPS. Three styling commands are shown: (1) Elements with classes "geneofficial", "Locusidwithcomment", "Signalidwithcomment", "validationwithcomment", "DMwithcomment", "declinewithcommen"t, "eQTLglowithcomment", "eQTLtubwithcomment", "PPA", "REF", "ALT", "CADDscore", "pos", "snpsearch", and "maxPPA" are styled with a grey dotted underline to make it recognizable as text with additional information when hovered. (2) The elements that contain additional information with the classes "Genecomment", "Locusidcomment", "Signalidcomment", "validationcomment", "DMcomment", "declinecomment", "PPAcomment", "REFcomment", "ALTcomment", "CADDscorecomment", "poscomment", "snpsearchcomment" and "maxPPAcomment" are styled with a grey border and are moved 60% to the right from their original position. (3) The "span"-object with class "Signalidcomment" is displayed when an object with the class "Signalidwithcomment" is hovered.

3.2.2.4. Summary of code availability and technical requirements

The source code of KidneyGPS is available at https://github.com/kjstanzick/KidneyGPS so that it can be extended or adapted to other phenotypes by other scientists. This includes the run_before_app_server.r script, the app.r script, the R-workspace, the CSS sheet and the script to load the required R-packages. These packages must be installed prior to the run of the application and include the packages: “data.table”, “shiny”, “shinyBS”, “DT”, “shinyjs”, “shinyWidgets”, “htmltools” and “writexl”. All can be directly installed from the CRAN (Comprehensive R Archive Network) repository. Downloaded from the Github repository each user can run a local version of KidneyGPS. The RShiny web-server hosting KidneyGPS is operated by the University of Regensburg.

3.2.2.5. Prioritisation example

Customizable gene prioritisation is the most important function of KidneyGPS. Thus, I will provide two examples of how to use the GPS tab for prioritisation:

(1) The recovery of kidney function may be the ultimate goal of drug development. However, halting the progressive loss of kidney function is equally important to prevent patients from needing dialysis. Targets for such drugs might be genes (and their proteins) associated with eGFR decline. Thus, for the first prioritisation example the signal filter is set to signals with association with eGFR decline and signals with a small credible set size to have a manageable number of SNPs for potential laboratory analyses. The variant-to-gene mapping is set to “stop-gained, stop-lost, non-synonymous”, “eQTL in glomerular tissue”, “eQTL in tubulo-interstitial tissue”, “eQTL in kidney cortex tissue” and “sQTL in kidney cortex tissue” with no restriction of the variants by PPA. This results in 4 genes: *GP2* (signal k2.1), *UMOD* (signal k2.1), *GALNTL5* (signal k4.1) and *TPPP* (signal k88.1). **Table 12** shows an excerpt of the downloaded prioritisation results from KidneyGPS. All respective signals are the first signal of their locus and validated by association with eGFR estimated from serum cystatin C (eGFRcys). The signals of *GP2*, *UMOD* and *GALNTL5* are additionally validated by association with blood urea nitrogen (BUN). The signal index variant or a highly correlated variant of the signals k2.1 and k88.1 show stronger effects on eGFR in persons with diabetes mellitus than without. All genes are prioritised by an eQTL in

glomerular (*GP2*, *UMOD* and *TPPP*) or tubulo-interstitial tissue (*GP2*, *UMOD*, *GALNTL5* and *TPPP*). Only for the *UMOD* gene there are known kidney phenotypes in mice and human. None of the genes is a known drug target for any disease.

For more detailed information on the eQTLs all genes are searched in KidneyGPS gene tab. *UMOD* and *GP2* are located in the same locus and are prioritised by the same variant: rs77924615. This is an eQTL in tubulo-interstitial and glomerular tissue and the eGFR lowering allele (effect allele frequency, EAF: 80%) has an upregulating effect on both genes with a smaller p-value for the *UMOD* gene. *GALNTL5* is prioritised because of the upregulating effect in tubulo-interstitial tissue of 2 out of the 3 credible set variants of the respective signal: rs10224002 (EAF: 28%, PPA 80%) and rs6464165 (EAF: 28%, PPA 1%). The two credible set variants of signal k88.1 (rs440225, EAF 28%, PPA 10%; rs434215, EAF 28%, PPA 90%) are both eQTLs for TPPP in glomerular and tubulo-interstitial tissue with an upregulating effect of the eGFR lowering allele.

(2) The second example focuses on protein-altering variants in genes that are known to cause mendelian diseases with kidney phenotype in humans. Therefore, the signal-filter is set to signals with small credible set, the “stop-gained, stop-lost, non-synonymous” option of the variant-to-gene mapping is selected, and the filter “Gene is described to cause a human disease with kidney phenotype” of the gene-to-phenotype mapping section is applied. This also results in 4 prioritised genes (**Table 13**): *CACNA1S* (signal k48.1), *POR* (signal k86.1), *PKHD1* (signal k128.2) and *NPHS1* (signal n24.1). All signals are either validated by association with eGFRcys or BUN. None of the signals is associated with eGFR decline or has diabetes mellitus depending effect on eGFR. Besides the protein-altering variants no eQTLs or sQTLs in kidney tissue have been found for the four genes. A search of those genes in KidneyGPS reveals the mapped diseases: *CACNA1S* causes Hypokalemic periodic paralysis, *NPHS1* Nephrotic Syndrome, *PKHD1* Polycystic kidney disease 4, and *POR* Congenital adrenal hyperplasia due to cytochrome P450 oxidoreductase deficiency and Antley-Bixler Syndrome with Genital Anomalies and Disordered Steroidogenesis. Additionally, the gene search revealed that POR is a terminated/discounted drug target for Psoriatic disorder, Pruritus and Dermatitis, while CACNA1S is an investigative target of different drugs without a known indication.

Table 12: First prioritisation example. Excerpt of the downloaded prioritisation results from KidneyGPS with filtering for eGFR decline associated signals with small credible set containing variants which are protein altering or eQTL or sQTL in kidney tissue. For better visualization 8 columns are omitted.

Gene	Locus name	Locus ID	Signal ID	eGFRcys or BUN validation	DM effect	Decline effect	# credible variants in signal	stop-gained, stop-lost, non-synonymus	eQTL glomerulus	eQTL tubulo-interstitium	eQTL kidney cortex	sQTL kidney cortex	Coloc in NEPTUNE tissue	# kidney phenotypes in mouse	# kidney phenotypes in human	known_drug_target
GP2	[UMOD-PDILT]	k2	Signal 1	cys/bun	DM>NoDM	decline	1	0	1	1	0	0	-	0	0	no
UMOD	[UMOD-PDILT]	k2	Signal 1	cys/bun	DM>NoDM	decline	1	0	1	1	0	0	-	25	2	no
GALNTL5	[PRKAG2]	k4	Signal 1	cys/bun	-	decline	3	0	0	2	0	0	tubulo-interstitial	0	0	no
TPPP	[TPPP]	k88	Signal 1	cys	DM>NoDM	decline	2	0	2	2	0	0	tubulo-interstitial	0	0	no

Table 13: Second prioritisation example. Excerpt of the downloaded prioritisation results from KidneyGPS with filtering for signals with small credible sets containing protein-altering variants of genes that cause a mendelian disease with kidney phenotype in human. For better visualization 8 columns are omitted.

Gene	Locus name	Locus ID	Signal ID	eGFRcys or BUN validation	DM effect	Decline effect	# credible variants in signal	stop-gained, stop-lost, non-synonymus	eQTL glomerulus	eQTL tubulo-interstitium	eQTL kidney cortex	sQTL kidney cortex	Coloc in NEPTUNE tissue	# kidney phenotypes in mouse	# kidney phenotypes in human	known_drug_target
CACNA1S	[CACNA1S]	k48	Signal 1	bun	-	-	1	1	0	0	0	0	-	0	1	yes for other disease
POR	[POR]	k86	Signal 1	cys/bun	-	-	4	1	0	0	0	0	-	1	1	yes for other disease
PKHD1	[PKHD1]	k128	Signal 2	cys	-	-	1	1	0	0	0	0	-	22	1	no
NPHS1	[NPHS1]	n24	Signal 1	cys/bun	-	-	2	1	0	0	0	0	-	23	1	no

4. Discussion

Pitfalls and limitations of fine-mapping GWAS results

Loss of kidney function is a major global health burden and genome wide association studies (GWAS) can help to identify novel drug targets. Therefore, it is important to identify the causal SNPs for the association to understand the underlying biological processes. Fine-mapping aims to identify those SNPs by providing a statistical likelihood for each SNP to be the causal SNP in its signal. Different methods exist for fine-mapping. One is the combination of using GCTA to identify independent signals and the Wakefield-method which calculates approximate bayes factors (ABF) and posterior probabilities of association (PPA) for each variant and signal based on the association statistics. It is often discussed that the use of GCTA and other fine-mapping methods strongly depends on a matching reference panel. However, this work shows that there is another factor that strongly impacts GCTA and the following fine-mapping results: the number of decimal digits used for effect estimates and standard errors of GWAS results which are the input of GCTA. To improve the power of GWAS by increasing sample size, often the results of multiple GWAS are meta-analyses. A widely used tool to generate GWAS meta-analyses (GWAMA) is METAL. Standard settings of METAL store GWAMA results with 4 decimal digits for effect estimate and standard error while calculating p-values internally with the precision provided by the input files. This means, that for traits with very small effects, like log-transformed phenotypes as investigated in this work, the effect estimates and standard errors might be rounded to only one or two telling digits. GCTA can work with this data, but results can lack stability. This work shows that at least 4 telling digits are necessary to achieve robust signal identification and fine-mapping results with those methods. In the case of log-transformed eGFR this means 6 and 7 decimal digits for effect estimate and standard error. However, only 15 of the original 210 secondary signals for eGFR were suspicious when the original analysis was rebuilt with more decimal digits. Completely new conditioning with 4 telling digits for effect estimate and standard error revealed only 170 secondary signals of which 118 were correlated to one of the previously reported signals. In total, 35,885 credible set variants were identified in this work and ultimately integrated in KidneyGPS.

However, some limitations of the methods and data remain. While the original GWAMA for eGFR was a trans-ethnic meta-analysis, fine-mapping is still based on the

European-ancestry subset due to a lack of an appropriate trans-ethnic reference panel. This results in loci, where not a single variant is genome-wide associated due to a lack of power because of the reduced sample size and due to other allele frequencies in non-European ancestry. Even with a sufficient number of telling digits it was not possible to fine-map any of those 51 signals down to 5 or less SNPs. A matching trans-ethnic reference panel might help to better fine-map such signals. Additionally, the use of trans-ethnic data in GWAMA and reference panel can help to dissect regions of jointly inherited variants (variants with high LD), that are now a problem in fine-mapping. Due to their high LD these variants have nearly identical association statistics and ABFs. This leads to large credible sets with over 100 or even 1000 SNPs, all of which having small PPAs. Besides, the integration of more non-European data opens up the possibility of globalizing the results.

Another limitation of the Wakefield method is the assumption of a single causal SNP per signal making it strongly dependent on signal identification. Alternative fine-mapping methods like SuSiE or FINEMAP circumvent this problem by simultaneously assuming several causal variants in a region. However, these methods are still strongly dependent on matching reference panels and well powered GWAS data [30].

KidneyGPS in the context of other GWAS platforms

Statistical fine-mapping alone is not sufficient to prioritise genes for laboratory testing. Functional annotation is required to assign the prioritised SNPs to genes they may influence. Linkage to additional information about the genes such as known phenotypes in mice or humans is also helpful to prioritise genes. However, for eGFR this data includes information on 35,885 credible set variants and 5906 genes located in eGFR associated loci. KidneyGPS was developed to make this amount of data available to kidney researchers.

As recently discussed in Stanzick et al. 2023, other comprehensive GWAS platforms exist that aim to present GWAS and post-GWAS data accessible to non-GWAS researchers [43]. Three major tools are GWAS Catalog [19], HugeAMP [60] and Open Targets Genetics [61], which, unlike KidneyGPS, present information for various traits and diseases and have no focus on kidney function [43]. All platforms have in common that they rely on the cooperation of researchers to keep their database up to date. As a result, not all study results can always be found on every platform. While KidneyGPS is based exclusively on the data from Stanzick et al. 2021 [25] and the renewed fine-

mapping presented here [43], a newer GWAMA for eGFR [24] can be found in the GWAS Catalog and HugeAMP. However, the most recent GWAMA in Open Targets Genetics regarding eGFR is from 2019 [35]. Only KidneyGPS and HugeAMP provide GWAS information for the more specialized traits for kidney function like eGFR decline [58] or eGFR associations stratified by diabetes status [57]. GWAS Catalog focuses on collecting and providing GWAS summary statistics, thus there is no feature for gene prioritisation and only a limited number of annotation information provided. The other platforms, including KidneyGPS, mainly focus on the annotation and prioritisation aspect. Open Targets Genetics and HugeAMP provide machine learning derived gene scores, while KidneyGPS allows transparent prioritisation with various filtering options. In summary, all platforms have their advantages and unique features. While HugeAMP might be the most comprehensive platform, it can be difficult to extract relevant information due to the amount of data presented on different sites. On the other hand, KidneyGPS is the platform with the most reduced data, but its focus on kidney function regarding aspects makes it easier to the user to extract relevant information.

Prioritised genes

Eight genes in 7 loci were prioritised in the two example searches using KidneyGPS. 4 of the 8 genes (*POR*, *CACNA1S*, *PKHD1* and *NPHS1*) were prioritised by a common missense variant, with the eGFR-lowering allele being the more common allele in all 4 variants. Additional database queries revealed “benign” prediction by Variant Effect Predictor [54] and no structural damage in the 3D model prediction (Supplementary Figure 4, Supplementary Figure 5, Supplementary Figure 6) [62]. However, the following is known for these 4 genes regarding their role for kidney function:

The *POR* gene encodes for the Cytochrome P450 Oxidoreductase which is the only protein donating electrons to CYP450 enzymes and is therefore relevant for drug metabolism [63]. In the context of kidney function the minor allele of rs1057868 (A503V) is associated with better kidney function [43]. The same mutation is described to have different effects on various CYP enzymes which have effects on steroid levels and drug clearance [64]. One drug with higher clearance in CC homozygotes which express CYP3A5 is tacrolimus [65]. Tacrolimus is immunosuppressive and commonly used after kidney transplantation to minimize the risks of graft rejection and death [66].

The voltage-dependent L-type calcium channel subunit alpha-1S encoded by *CACNA1S* is specifically expressed in skeletal muscle tissue [67, 68]. Thus, the

association with eGFR is possibly mediated by creatinine production [69]. However, calcium channel blockers are used in treatment of hypertension and are discussed to have reno-protective effects. Conflicting data is reported suggesting differential effects of L-type or T/L-type calcium channel blocking because of the different localization in efferent and afferent arterioles [70, 71].

PKHD1 is directly expressed in the kidney and encodes for fibrocystin/polyductin, which is a transmembrane protein found in cells of the loop of Henle and adjacent parts of the nephron as well as the collecting duct [72, 73]. Fibrocystin plays a crucial role in the control of cytoskeletal forces and adhesion signalling and thus epithelial morphogenesis [74]. Mutations of *PKHD1* can lead to autosomal recessive polycystic kidney disease (ARPKD), a ciliopathy characterized by dilatation of the renal collecting duct, which can already be present in utero [75, 76]. Most patients suffer from impaired renal function ranging from mild insufficiency to end stage renal disease (ESDR) [75]. According to ClinVar, the missense mutation which prioritised *PKHD1* in this work (R3842L) is likely benign and not causative for polycystic kidney disease [77, 78].

The *NPHS1* gene encodes for the protein nephrin which is an essential component of the slit diaphragm (SD) between podocytes [79]. It is therefore important for the proper function of the glomerular filtration barrier. Various mutations in *NPHS1* are known to cause Nephrotic Syndrome Type 1 also called congenital nephrotic syndrome [80]. The disease is characterized by heavy proteinuria and nephrosis soon after birth caused by structural abnormalities of the SD [81, 82]. Nephrin interacts with P-cadherin, ZO-1, fat and the kidney specific proteins nephrin-like 1 and podocin to form the SD [82–87]. While rare genetic kidney diseases are also described for nephrin-like1 and podocin mutations [88] both genes are not included in any eGFR associated locus [25]. In contrast, the *CDH3* gene, which codes for P-cadherin, is located in an eGFR associated locus (Locus k90) [25]. Although there is no further evidence for *CDH3* in KidneyGPs, the locus generally lacks a potentially causal gene [43].

Four of the eight genes from the prioritisation example were prioritised by an eQTL in kidney tissue. As a new meta-analysis for kidney eQTL data was released following the development of KidneyGPS version 2.3.1, this dataset was interrogated for the three prioritised SNPs to confirm associations with expression of the prioritised genes [24]. The two SNPs rs77924615 and rs434215, which were described as eQTLs by Sheng et al. previously [89], were confirmed in their effect on the expression of *UMOD*

and *GP2*, and *TPPP* [24]. For the third SNP rs10224002 no association with expression of *GALNTL5* was found in the novel dataset of Liu et al [24]. This may be due to the fact that there is no association or that the SNP is not included in the analyses by Liu et al. [24]. However, another SNP of the same small credible set (rs10254101, PPA 19%) is now described as an eQTL in kidney tissue for *PRKAG2* [24], so *PRKAG2* would meet the criteria for prioritisation in this example.

GALNTL5, the gene encoding for polypeptide N-acetylgalactosaminyltransferase-like protein 5, is highly expressed in the testis and at low level at the cerebral cortex [67]. Besides the association of the eQTL rs10224002 with eGFR no link to kidney function is described in literature [58, 25, 90]. However, a heterozygous mutation of *GALNTL5* is described to affect male infertility [91, 92]. In contrast, *PRKAG2*, encoding for Protein kinase AMP-activated non-catalytic subunit gamma 2, is expressed in various tissues including the kidney [67]. Mutations in *PRKAG2* are mostly described in the context of cardiac diseases, but a case report links a mutation of *PRKAG2* also with Immunoglobulin A nephropathy via the AMP-activated protein kinase (AMPK) of which PRKAG2 is a subunit [51, 93]. The conserved energy sensor AMPK is reported to be part of many physiological and pathological processes in the kidney including sodium and general ion transport, and fibrosis [94, 95]. In addition, the type 2 diabetes drug metformin which is also described to be beneficial for different kidney diseases is an activator of AMPK [96, 97]. Interestingly, two other subunits of AMPK (PRKAA2 and PRKAG3) are located in eGFR associated loci but with no further evidence linking them to the associated SNPs [25, 43].

The *TPPP* gene encoding for tubulin polymerization promoting protein is mostly expressed in the brain but also at lower levels in the kidney [67, 73]. It is a microtubule associated protein playing a role in microtubule polymerization and stabilization and microtubule dynamics have an impact on kidney cell regeneration after ischemic injuries [98, 99]. However, a specific role of *TPPP* in microtubules of renal cells is not described. Additionally, another gene (*SLC9A3*) in the same locus is described as drug target for “Tenapanor”, an approved drug for Irritable bowl syndrome and phase 2 drug for kidney disease [43, 55]. Expression of this gene in other tissues than kidney is influenced by credible set variants of the first and second signal of this locus [43].

The expression of both genes, *UMOD* (encoding for uromodulin) and *GP2* (encoding for glycoprotein 2) is influenced by the same SNP rs77924615 in the same direction

[25]. Both genes are expressed by cells of the thick ascending limb in the kidney [73]. However, *GP2* is more strongly expressed in the pancreas than the kidney, while *UMOD* is mostly expressed in kidney cells [67]. The two genes are described to be paralogs due to their sequence and structural similarities [100]. It is therefore assumed that glycoprotein 2 fulfils a similar function to uromodulin: Protection against infection by binding bacteria, especially *Escherichia coli*, to the highly glycolyzed proteins [100, 101]. Additionally, *UMOD* impacts the renal sodium transport via activity regulation of NKCC2 and ROMK, as well as calcium and magnesium reabsorption by effecting the endocytosis of calcium and magnesium channels in the distal convoluted tubule [101–104]. Because of its higher expression in the kidney and the fact that multiple autosomal dominant mutations causing autosomal dominant tubule-interstitial kidney disease are described for *UMOD*, it is more likely that the altered expression of *UMOD* is causal for the association with eGFR in this locus.

The above examples show how potential kidney-relevant genes can be prioritised with KidneyGPS among hundreds of other genes located in the same eGFR associated loci. Nevertheless, for many loci/signals no potential causal gene can be prioritised at this stage of KidneyGPS.

5. Outlook

The here presented version 2.3.1 of KidneyGPS can be extended with existing and upcoming data potentially resulting in an improved understanding of the genetic effects on kidney function. On the GWAS level, the upcoming CKDGen analysis round 5 data for kidney function biomarker (including eGFR) will improve statistical power to identify novel and to improve resolution of known loci and signals by increasing sample size. In addition, multi-trait GWAS will help to identify loci affecting more than one aspect of kidney function. This new GWAS data will offer the possibility to improve the fine-mapping with the conventional Wakefield method and also provide the statistical power for other fine-mapping methods with more complex approaches, so that the statistical prioritization of variants is enhanced. For the functional annotation of variants, updates of eQTL sources with larger sample size will help to close the “power-gap” between GWAS and eQTL data and cell-type specific data will help to distinguish functional effects of eQTLs [24]. Updates of CADD and VEP are available and should be used to assess functional consequences. In addition to these sources for variant annotation,

new sources can be included to capture additional functional aspects of variants, e.g. their impact on chromatin status. Ultimately, this will lead to more functionally annotated variants. On gene level, the sources MGI and OMIM are regularly updated with novel mouse model results and human case reports. Including this data in upcoming versions of KidneyGPS might help connect causal genes for monogenic diseases with complex forms of kidney diseases. Also interrogating data of gene burden tests [105] will further improve the level of gene-prioritisation.

6. Summary

Impaired kidney function represents a major global health burden. However, therapeutic options are still limited. The understanding of kidney physiology and its influencing genetic factors is important for the development of new therapeutic targets. Genome-wide association studies (GWAS) can highlight the connections between complex traits and genetic polymorphisms. Nevertheless, GWAS do not identify single causal variants, but regions of highly correlated variants, which are all statistically associated with the trait. Thus, additional statistical and bioinformatic analyses are necessary to determine likely causal polymorphisms and genes.

One sign of impaired kidney function is a reduced glomerular filtration rate (GFR), which can be estimated from the blood serum biomarker creatinine (eGFR). Previous GWAS meta-analyses identified 424 genetic loci and 634 independent signals associated with eGFR. In this thesis, this GWAS meta-analysis was replicated with more decimal digits for effect estimate and standard error than the standard software provides by default to analyse the impact on the identification of independent signals and the subsequent statistical fine-mapping to determine the most likely causal variants. Secondary signal identification with 6 and 7 decimal digits for effect estimate and standard error instead of 4 revealed 52 novel independent signals while 542 signals were replicated. The number of credible set variants, which are the result of fine-mapping, was reduced from 38,306 to 35,885. A replication with 8 decimal digits showed no relevant changes, which implies 4 telling digits to be sufficient for stable secondary signal identification and fine-mapping.

The credible set variants from the statistical fine-mapping pinpoint to the effector variants which may act on the causal genes. Therefore, publicly available functional

and regulatory data from Combined Annotation Dependent Depletion (CADD) database, NEPTUNE study, Susztak lab and GTEx platform was used to annotate the credible set variants. Data from Mouse Genome Informatics (MGI), Online Mendelian Inheritance in Men (OMIM), Groopman et al., Wopperer et al. and Therapeutic Drug Database was used to identify genes in eGFR associated loci that are already described in the context of kidney function or drug development.

KidneyGPS was developed as a web application based on this comprehensive data. It enables easy access to the statistical and functional data for individual variants, genes or genetic regions. In addition, various filters can be used to prioritize genes that could be suitable as candidates for further functional studies. In two prioritization examples, the 8 genes *UMOD*, *GP2*, *GALNTL5*, *TPPP*, *CACNA1S*, *POR*, *PKHD1* and *NPHS1* were highlighted. KidneyGPS is available under https://kidneygps.ur.de/gps/ .

7. Zusammenfassung

Die Einschränkung und ultimativ der Verlust der Nierenfunktion bei Millionen von Menschen sind ein weltweites Gesundheitsproblem mit nur wenigen Therapie-Optionen. Für die Entdeckung neuer Therapie-Ansätze sind ein tiefgreifendes Verständnis der Nieren-Physiologie und der zugrunde liegenden genetischen Einflussfaktoren von großer Bedeutung. Mit genom-weiten Assoziationsstudien (GWAS) wird nach den Zusammenhängen zwischen natürlichen genetischen Polymorphismen und komplexen Erkrankungen gesucht. Das Ergebnis sind jedoch keine einzelnen, kausalen Varianten, sondern genomische Regionen mit hochgradig korrelierten Varianten, welche allesamt statistisch signifikante Assoziationen zeigen. Somit sind weitere statistische und bioinformatorische Analysen notwendig, um mögliche kausale Varianten und Gene ausfindig zu machen.

Ein Anzeichen einer verringerten Nierenfunktion ist eine reduzierte glomeruläre Filtrationsrate (GFR), welche leicht basierend auf dem Blutserum-Marker Kreatinine geschätzt werden kann (eGFR). Frühere GWAS Meta-Analysen haben 424 genetische Regionen mit insgesamt 634 unabhängigen Signalen identifiziert, die eine Assoziation mit eGFR zeigen. In dieser Arbeit wurde die frühere Meta-Analyse mit einer erhöhten Anzahl an Dezimalstellen als üblicherweise verwendet repliziert, um den Einfluss auf die Identifizierung unabhängiger Signale und die statische Feinkartierung zur Bestimmung möglicher kausaler Varianten zu untersuchen. Die Verwendung von 6 bzw. 7 Dezimalstellen für Effektschätzer und Standardfehler anstelle der üblichen 4 Dezimalstellen führte zur Identifizierung von 53 neuen unabhängigen Signalen und der Replikation von 542 der zuvor beschriebenen Signale. Die Anzahl statische wahrscheinlicher kausaler Varianten reduzierte sich von 38,306 auf 35,885. Eine weitere Erhöhung der Dezimalstellen auf 8 zeigte keinen nennenswerten Einfluss, sodass 4 informative Dezimalstellen als ausreichend erachtet werden können für die stabile Identifizierung von unabhängigen Signalen und die statistische Feinkartierung dieser.

Unter den statistisch wahrscheinlich kausalen Varianten befinden sich Polymorphismen, die einen funktionellen bzw. regulatorischen Einfluss auf in der Nähe befindliche Gene haben. Um diese ausfindig zu machen, wurden die öffentlich zugänglichen Datenbanken *Combined Annotation Depletion (CADD)*, *NephQTL (NEPTUNE)*, *Susztaklab* und *GTEx* verwendet. Um zusätzlich Gene in den assoziierten

genomischen Bereichen hervorzuheben, die bereits im Kontext der Nierenfunktion oder im Zusammenhang mit Medikamentenentwicklung beschrieben wurden, wurden weitere öffentliche Datenquellen verwendet (*Mouse Genome Informatics (MGI), Online Mendelian Inheritance in Men (OMIM)* und *Therapeutic Drug Database,* sowie die Publikationen *Groopman et al.* und *Wopperer et al.)*

Die Web-basierte Anwendung KidneyGPS wurde im Rahmen dieser Arbeit entwickelt. Sie ermöglicht einfachen Zugriff auf die umfangreichen Daten zu Varianten, Genen und genomischen Regionen, die während der GWAS-Meta-Analyse für eGFR, Feinkartierung und Annotation entstanden. Eine ihrer wichtigsten Funktionen ist die Priorisierung von Genen mit Hilfe diverser Filter-Optionen, welche mögliche Kandidaten für funktionelle Studien darstellen. In zwei Beispielen wurden die 8 Gene *UMOD*, *GP2*, *GALNTL5*, *TPPP*, *CACNA1S*, *POR*, *PKHD1* und *NPHS1* priorisiert. KidneyGPS ist unter https://kidneygps.ur.de/gps/ verfügbar.

References

1. Brandes R, Lang F, Schmidt RF (eds) (2020) Physiologie des Menschen: Mit Pathophysiologie, 32. Auflage. Springer-Lehrbuch. Springer, Berlin
2. (2024) KDIGO 2024 Clinical Practice Guideline for the Evaluation and Management of Chronic Kidney Disease. Kidney International 105:S117-S314. https://doi.org/10.1016/j.kint.2023.10.018
3. PEASE DC (1955) Fine structures of the kidney seen by electron microscopy. J Histochem Cytochem 3:295–308. https://doi.org/10.1177/3.4.295
4. Bertram JF, Douglas-Denton RN, Diouf B et al. (2011) Human nephron number: implications for health and disease. Pediatr Nephrol 26:1529–1533. https://doi.org/10.1007/s00467-011-1843-8
5. Renkin EM, Robinson RR (1974) Glomerular filtration. N Engl J Med 290:785–792. https://doi.org/10.1056/NEJM197404042901408
6. Mundel P, Kriz W (1995) Structure and function of podocytes: an update. Anat Embryol (Berl) 192:385–397. https://doi.org/10.1007/BF00240371
7. Girndt M, Trocchi P, Scheidt-Nave C et al. (2016) The Prevalence of Renal Failure. Results from the German Health Interview and Examination Survey for Adults, 2008-2011 (DEGS1), vol 113. Robert Koch-Institut, Epidemiologie und Gesundheitsberichterstattung
8. Mills KT, Xu Y, Zhang W et al. (2015) A systematic analysis of worldwide population-based data on the global burden of chronic kidney disease in 2010. Kidney International 88:950–957. https://doi.org/10.1038/ki.2015.230
9. Keith DS, Nichols GA, Gullion CM et al. (2004) Longitudinal follow-up and outcomes among a population with chronic kidney disease in a large managed care organization. Arch Intern Med 164:659–663. https://doi.org/10.1001/archinte.164.6.659
10. (2002) K/DOQI clinical practice guidelines for chronic kidney disease: evaluation, classification, and stratification. Am J Kidney Dis 39:S1-266
11. Fox CS, Larson MG, Leip EP et al. (2004) Predictors of new-onset kidney disease in a community-based population. JAMA 291:844–850. https://doi.org/10.1001/jama.291.7.844
12. Shi Z, Taylor AW, Riley M et al. (2018) Association between dietary patterns, cadmium intake and chronic kidney disease among adults. Clin Nutr 37:276–284. https://doi.org/10.1016/j.clnu.2016.12.025
13. Lin J-L, Lin-Tan D-T, Hsu K-H et al. (2003) Environmental lead exposure and progression of chronic renal diseases in patients without diabetes. N Engl J Med 348:277–286. https://doi.org/10.1056/NEJMoa021672
14. Levin A, Tonelli M, Bonventre J et al. (2017) Global kidney health 2017 and beyond: a roadmap for closing gaps in care, research, and policy. The Lancet 390:1888–1917. https://doi.org/10.1016/S0140-6736(17)30788-2
15. Uffelmann E, Huang QQ, Munung NS et al. (2021) Genome-wide association studies. Nat Rev Methods Primers 1:1–21. https://doi.org/10.1038/s43586-021-00056-9
16. Auton A, Abecasis GR, Altshuler DM et al. (2015) A global reference for human genetic variation. Nature 526:68–74. https://doi.org/10.1038/nature15393
17. Cordell HJ, Clayton DG (2005) Genetic association studies. Lancet 366:1121–1131. https://doi.org/10.1016/S0140-6736(05)67424-7
18. Klein RJ, Zeiss C, Chew EY et al. (2005) Complement factor H polymorphism in age-related macular degeneration. Science 308:385–389. https://doi.org/10.1126/science.1109557
19. Sollis E, Mosaku A, Abid A et al. (2023) The NHGRI-EBI GWAS Catalog: knowledgebase and deposition resource. Nucleic Acids Research 51:D977-D985. https://doi.org/10.1093/nar/gkac1010

20. Finan C, Gaulton A, Kruger FA et al. (2017) The druggable genome and support for target identification and validation in drug development. American Association for the Advancement of Science
21. Wong CH, Siah KW, Lo AW (2019) Estimation of clinical trial success rates and related parameters. Biostatistics 20:273–286. https://doi.org/10.1093/biostatistics/kxx069
22. Nelson MR, Tipney H, Painter JL et al. (2015) The support of human genetic evidence for approved drug indications. Nat Genet 47:856–860. https://doi.org/10.1038/ng.3314
23. Minikel EV, Painter JL, Dong CC et al. (2024) Refining the impact of genetic evidence on clinical success. Nature 629:624–629. https://doi.org/10.1038/s41586-024-07316-0
24. Liu H, Doke T, Guo D et al. (2022) Epigenomic and transcriptomic analyses define core cell types, genes and targetable mechanisms for kidney disease. Nature Genetics 54:950–962. https://doi.org/10.1038/s41588-022-01097-w
25. Stanzick KJ, Li Y, Schlosser P et al. (2021) Discovery and prioritization of variants and genes for kidney function in 1.2 million individuals. Nat Commun 12:4350. https://doi.org/10.1038/s41467-021-24491-0
26. Visscher PM, Wray NR, Zhang Q et al. (2017) 10 Years of GWAS Discovery: Biology, Function, and Translation. The American Journal of Human Genetics 101:5–22. https://doi.org/10.1016/j.ajhg.2017.06.005
27. Schaid DJ, Chen W, Larson NB (2018) From genome-wide associations to candidate causal variants by statistical fine-mapping. Nat Rev Genet 19:491–504. https://doi.org/10.1038/s41576-018-0016-z
28. Hutchinson A, Asimit J, Wallace C (2020) Fine-mapping genetic associations. Hum Mol Genet 29:R81-R88. https://doi.org/10.1093/hmg/ddaa148
29. Spain SL, Barrett JC (2015) Strategies for fine-mapping complex traits. Hum Mol Genet 24:R111-9. https://doi.org/10.1093/hmg/ddv260
30. Kanai M, Elzur R, Zhou W et al. (2022) Meta-analysis fine-mapping is often miscalibrated at single-variant resolution. Cell Genomics 2. https://doi.org/10.1016/j.xgen.2022.100210
31. Zou Y, Carbonetto P, Wang G et al. (2022) Fine-mapping from summary data with the "Sum of Single Effects" model. PLOS Genetics 18:e1010299. https://doi.org/10.1371/journal.pgen.1010299
32. Begum F, Ghosh D, Tseng GC et al. (2012) Comprehensive literature review and statistical considerations for GWAS meta-analysis. Nucleic Acids Research 40:3777–3784. https://doi.org/10.1093/nar/gkr1255
33. Makinde FL, Tchamga MSS, Jafali J et al. (2021) Reviewing and assessing existing meta-analysis models and tools. Brief Bioinform 22. https://doi.org/10.1093/bib/bbab324
34. Willer CJ, Li Y, Abecasis GR (2010) METAL: fast and efficient meta-analysis of genomewide association scans. Bioinformatics 26:2190–2191. https://doi.org/10.1093/bioinformatics/btq340
35. Wuttke M, Li Y, Li M et al. (2019) A catalog of genetic loci associated with kidney function from analyses of a million individuals. Nature Genetics 51:957–972. https://doi.org/10.1038/s41588-019-0407-x
36. Gallagher MD, Chen-Plotkin AS (2018) The Post-GWAS Era: From Association to Function. The American Journal of Human Genetics 102:717–730. https://doi.org/10.1016/j.ajhg.2018.04.002
37. Aguet F, Brown AA, Castel SE et al. (2017) Genetic effects on gene expression across human tissues. Nature 550:204–213. https://doi.org/10.1038/nature24277
38. Rentzsch P, Witten D, Cooper GM et al. (2018) CADD: predicting the deleteriousness of variants throughout the human genome. Nucleic Acids Research 47:D886-D894. https://doi.org/10.1093/nar/gky1016

39. Luo Y, Hitz BC, Gabdank I et al. (2020) New developments on the Encyclopedia of DNA Elements (ENCODE) data portal. Nucleic Acids Research 48:D882-D889. https://doi.org/10.1093/nar/gkz1062
40. The ENCODE Project Consortium (2012) An integrated encyclopedia of DNA elements in the human genome. Nature 489:57–74. https://doi.org/10.1038/nature11247
41. Jassal B, Matthews L, Viteri G et al. (2020) The reactome pathway knowledgebase. Nucleic Acids Research 48:D498-D503. https://doi.org/10.1093/nar/gkz1031
42. Zhang J, Dutta D, Köttgen A et al. (2022) Plasma proteome analyses in individuals of European and African ancestry identify cis-pQTLs and models for proteome-wide association studies. Nature Genetics 54:593–602. https://doi.org/10.1038/s41588-022-01051-w
43. Stanzick KJ, Stark KJ, Gorski M et al. (2023) KidneyGPS: a user-friendly web application to help prioritize kidney function genes and variants based on evidence from genome-wide association studies. BMC Bioinformatics 24:355. https://doi.org/10.1186/s12859-023-05472-0
44. Bycroft C, Freeman C, Petkova D et al. (2018) The UK Biobank resource with deep phenotyping and genomic data. Nature 562:203–209. https://doi.org/10.1038/s41586-018-0579-z
45. Jiang L, Zheng Z, Qi T et al. (2019) A resource-efficient tool for mixed model association analysis of large-scale data. Nature Genetics 51:1749–1755. https://doi.org/10.1038/s41588-019-0530-8
46. Winkler TW, Day FR, Croteau-Chonka DC et al. (2014) Quality control and conduct of genome-wide association meta-analyses. Nature Protocols 9:1192–1212. https://doi.org/10.1038/nprot.2014.071
47. Yang J, Ferreira T, Morris AP et al. (2012) Conditional and joint multiple-SNP analysis of GWAS summary statistics identifies additional variants influencing complex traits. Nature Genetics 44:369–375. https://doi.org/10.1038/ng.2213
48. Wakefield J (2007) A Bayesian Measure of the Probability of False Discovery in Genetic Epidemiology Studies. The American Journal of Human Genetics 81:208–227. https://doi.org/10.1086/519024
49. Groopman EE, Marasa M, Cameron-Christie S et al. (2019) Diagnostic Utility of Exome Sequencing for Kidney Disease. New England Journal of Medicine 380:142–151. https://doi.org/10.1056/NEJMoa1806891
50. Wopperer FJ, Knaup KX, Stanzick KJ et al. (2022) Diverse molecular causes of unsolved autosomal dominant tubulointerstitial kidney diseases. Kidney International 102:405–420. https://doi.org/10.1016/j.kint.2022.04.031
51. Hamosh A, Scott AF, Amberger J et al. (2000) Online Mendelian Inheritance In Man (OMIM). Human Mutation 15:57–61. https://doi.org/10.1002/(SICI)1098-1004(200001)15:1<57:AID-HUMU12>3.0.CO;2-G
52. Bult CJ, Blake JA, Smith CL et al. (2018) Mouse Genome Database (MGD) 2019. Nucleic Acids Research 47:D801-D806. https://doi.org/10.1093/nar/gky1056
53. Gillies CE, Putler R, Menon R et al. (2018) An eQTL Landscape of Kidney Tissue in Human Nephrotic Syndrome. The American Journal of Human Genetics 103:232–244. https://doi.org/10.1016/j.ajhg.2018.07.004
54. McLaren W, Gil L, Hunt SE et al. (2016) The Ensembl Variant Effect Predictor. Genome Biol 17:122. https://doi.org/10.1186/s13059-016-0974-4
55. Zhou Y, Zhang Y, Lian X et al. (2022) Therapeutic target database update 2022: facilitating drug discovery with enriched comparative data of targeted agents. Nucleic Acids Research 50:D1398-D1407. https://doi.org/10.1093/nar/gkab953
56. Giambartolomei C, Vukcevic D, Schadt EE et al. (2014) Bayesian Test for Colocalisation between Pairs of Genetic Association Studies Using Summary Statistics. PLoS Genetics 10:e1004383. https://doi.org/10.1371/journal.pgen.1004383

57. Winkler TW, Rasheed H, Teumer A et al. (2022) Differential and shared genetic effects on kidney function between diabetic and non-diabetic individuals. Commun Biol 5:580. https://doi.org/10.1038/s42003-022-03448-z
58. Gorski M, Rasheed H, Teumer A et al. (2022) Genetic loci and prioritization of genes for kidney function decline derived from a meta-analysis of 62 longitudinal genome-wide association studies. Kidney International 102:624–639. https://doi.org/10.1016/j.kint.2022.05.021
59. Pruim RJ, Welch RP, Sanna S et al. (2010) LocusZoom: regional visualization of genome-wide association scan results. Bioinformatics 26:2336–2337. https://doi.org/10.1093/bioinformatics/btq419
60. Common Metabolic Diseases Knowledge Portal (cmdkp.org). https://hugeamp.org/dinspector.html?dataset=Stanzick2021_CKD_mixed&phenotype=eGFRcrea. Accessed 16 Apr 2024
61. Ghoussaini M, Mountjoy E, Carmona M et al. (2021) Open Targets Genetics: systematic identification of trait-associated genes using large-scale genetics and functional genomics. Nucleic Acids Research 49:D1311-D1320. https://doi.org/10.1093/nar/gkaa840
62. Ittisoponpisan S, Islam SA, Khanna T et al. (2019) Can Predicted Protein 3D Structures Provide Reliable Insights into whether Missense Variants Are Disease Associated? J Mol Biol 431:2197–2212. https://doi.org/10.1016/j.jmb.2019.04.009
63. Hubbard PA, Shen AL, Paschke R et al. (2001) NADPH-cytochrome P450 oxidoreductase. Structural basis for hydride and electron transfer. J Biol Chem 276:29163–29170. https://doi.org/10.1074/jbc.M101731200
64. Pandey AV, Sproll P (2014) Pharmacogenomics of human P450 oxidoreductase. Front Pharmacol 5:103. https://doi.org/10.3389/fphar.2014.00103
65. Zhang J-J, Zhang H, Ding X-L et al. (2013) Effect of the P450 oxidoreductase 28 polymorphism on the pharmacokinetics of tacrolimus in Chinese healthy male volunteers. Eur J Clin Pharmacol 69:807–812. https://doi.org/10.1007/s00228-012-1432-1
66. Banas B, Krämer BK, Krüger B et al. (2020) Long-Term Kidney Transplant Outcomes: Role of Prolonged-Release Tacrolimus. Transplant Proc 52:102–110. https://doi.org/10.1016/j.transproceed.2019.11.003
67. Uhlén M, Fagerberg L, Hallström BM et al. (2015) Proteomics. Tissue-based map of the human proteome. Science 347:1260419. https://doi.org/10.1126/science.1260419
68. Human Protein Atlas: proteinatlas.org. Accessed 30 Jun 2024
69. Morris AP, Le TH, Wu H et al. (2019) Trans-ethnic kidney function association study reveals putative causal genes and effects on kidney-specific disease aetiologies. Nat Commun 10:29. https://doi.org/10.1038/s41467-018-07867-7
70. Moore KH, Clemmer JS (2021) Questioning the renoprotective role of L-type calcium channel blockers in chronic kidney disease using physiological modeling. Am J Physiol Renal Physiol 321:F548-F557. https://doi.org/10.1152/ajprenal.00233.2021
71. Hayashi K, Nagahama T, Oka K et al. (1996) Disparate effects of calcium antagonists on renal microcirculation. Hypertens Res 19:31–36. https://doi.org/10.1291/hypres.19.31
72. Lake BB, Menon R, Winfree S et al. (2023) An atlas of healthy and injured cell states and niches in the human kidney. Nature 619:585–594. https://doi.org/10.1038/s41586-023-05769-3
73. Hinze C, Kocks C, Leiz J et al. (2022) Single-cell transcriptomics reveals common epithelial response patterns in human acute kidney injury. Genome Medicine 14:103. https://doi.org/10.1186/s13073-022-01108-9
74. Ziegler WH, Soetje B, Marten LP et al. (2020) Fibrocystin Is Essential to Cellular Control of Adhesion and Epithelial Morphogenesis. Int J Mol Sci 21. https://doi.org/10.3390/ijms21145140

75. Büscher R, Büscher AK, Weber S et al. (2014) Clinical manifestations of autosomal recessive polycystic kidney disease (ARPKD): kidney-related and non-kidney-related phenotypes. Pediatr Nephrol 29:1915–1925. https://doi.org/10.1007/s00467-013-2634-1
76. Ward CJ, Hogan MC, Rossetti S et al. (2002) The gene mutated in autosomal recessive polycystic kidney disease encodes a large, receptor-like protein. Nature Genetics 30:259–269. https://doi.org/10.1038/ng833
77. Landrum MJ, Lee JM, Benson M et al. (2017) ClinVar: improving access to variant interpretations and supporting evidence. Nucleic Acids Research 46:D1062-D1067. https://doi.org/10.1093/nar/gkx1153
78. ClinVar (2024) NM_138694.4(PKHD1):c.11525G>T (p.Arg3842Leu) AND Polycystic kidney disease - ClinVar - NCBI. https://www.ncbi.nlm.nih.gov/clinvar/RCV001292514/. Accessed 30 Jun 2024
79. Ruotsalainen V, Ljungberg P, Wartiovaara J et al. (1999) Nephrin is specifically located at the slit diaphragm of glomerular podocytes. Proc Natl Acad Sci U S A 96:7962–7967. https://doi.org/10.1073/pnas.96.14.7962
80. Beltcheva O, Martin P, Lenkkeri U et al. (2001) Mutation spectrum in the nephrin gene (NPHS1) in congenital nephrotic syndrome. Human Mutation 17:368–373. https://doi.org/10.1002/humu.1111
81. Jalanko H (2009) Congenital nephrotic syndrome. Pediatr Nephrol 24:2121–2128. https://doi.org/10.1007/s00467-007-0633-9
82. Patrakka J, Kestilä M, Wartiovaara J et al. (2000) Congenital nephrotic syndrome (NPHS1): features resulting from different mutations in Finnish patients. Kidney International 58:972–980. https://doi.org/10.1046/j.1523-1755.2000.00254.x
83. Grahammer F, Wigge C, Schell C et al. (2016) A flexible, multilayered protein scaffold maintains the slit in between glomerular podocytes. JCI Insight 1. https://doi.org/10.1172/jci.insight.86177
84. Schwarz K, Simons M, Reiser J et al. (2001) Podocin, a raft-associated component of the glomerular slit diaphragm, interacts with CD2AP and nephrin. J Clin Invest 108:1621–1629. https://doi.org/10.1172/JCI12849
85. Martin CE, Jones N (2018) Nephrin Signaling in the Podocyte: An Updated View of Signal Regulation at the Slit Diaphragm and Beyond. Front Endocrinol (Lausanne) 9:302. https://doi.org/10.3389/fendo.2018.00302
86. Reiser J, Kriz W, Kretzler M et al. (2000) The glomerular slit diaphragm is a modified adherens junction. J Am Soc Nephrol 11:1–8. https://doi.org/10.1681/ASN.V1111
87. Schnabel E, Anderson JM, Farquhar MG (1990) The tight junction protein ZO-1 is concentrated along slit diaphragms of the glomerular epithelium. J Cell Biol 111:1255–1263. https://doi.org/10.1083/jcb.111.3.1255
88. (2024) Online Mendelian Inheritance in Man (OMIM). https://www.omim.org/entry/607428?search=kirrel1&highlight=kirrel1. Accessed 20 Jul 2024
89. Sheng X, Guan Y, Ma Z et al. (2021) Mapping the genetic architecture of human traits to cell types in the kidney identifies mechanisms of disease and potential treatments. Nature Genetics 53:1322–1333. https://doi.org/10.1038/s41588-021-00909-9
90. Gorski M, Tin A, Garnaas M et al. (2015) Genome-wide association study of kidney function decline in individuals of European descent. Kidney International 87:1017–1029. https://doi.org/10.1038/ki.2014.361
91. Kato K, Hansen L, Clausen H (2021) Polypeptide N-acetylgalactosaminyltransferase-Associated Phenotypes in Mammals. Molecules 26. https://doi.org/10.3390/molecules26185504

92. Takasaki N, Tachibana K, Ogasawara S et al. (2014) A heterozygous mutation of GALNTL5 affects male infertility with impairment of sperm motility. Proc Natl Acad Sci U S A 111:1120–1125. https://doi.org/10.1073/pnas.1310777111
93. Giudici MC, Ahmad F, Holanda DG (2019) Patient with a PRKAG2 mutation who developed Immunoglobulin A nephropathy: a case report. Eur Heart J Case Rep 3. https://doi.org/10.1093/ehjcr/ytz038
94. Hallows KR, Mount PF, Pastor-Soler NM et al. (2010) Role of the energy sensor AMP-activated protein kinase in renal physiology and disease. Am J Physiol Renal Physiol 298:F1067-77. https://doi.org/10.1152/ajprenal.00005.2010
95. Wang Y, Jia L, Hu Z et al. (2018) AMP-activated protein kinase/myocardin-related transcription factor-A signaling regulates fibroblast activation and renal fibrosis. Kidney International 93:81–94. https://doi.org/10.1016/j.kint.2017.04.033
96. Goel S, Singh R, Singh V et al. (2022) Metformin: Activation of 5' AMP-activated protein kinase and its emerging potential beyond anti-hyperglycemic action. Frontiers in Genetics 13:1022739. https://doi.org/10.3389/fgene.2022.1022739
97. Song A, Zhang C, Meng X (2021) Mechanism and application of metformin in kidney diseases: An update. Biomed Pharmacother 138:111454. https://doi.org/10.1016/j.biopha.2021.111454
98. Schofield A, Bernard O (2013) Tubulin polymerization promoting protein 1 (TPPP1): A DNA-damage induced microtubule regulatory gene. Commun Integr Biol 6:e26316. https://doi.org/10.4161/cib.26316
99. Han SJ, Kim J-H, Kim J in et al. (2016) Inhibition of microtubule dynamics impedes repair of kidney ischemia/reperfusion injury and increases fibrosis. Scientific Reports 6:27775. https://doi.org/10.1038/srep27775
100. Lin Y, Nakatochi M, Sasahira N et al. (2021) Glycoprotein 2 in health and disease: lifting the veil. Genes Environ 43:53. https://doi.org/10.1186/s41021-021-00229-8
101. Schaeffer C, Devuyst O, Rampoldi L (2021) Uromodulin: Roles in Health and Disease. Annu Rev Physiol 83:477–501. https://doi.org/10.1146/annurev-physiol-031620-092817
102. Mutig K, Kahl T, Saritas T et al. (2011) Activation of the bumetanide-sensitive Na+,K+,2Cl- cotransporter (NKCC2) is facilitated by Tamm-Horsfall protein in a chloride-sensitive manner. J Biol Chem 286:30200–30210. https://doi.org/10.1074/jbc.M111.222968
103. Wolf MTF, Wu X-R, Huang C-L (2013) Uromodulin upregulates TRPV5 by impairing caveolin-mediated endocytosis. Kidney International 84:130–137. https://doi.org/10.1038/ki.2013.63
104. Wolf MTF, Zhang J, Nie M (2019) Uromodulin in mineral metabolism. Curr Opin Nephrol Hypertens 28:481–489. https://doi.org/10.1097/MNH.0000000000000522
105. Wuttke M, König E, Katsara M-A et al. (2023) Imputation-powered whole-exome analysis identifies genes associated with kidney function and disease in the UK Biobank. Nat Commun 14:1287. https://doi.org/10.1038/s41467-023-36864-8

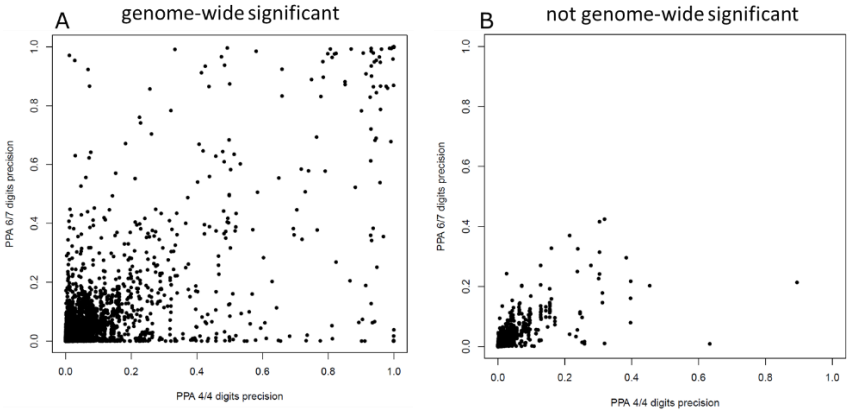
Appendix

a) OMIM API request

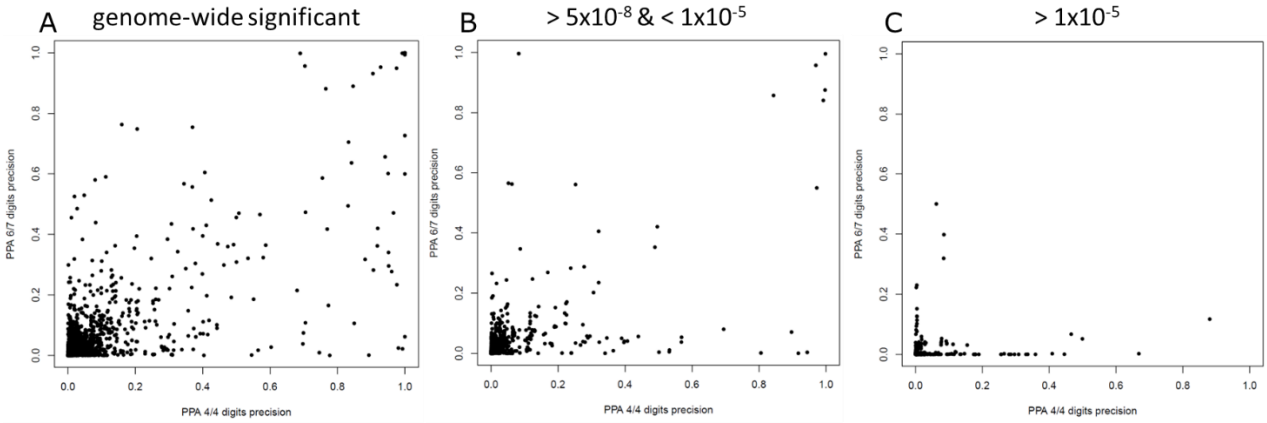
To extract all diseases from the OMIM database which include phenotypes belonging to the category genitourinary/kidney multiple API requests were used. The following API request was used to obtain the first 19 of 739 entries:

https://api.omim.org/api/entry/search?search=cs_genitourinary_kidneys:*&include=clinicalSynopsis&include=geneMap&format=xml&start=0&limit=19

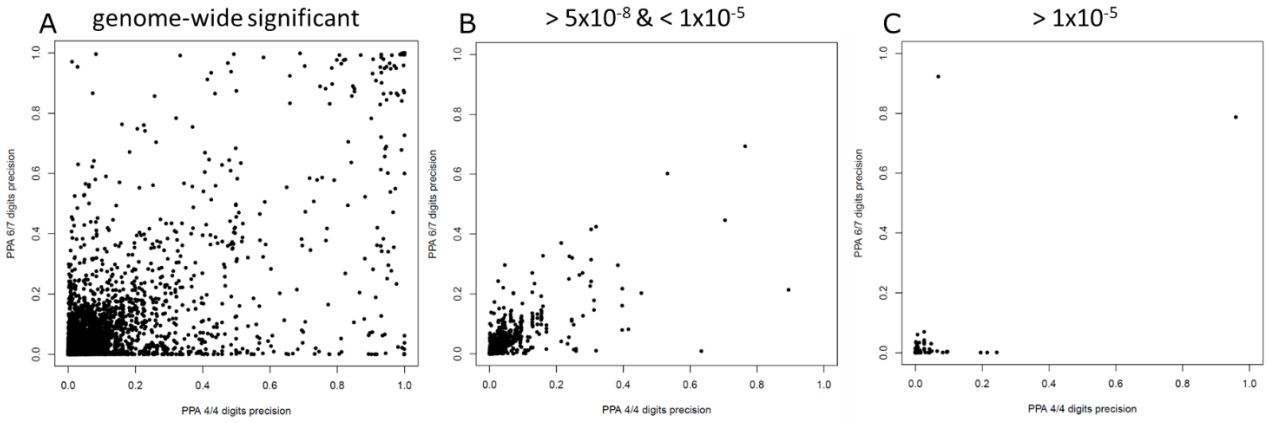
b) Supplementary Figures



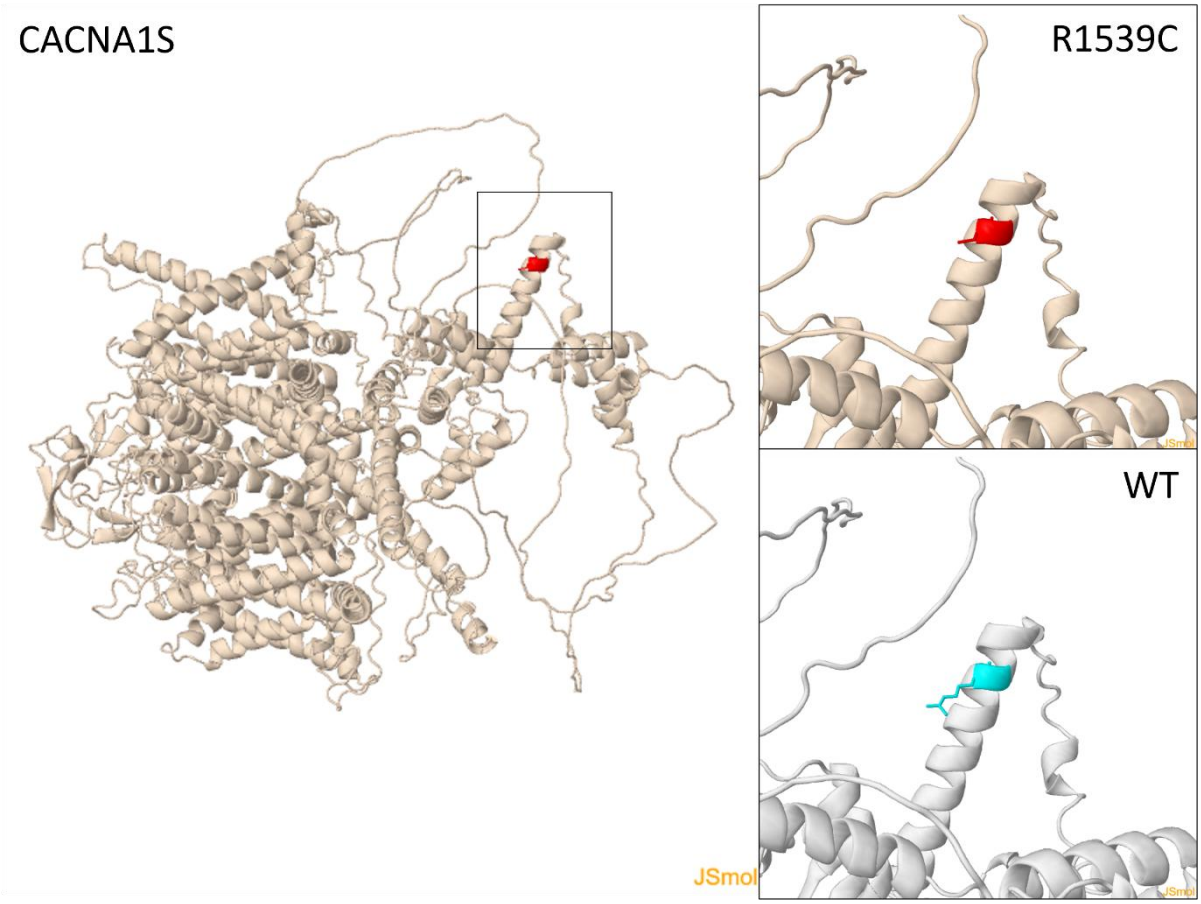
Supplementary Figure 1: PPA comparison among 424 first signals between 4/4 decimal places and 6/7 decimal places. A: PPA comparison of credible set variants from 4/4 decimal places analysis of 373 signals with genome-wide significant signal index variant in EUR and the corresponding PPA in 6/7 decimal places. B: PPA comparison of credible set variants in 4/4 decimal places of 51 signals with not genome-wide significant signal index variant in EUR and the corresponding PPA in 6/7 decimal places.



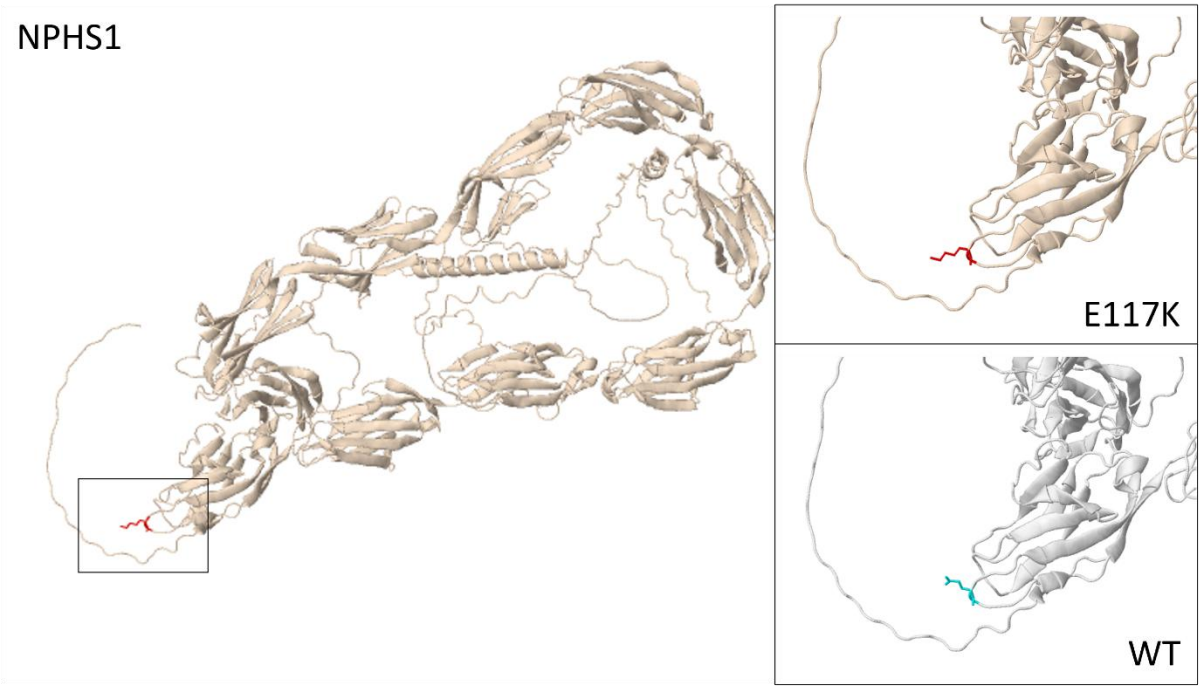
Supplementary Figure 2: PPA comparison among 210 secondary signals between 4/4 decimal places and 6/7 decimal places. A: PPA comparison of credible set variants from 4/4 decimal places analysis of 143 signals with genome-wide significant identifying p-value in the 6/7 decimal places analysis. B: PPA comparison of credible set variants from 4/4 decimal places analysis of 52 signals with identifying p-value between $5x10^{-8}$ and $1x10^{-5}$ in the 6/7 decimal places analysis. C: PPA comparison of credible set variants from 4/4 decimal places analysis of 15 signals with identifying p-value above $1x10^{-5}$ in the 6/7 decimal places analysis.



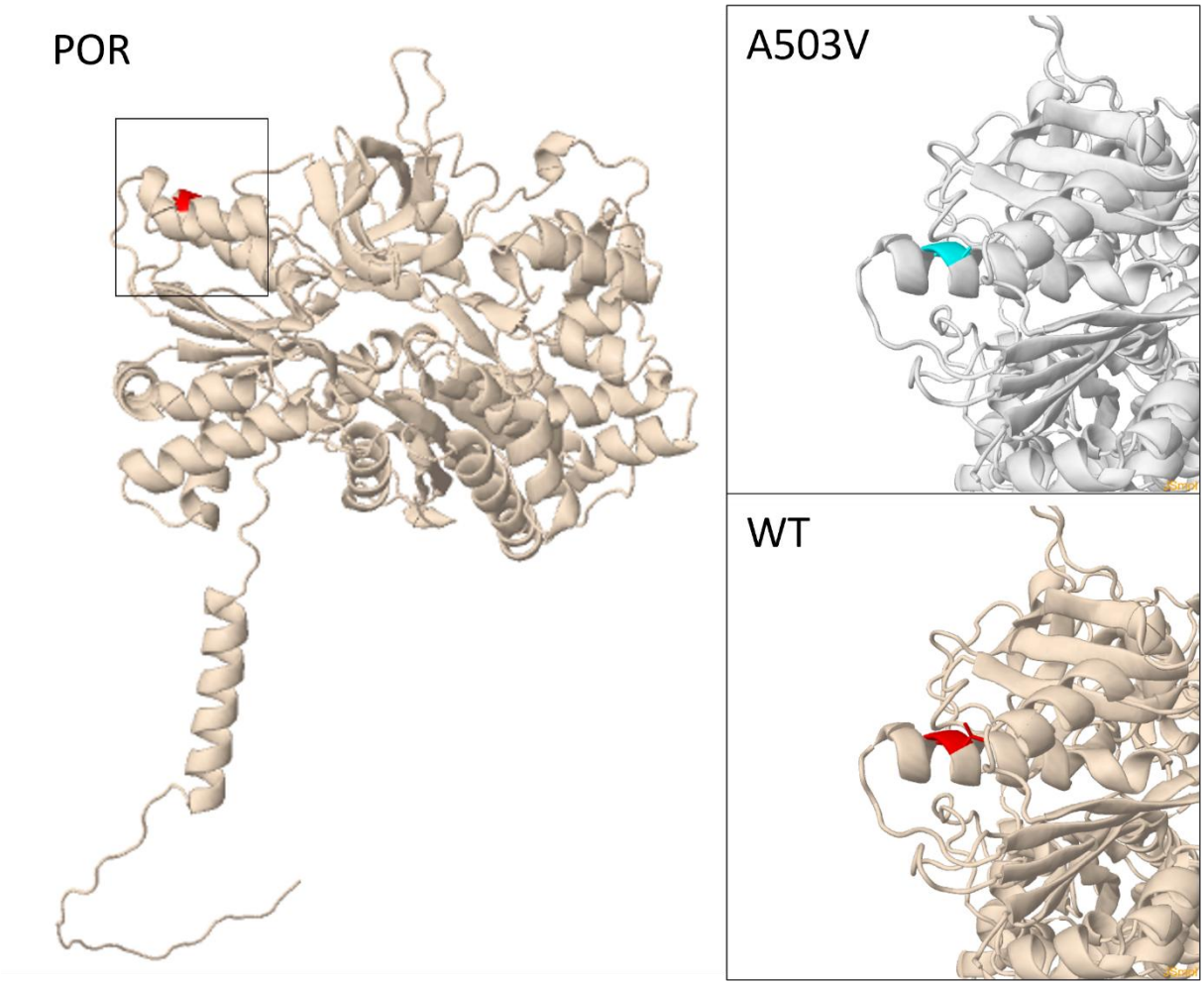
Supplementary Figure 3: PPA comparison among all 634 signals between 4/4 decimal places and 6/7 decimal places. A: PPA comparison of credible set variants from 4/4 decimal places analysis of 571 signals with genome-wide significant fully conditioned p-value in the 4/4 decimal places analysis. B: PPA comparison of credible set variants from 4/4 decimal places analysis of 59 signals with fully conditioned p-value between $5x10^{-8}$ and $1x10^{-5}$ in the 4/4 decimal places analysis. C: PPA comparison of credible set variants from 4/4 decimal places analysis of 4 signals with fully conditioned p-value above $1x10^{-5}$ in the 4/4 decimal places analysis.



Supplementary Figure 4: Predicted structural changes in CACNA1S by the R1539C mutation [62]. The left side shows the whole predicted protein 3D structure. The two pictures on the right focus on the aminoacid position with the wilde-type (WT, blue) aminoacid arginine and the mutation to cysteine (red).



Supplementary Figure 5 Predicted structural changes in NPHS1 by the E117K mutation [62]. The left side shows the whole predicted protein 3D structure. The two pictures on the right focus on the aminoacid position with the wilde-type (WT, blue) aminoacid glutamic acid and the mutation to lysine (red).



Supplementary Figure 6 Predicted structural changes in POR by the A503V mutation [62]. The left side shows the whole predicted protein 3D structure. The two pictures on the right focus on the aminoacid position with the wilde-type (WT, blue) aminoacid alanine and the mutation to valine (red).

c) Supplementary Table

Supplementary Table 1: Summary of prioritised genes from prioritisation examples of KidneyGPS. The effect allele is the eGFR lowering allele. Affected genes are influenced by the SNP by the stated SNP effect.

SNP (other allele → effect allele)	**Affected gene(s)**	**Type of SNP effect**	**SNP effect direction compared to eGFR lowering allele**
rs77924615 (A → G)	*UMOD, GP2*	eQTL	upregulating expression
rs10224002 (A → G)	*GALNTL5*	eQTL	upregulating expression
rs434215 (G → A)	*TPPP*	eQTL	upregulating expression
rs3850625 (A → G)	*CACNA1S*	missense (R1539C)	Lowering allele G is major (reference) allele
rs1057868 (T → C)	*POR*	missense (A503V)	Lowering allele C is major (reference) allele
rs76572975 (A → C)	*PKHD1*	missense (R3842L)	Lowering allele C is major (reference) allele
rs3814995 (T → C)	*NPHS1*	missense (E117K)	Lowering allele C is major (reference) allele

d) Complete KidneyGPS source code

```
################################################################################
#  KidneyGPS2.3.1 by Kira Stanzick
# This script contains the defintion of the user interface (UI) and the server part of
# KidneyGPS.
#
# Most functions and commands are used multiple times throughout the script. They are
# commented when they first appear.
################################################################################

# delete existing workspace:
rm(list = ls(all = TRUE))
# load required packages:
source("data/packages.r")
# load workspace:
load("data/workspace.RData")

############################
#
### Design User Interface
#
############################
ui <-
  fluidPage(# whole KidneyGPS is a fluid page
    # header which loads the css file and defines the function reacting on the "Enter" key
  tags$head(
  tags$link(rel = "stylesheet", type = "text/css", href =
                "style.css"),
  tags$script('
                    $(document).on("keyup", function(e) {
                     if(e.keyCode == 13){
                     Shiny.onInputChange("keyPressed",
                     Math.random());
                     }});
      ')
    ),
  title="KidneyGPS",  #Name that is shown in the Browser Tab
  #main header including the kidney image:
  tags$h1(list(tags$img(id="kidney_image",
                    height="50px",
                    width="50px",
                    src= "kidney-g8a69f0709_1920.png"),
          HTML(paste(tags$em("Kidney "), tags$b("G"), "ene",
               tags$b("P"), "rioriti",tags$b("S"),"ation - ",
               tags$b("KidneyGPS"),sep="")))),
  tags$p(list("This platform summarizes information on each of 5906
  genes overlapping the 424 loci identified for ",
  # example of abbreviations that will be shown, when hovered:
  tags$abbr(title="estimated glomerular filtration rate from serum
  creatinine",tags$b("eGFRcrea")), " based on a ",
  tags$abbr(title="genome-wide association studies","GWAS"),
  " meta-analysis of ",
  tags$a(href="https://www.ukbiobank.ac.uk/", target="_blank",
  "UK Biobank data"), # tags$a: format link
  "and ", tags$a(href="https://ckdgen.imbi.uni-freiburg.de",
```

```
target="_blank", "CKDGen consortium"), "data",
tags$b("(n=1,201,909)"),
tags$strong(tags$a(href="https://www.nature.com/articles/
s41467-021-24491-0#Sec31",
target="_blank", "[Stanzick et al. Nat. Commun. 2021]")),
". Focussed on European ancestry ",tags$b("(n=1,004,040)"),
", 594 independent signals across the 424 loci were identified
using approximate conditioned analysis. ",
"For each variant in these 594 signals, the posterior probability
of association (PPA) was computed and, for each signal, a 99%
credible set of variants was derived ",
"(i.e. smallest set of variants with >99% cumultative PPA). ",
"Credible (set) variants are considered the most likely variants
to drive the association signal (particularly those with high
PPA).")),

# From here definition of the 5 tabs inside KidneyGPS:

navbarPage(tags$p(HTML(paste(tags$b("G"),"ene",tags$b("P"),
"rioriti",tags$b("S"),"ation", sep=""))),

tabPanel("Genes", #Every Tab starts directly with the name of the Tab
tags$h3("Gene Search"), #header of category 3
p("Search for genes overlapping any of the 424 loci:"), # p: paragraph
sidebarLayout(# this layout allows for one sidebar Panel and one main Panel.
# Sidebar panel for inputs: ----
sidebarPanel(
# single gene input:
textInput(inputId="Genename", label="Gene Name", value = ""),
tags$h4(actionButton(inputId="question_gene_batch",label="?",
class="question"),"Upload Gene list:"),
tags$br(), # line break
# multiple gene input:
div(class="gene-batch",textAreaInput("gene_batch",
label="Paste a list of genes",value=NULL)),
tags$hr(),# Horizontal line ----
# Input: Select a file: ----
fluidRow( #each fluid row can have multiple columns with a total width of 12
column(12, #only one column with the full width of the fluid row, which can be as wide
as the sidebar panel
div(id="gene-upload-section",
fileInput("file2", "Choose txt file with a list of genes
(max. 2000)", #input file
multiple = FALSE,
accept = c("text/txt",
"text/comma-separated-values,text/plain",
".txt")),
actionButton(inputId="reset_gene", label="reset", class="go"))
),
),
# Horizontal line: ----
tags$hr(),
# the following section is hidden until a file is uploaded. This behaviour is controlled in the
server part:
shinyjs::hidden(
```

```
div(id="gene_batch_upload_options",
# Input: Select separator ----
radioButtons("sep_gene", "Separator",
choices = c(Comma = ",",Semicolon = ";",Tab = "\t"),
selected = ","),
)),

fluidRow(
column(2, offset=10, # column of width 2 with offset of 10 to locate it at the right
shinyjs::hidden(div(id="gene_go_div",actionButton(
inputId="gene_go", label="go", class="go"))), # The go button will be
shown if there is input in any of the input fields
))),

mainPanel( # main Panel of the sidebarLayout
wellPanel( # well Panels have a specific format
tags$h4("Specification of functional evidence for the searched
gene (s) displayed in GPS table:",
actionButton(inputId="question_GPS_gene",label="?",
class="question")),

div(class="CADD",checkboxGroupInput(inputId="columns1",
label=span("Gene contains credible set variant that is protein-
relevant"), selected = c(11:13), # pre-selection of options
inline = FALSE, width = NULL, choiceNames = c("stop-gained,
stop-lost, non-synonymus","canonical splice, noncoding change,
synonymous, splice site","other functional consequence with high
predicted deleteriousness (CADD-Phred \u2265 15)"),
choiceValues = c(11:13))), # these values represent the respective columns of
the GPS table.

div(class="eqtl",checkboxGroupInput(inputId="columns2",
label="Gene maps to credible variant that modulates gene expression
(eQTLs) or splicing (sQTLs), FDR < 5%",  selected = c(14:19,22),
inline = FALSE, width = NULL, choiceNames = c("eQTL in glomerular
tissue (NEPTUNE, or Sheng et al [Nat.Genet. 2021])","eQTL in
tubulo-interstitial tissue (NEPTUNE, or Sheng et al [Nat.Genet.
2021])","eQTL in kidney cortex tissue (GTEx)","eQTL in other tissue
(GTEx)","sQTL in kidney cortex tissue (GTEx)","sQTL in other tissue
(GTEx)", "Colocalization of gene-expression signal and eGFRcrea
signal (NEPTUNE)"), choiceValues = c(14:19,22))),

div(class="pheno",checkboxGroupInput(inputId="columns3",
label="Gene has known kidney phenotype in mouse or human:",
selected = c(20,21), inline = FALSE, width = NULL,
choiceNames = c("mouse (Mouse Genome Informatics, MGI)","human
(online mendelian inheritance in man, OMIM, Groopman et al.
[N.Engl.J.Med,2019], or Wopperer et al. [Kidney Int.,2022])"),
choiceValues = c(20,21))),

div(class="drug",checkboxGroupInput(inputId="columns4",
label="Gene is known as drug target or to show a drug
interaction:", selected = c(25,26),
inline = FALSE, width = NULL, choiceNames = c("Drug information
(kidney diseases)","Drug information (other or missing
indication)" ), choiceValues = c(25,26))),
```

```
tags$br(),
tags$h4("Specification of additional information",
actionButton(inputId="question_details_gene",
label="?",class="question")),

checkboxGroupInput(inputId="detail_evidence_locus_gene",
label=NULL,  selected = NULL, inline = FALSE, width = NULL,
choiceNames = c("Locus information for loci containing the searched
genes", "eGFR association statistics for all credible set variants
in the locus","eGFR association statistics for all credible set
variants in the locus separated by diabetes status"), choiceValues
= c("summary","cred_var","dm_stat")),

div(class="genelocuszoom",checkboxGroupInput(inputId=
"gene.locuszoom", label=NULL,  selected = NULL,inline = FALSE,
width = NULL, choiceNames = c("Regional association plot of the
eGFRcrea locus containing the searched gene (LocusZoom)"),
choiceValues = c("locus_zoom")),
),

tags$br(),
tags$h5(tags$b("Restrict results to credible set variants with a
minimum PPA* of:")),
# Input for PPA restriction
numericInput(inputId="ppa", label=NULL, value =0, min=0, max=1,
width="20%"),
tags$p("*PPA: posterior probability of the eGFRcrea association
(max. PPA 1.0)"),
))),

useShinyjs(),
shinyjs::hidden(
# here starts the results section, which is hidden as long as no search is started:
div(id="Gene_div",
tags$p(textOutput("testforme")),
tags$p(textOutput("namecheck_gene")),

shinyjs::hidden(
# this is the section of the GPS. It is only shown when the searched gene is included in
the GPS Table:
div( id ="GPS_gene",
tags$h4("GPS Table", class="Output_header"),
DT::dataTableOutput("GPSrows"),
br(),
)),
hr(),

# all these other section will be shown if there is content and the respective option is
selected.
shinyjs::hidden(div(id="div_cadd_gene",tags$h4(span("Protein-
relevant variants with predicted deleteriousness:"),
class="Output_header"),
textOutput("CADD_gene_text"),
br(),
DT::dataTableOutput("CADD_gene"),
hr())),
```

```
shinyjs::hidden(div(id="div_glo_gene",tags$h4("eQTLs in
glomerular tissue (NEPTUNE, or Sheng et al [Nat.Genet. 2021]):",
class="Output_header"),
textOutput("glo_gene_text"),
br(),
DT::dataTableOutput("glo_gene"),
hr())),

shinyjs::hidden(div(id="div_tub_gene",tags$h4("eQTLs in tubulo-
interstitial tissue (NEPTUNE, or Sheng et al [Nat.Genet. 2021]):",
class="Output_header"),
textOutput("tub_gene_text"),
br(),
DT::dataTableOutput("tub_gene"),
hr())),

shinyjs::hidden(div(id="div_GTEx_eqtl_gene",tags$h4("eQTLs in
kidney-cortex tissue (GTEx):", class="Output_header"),
textOutput("eqtl_gene_text"),
br(),
DT::dataTableOutput("eqtl_gene"),
hr())),

shinyjs::hidden(div(id="div_GTEx_sqtl_gene",tags$h4("sQTLs in
kidney-cortex tissue (GTEx):", class="Output_header"),
textOutput("sqtl_gene_text"),
br(),
DT::dataTableOutput("sqtl_gene"),
hr())),

shinyjs::hidden(div(id="div_GTEx_wo_kidney_eqtl_gene",
tags$h4("eQTLs in other tissues (GTEx):", class="Output_header"),
textOutput("eqtl_wo_kidney_gene_text"),
br(),
DT::dataTableOutput("eqtl_wo_kidney_gene"),
hr())),

shinyjs::hidden(div(id="div_GTEx_wo_kidney_sqtl_gene",
tags$h4("sQTLs in other tissues (GTEx):", class="Output_header"),
textOutput("sqtl_wo_kidney_gene_text"),
br(),
DT::dataTableOutput("sqtl_wo_kidney_gene"),
hr())),

shinyjs::hidden(div(id="div_mgi_gene",tags$h4("Kidney phenotypes
in mouse:", class="Output_header"),
textOutput("mgi_gene_text"),
br(),
DT::dataTableOutput("mgi_gene"),
hr())),

shinyjs::hidden(div(id="div_omim_gene",tags$h4("Kidney phenotypes
in human:", class="Output_header"),
textOutput("omim_gene_text"),
br(),
```

```
DT::dataTableOutput("omim_gene"),
hr())),

shinyjs::hidden(div(id="div_drug_gene_kid",tags$h4("Drugability
or Interaction (indication for kidney diseases):",
class="Output_header"),
textOutput("drug_gene_kid_text"),
br(),
DT::dataTableOutput("drug_gene_kid"),
hr())),

shinyjs::hidden(div(id="div_drug_gene_oth",tags$h4("Drugability
or Interaction (other indications):", class="Output_header"),
textOutput("drug_gene_oth_text"),
br(),
DT::dataTableOutput("drug_gene_oth"),
hr())),

shinyjs::hidden(div(id="div_summary",tags$h4("Summary of loci and
signals containing the searched gene(s):", class="Output_header"),
textOutput("summary_text"),
br(),
DT::dataTableOutput("summary"),
hr())),

shinyjs::hidden(div(id="div_cred_var_gene",tags$h4("List of all
credible variants in the locus containing the searched gene:",
class="Output_header"),
p("Shown are results from GWAS meta-analyses in European
ancestry."),
br(),
DT::dataTableOutput("cred_var_gene"),
hr())),

shinyjs::hidden(div(id="div_dm_stat_gene",tags$h4("eGFR
association statistics separated by diabetes status for all
credible variants in the locus containing the searched gene:",
class="Output_header"),
p("Shown are results from GWAS meta-analyses in European
ancestry."),
br(),
DT::dataTableOutput("dm_stat_gene"),
hr())),

shinyjs::hidden(div(id="div_locus_zoom_gene",tags$h4("Locus Zoom
Plot:", class="Output_header"),
textOutput("LocusZoom_gene_text"),
uiOutput("plot_download_links_gene"),
)),
)),
),
```

```
#here starts the next tab
tabPanel("Variants",
p(tags$h3("SNP Search",
actionButton(inputId="question_SNP",label="?",class="question"),
style="display:inline")),

p("Search for SNPs which are associated with log(eGFRcrea) at
P<5E-8 (all ancestries, unconditioned, n=1,201,909) or for SNPs
being a credible variant for any of the 594 signals (European
ancestry, conditioned, n=1,004,040):"),

useShinyjs(),
sidebarLayout(
sidebarPanel(
textInput(inputId="snp", label="Single SNP search", value=NULL,
placeholder = "rs12345 | chr:pos"),
tags$br(),

div(class="SNP-batch", textAreaInput("SNP_batch",label="Paste a
list of RSIDs",value=NULL, placeholder= "rs1234, rs4567,
chr:pos")),
tags$hr(),

# Input: Select a file ----
fluidRow(
column(12,
div(id="snp-upload-section",
fileInput("file1", "Choose txt file with a list of RSIDs (max.
2000)",
multiple = FALSE, accept = c("text/txt", "text/comma-separated-
values,text/plain",".txt")),
actionButton(inputId="reset_variant", label="reset", class="go")
)),
),
 tags$hr(),

shinyjs::hidden(
div(id="SNP_batch_upload_options",
# Input: Select separator
radioButtons("sep_snp", "Separator",
choices = c(Comma = ",",Semicolon = ";",Tab = "\t",
Space=" "),selected = ","),
)),

fluidRow(
column(12,shinyjs::hidden(div(id="snp_go_div",
actionButton(inputId="snp_go", label="go", class="go"))))
),
),

mainPanel(
wellPanel(
div(class="snp-search-options",
h4("Search options:"),
```

```
div(class="gwscredinputbox",checkboxGroupInput(inputId="variant-
search-options", label=NULL,  selected = c("gws","credvar"),
inline = FALSE, width = NULL, choiceNames = c("search for SNPs at
P<5E-8", "search for credible variants"),
choiceValues = c("gws","credvar")),
),
```
#uiOutput is a section of UI that will be generated in the server part. Here additional options get available when "credvar" is selected. Otherwise, these options are disabled.
```
uiOutput("detail.evidence.snp"),

div(class="snplocuszoom",
checkboxGroupInput(inputId="snp.locuszoom", label="Specification
of locus-based information", selected = NULL, inline = FALSE,
width = NULL, choiceNames = c("Regional association plot of the
eGFRcrea locus containing the searched SNP"),
choiceValues = c("locus_zoom")),
),
)))),

shinyjs::hidden(
div(id="SNP_div",
textOutput("snpinputcheck"),
div(id="gws_div",tags$h4("eGFRcrea Association (P<5E-8 in all-
ancestry GWAS meta-analyis, unconditioned):",
class="Output_header"),
textOutput("namecheck"),
DT::dataTableOutput("all_var"),
tags$hr()),

div(id="cred_var_div",tags$h4("Credible variant(s):",
class="Output_header"),
textOutput("namecheck2"),
DT::dataTableOutput("cred_var"),
tags$hr()),

shinyjs::hidden(div(id="div_cadd_snp",tags$h4(span("Protein-
relevance and predicted deleteriousness:"),
class="Output_header"),
textOutput("CADD_text"),
DT::dataTableOutput("CADD"),
tags$hr())),

shinyjs::hidden(div(id="div_glo_snp",tags$h4("eQTL in glomerular
tissue (NEPTUNE, or Sheng et al [Nat.Genet. 2021]):",
class="Output_header"),
textOutput("glo_text"),
DT::dataTableOutput("glo"),
tags$hr())),

shinyjs::hidden(div(id="div_tub_snp",tags$h4("eQTL in tubulo-
interstitial tissue (NEPTUNE, or Sheng et al [Nat.Genet. 2021]):",
class="Output_header"),
textOutput("tub_text"),
DT::dataTableOutput("tub"),
tags$hr())),
```

```
shinyjs::hidden(div(id="div_GTEx_eqtl_snp",tags$h4("eQTL in
kidney-cortex tissue (GTEx):", class="Output_header"),
textOutput("eqtl_text"),
DT::dataTableOutput("eqtl"),
tags$hr())),

shinyjs::hidden(
div(id="div_GTEx_sqtl_snp",tags$h4("sQTL in kidney-cortex tissue
(GTEx):", class="Output_header"),
textOutput("sqtl_text"),
DT::dataTableOutput("sqtl"),
tags$hr())),

shinyjs::hidden(div(id="div_GTEx_eqtl_wo_kidney_snp",
tags$h4("eQTL in other tissues (GTEx):", class="Output_header"),
textOutput("eqtl_wo_kidney_text"),
DT::dataTableOutput("eqtl_wo_kidney"),
tags$hr())),

shinyjs::hidden(div(id="div_GTEx_sqtl_wo_kidney_snp",
tags$h4("sQTL in other tissues (GTEx):", class="Output_header"),
textOutput("sqtl_wo_kidney_text"),
DT::dataTableOutput("sqtl_wo_kidney"),
tags$hr())),

shinyjs::hidden(div(id="div_dm_stat_snp",
tags$h4("eGFR association statistics separated by diabetes
status", class="Output_header"),
p("Shown are results from GWAS meta-analyses in European
ancestry."),
DT::dataTableOutput("dm_stat_snp"),
tags$hr())),

shinyjs::hidden(div(id="div_locus_zoom_snp",tags$h4("Locus Zoom
Plot:", class="Output_header"),
textOutput("LocusZoom_SNP_text"),
uiOutput("plot_download_links_snp"),
))))
),

# region tab
tabPanel("Region",
useShinyjs(),
tags$h3("Region search",
actionButton(inputId="question_region_search",label="?",
class="question")),
wellPanel(
fluidRow(

column(4,
selectInput("Chromosome", label = "Chromosome",
choices = list("Chromosome 1" = 1, "Chromosome 2" = 2,
"Chromosome 3" = 3,"Chromosome 4" = 4,"Chromosome 5" = 5,
"Chromosome 6" = 6,"Chromosome 7" = 7,"Chromosome 8" = 8,
"Chromosome 9" = 9,"Chromosome 10" = 10,"Chromosome 11" = 11,
"Chromosome 12" = 12,"Chromosome 13" = 13, "Chromosome 14" = 14,
```

```
"Chromosome 15" = 15,"Chromosome 16" = 16,"Chromosome 17" = 17,
"Chromosome 18" = 18,"Chromosome 19" = 19,"Chromosome 20" = 20,
"Chromosome 21" = 21,"Chromosome 22" = 22),
selected = 1)
),

column(8,
numericInput(inputId="pos_start", label="Start of region [bp]",
value = 0),
numericInput(inputId="pos_end", label="End of region [bp]",
value = 0),
shinyjs::hidden(div(id="region_go_div",p(
actionButton(inputId="region_go", label="go", class="go"),
class="go"))),
),
)),

shinyjs::hidden(
div(id="region_div",
span(textOutput("valid_gene_test")),
bsCollapse(id="regionCollapse",multiple=TRUE,
bsCollapsePanel("Overlapping genetic eGFRcrea signals",
span(textOutput("ov_loci_test"),style="color:red"),
DT::dataTableOutput("ov_loci")
),
bsCollapsePanel("GPS entries for genes in overlapping eGFRcrea
loci",
shinyjs::hidden(
div(id="ov_genes_down",
DT::dataTableOutput("GPS_region_search"),
)))
))),
),

#GPS tab
tabPanel("GPS",
p(tags$h3("Gene Prioritisation - Overview",
actionButton(inputId="question_GPS",label="?",
class="question"))),

# all filter options are in this fluid row, spaces between the columns are a result of the css sheet
fluidRow(
style = "display: flex; flex-direction: column;",
div(class = "gps-container",
column(3,  id = "gpscontainersignal",
h4("1. Signal-filter:"),
h4("Restrict to signals with any of the following properties
regarding the strength of statistical support:"),
checkboxInput(inputId = "signalppafilter", label ="The signal's
credible set contains a variant with a postior probability of
association (PPA) of:", value = FALSE),

div(id="gpsppafilter",
uiOutput("signalppafilterdetail")),
```

```
checkboxInput(inputId = "signalsmall", label ="signal has a small
credible set (1-5 variants)", value = FALSE), # checkboxes don't need
a selected value

div(id="nofiltersignal",checkboxInput(inputId = "signalnofilter",
label =p("no filter (show all signals)"), value = TRUE) ),
tags$hr(),

div(id="furthersignalfilter",h4("Restrict further to:"),
radioButtons(inputId = "signalcysbun", label="Signals located in
a eGFRcys/BUN validated locus", inline = TRUE,
choiceNames = c("yes", "no"), choiceValues = c(TRUE,FALSE),
selected = FALSE),
h4("Restrict further to:"),
radioButtons(inputId = "signalDM", label="Signals with
differential association in individuals with diabetes vs. without
diabetes", inline = TRUE, choiceNames = c("yes", "no"),
choiceValues = c(TRUE,FALSE), selected = FALSE),
#radio buttons always have a selected value
radioButtons(inputId = "signaldecline", label="Signals with
association with eGFR decline", inline = TRUE,
choiceNames = c("yes", "no"), choiceValues = c(TRUE,FALSE),
 selected = FALSE),
)),

column(5, id = "gpscontainermap",
div(fluidRow(
# opening a new fluid row inside the column of the parental fluid row.
column(8,
h4("2. Variant-to-gene mapping:"),
h4("Restrict to genes which are mapped to a credible set variant
by any of the following properties:"),

div(class = "CADD",
checkboxGroupInput(inputId="genemapprot",
label=span("Gene contains a credible set variant that is protein-
relevant"),selected = NULL, inline = FALSE, width = NULL,
choiceNames = c("stop-gained, stop-lost, non-synonymus","canonical
splice, noncoding change, synonymous, splice site","other
functional consequence with high predicted deleteriousness
(CADD-Phred \u2265 15)"), choiceValues = c(11:13)),
),

div(class = "eqtl",
checkboxGroupInput(inputId="genemapeqtl",
label="Gene maps to a credible set variant that modulates gene
expression (eQTLs) or splicing (sQTLs), FDR < 5%:",
selected = NULL, inline = FALSE, width = NULL,
choiceNames = c("eQTL in glomerular tissue (NEPTUNE, or Sheng et
al [Nat.Genet. 2021])","eQTL in tubulo-interstitial tissue
(NEPTUNE, or Sheng et al [Nat.Genet. 2021])","eQTL in kidney cortex
tissue (GTEx)","eQTL in other tissue (GTEx)","sQTL in kidney cortex
tissue (GTEx)","sQTL in other tissue (GTEx)"),
choiceValues = c(14:19)),
)),
```

```
column(4, id="ppafiltergenemap", br(),
uiOutput("detail.genemap.ppa"),
))),
br(),

div(id="nofiltermap",checkboxInput(inputId = "mapnofilter",
label =p("no filter (show all genes in selected signals)"),
value = TRUE))),

column(3, id = "gpscontainerfilter",
h4("3. Gene-to-phenotype mapping:"),
h4("Restrict to genes with any of the following properties:"),
div(class = "pheno",
checkboxGroupInput(inputId = "genelistpheno", label="Kidney
phenotypes",choiceNames = c("Gene has known kidney phenotypes in
mouse models (Mouse Genome Informatics, MGI)","Gene is described
to cause a human disease with kidney phenotype (online mendelian
inheritance in man, OMIM, Groopman et al. [N.Engl.J.Med,2019], or
Wopperer et al. [Kidney Int.,2022])"),
choiceValues = c("mouse","human"), selected=NULL),
),

div(class = "drug",
checkboxGroupInput(inputId = "genelistdrug",
label = "Drug information",
choiceNames = c("Gene is a known drug target for kidney diseases
(Therapeutic Target Database)","Gene is a known drug target for
any other disease or with unspecific indication (Therapeutic Target
Database)"), choiceValues = c("kidney", "other"),
selected = NULL)),

div(id="nofiltergenelist",checkboxInput(inputId =
"genelistnofilter", label =p("no filter (show all genes in
selected signals)"), value = TRUE))
))),
br(),

actionButton(inputId="gpsgo", label="Go prioritize!"),
br(),
tags$hr(),
verbatimTextOutput("GPS_datadescription"),

 div(DT::dataTableOutput("GPS")),
br(),

# here is an extra download button. What it does is defined in the server part
div(downloadButton("downloadGPSTable", "Download Results"),
p("Cave: filtering via column headers has no impact on the
downloaded table", style="color: red;")),
br(),

div(class="genesInGPS", textOutput("GPS_Genes1")),
br()
),
```

```
tabPanel("Documentation & Help", #Documentation Tab
# like the navbarPage navListPanel allows for multiple Tabs but these are organized on the leftside while their content is on the right:
navlistPanel(

tabPanel("Release history",
h3("Release history"),
p("KidneyGPS 2.3.1 [2023-09] - update of documentation page"),
p("KidneyGPS 2.3.0 [2023-08] - integration of eGFR association statistics separated by diabetes status for all credible set variants."),
p("KidneyGPS 2.2.0 [2023-08] - re-organisation of the GPS tab with extendend filter options."),
p("KidneyGPS 2.1.0 [2023-07] - separated drug information by indication for kidney diseases according to the ICD-11 codes."),
p("KidneyGPS 2.0.1 [2023-07] - enable variant-search for chromosome and position (based on GRCh37)."),
p("KidneyGPS 2.0 [2023-07] - reanalysis of the 424 loci regarding stability of independent signals. 594 stable independent signals were identified using stepwise conditioning with GCTA. Credible sets were re-calculated for these signals resulting in 35,885 credible set variants."),
p("KidneyGPS 1.3.1 [2023-02] - additional summary numbers in \"GPS tab\" below GPS table; in minor fixes"),
("KidneyGPS 1.3.0 [2022-11] - integration of drug target and interaction information for all genes from Therapeutic Target Database"),
p("KidneyGPS 1.2.0 [2022-11] - integration of drug target and interaction information for all genes from Therapeutic Target Database"),
p("KidneyGPS 1.2.0 [2022-10] - integration of ADTKD genes, DM-status interaction and eGFRcrea decline association; loosening of the CADD-Phred cutoff for protein-altering variants with a clear functional cosequence"),
("KidneyGPS 1.1.2 [2022-06] - small layout changes"),
p("KidneyGPS 1.1.1 [2022-04] - minor fixes"),
p("KidneyGPS 1.1.0 [2022-03]- integration of eQTL data from Susztaklab"),
p("first publication KidneyGPS 1.0 [2022-03]"),
hr()
),

tabPanel("Gene Prioritisation (GPS) - User guide",
h3("Gene Prioritisation (GPS) – How to Generate a List of Genes with Defined Properties:"),
br(),
p("The GPS tab in KidneyGPS comprises of three filter sections that can be utilized to restrict the displayed GPS table to genes and signals with specific properties:"),
h4("Filtering sections:"),
tags$ol( #start of an ordered list
tags$li( # first ordered list item
h5("'Singal-filtering:'"),
p("The first section enables to restrict the GPS table to signals with specific properties."),
```

```
tags$ul( # start of an unordered list
tags$li(
p("Filter for the strength of statistical support: You can restrict
to signals based on the posterior probability of association (PPA)
of a signal's credible set variants. Each credible set is thought
to contain the variant causal for the association with a 99%
probability. For instance, selecting PPA>99% will show signals
containing only one credible set variant. PPA>50% indicates a high
probability of one variant being causal, and PPA>10% excludes
signals with only low probability variants. Alternatively, you can
filter for signals with a small credible set of five or fewer
variants. If the PPA filter and the small-set filter are selected,
signals that meet one of these criteria are shown in the displayed
GPS table. The 'no filter' option displays all signals.")
),

tags$li(
p("Further filter options:"),
tags$ul(
tags$li(
p("You can restrict to signals that are validated for kidney
function relevance based on eGFRcys or BUN associations.")
tags$li(
p("You can restrict to signals with different associations in
individuals with vs. without diabetes, or signals associated with
eGFR decline.")
)))),

tags$li(
h5("'Variant-to-gene mapping:'"),
p("The second section deals with variant-to-gene mapping. It allows
you to restrict the GPS-view to genes that are mapped by a credible
set variant with specific properties."),

tags$ul(
tags$li(
p("Feature selection: Choose one blue or orange feature to restrict
the GPS-view to genes mapped by a credible set variant with that
specific property. When multiple options are selected genes with
non-zero entries in any of the respective GPS columns will be
shown.")),

tags$li(
p("Additional mapping restrictions:
If you choose any blue or orange feature, you can further restrict
mapped credible set variants based on the strength of statistical
support of the mapped variant itself. As in the first section, you
can select PPA and small set criteria. If both PPA and small set
criteria are chosen, the mapped variant must satisfy at least one
of them; otherwise, the gene mapped by the variant is omitted from
the GPS-view.")),

tags$li(
p("To display all genes, regardless of variant-to-gene mapping,
select the 'no filter' option. Note: These genes may not directly
relate to kidney function and could simply be located in an
```

```
eGFRcrea locus by chance."))
)),

tags$li(
("'Gene-to-phenotype mapping:'"),
p("The third section deals with gene-to-phenotype mapping. It
allows you to restrict the GPS-view to genes with specific
properties."),

tags$ul(
tags$li(
p("Kidney phenotypes: Choose from two options to restrict the GPS-
view to genes with either a kidney phenotype in mouse models or
genes known to cause human genetic diseases with kidney phenotypes.
If both options are selected, the gene must satisfy at least one
of them.")),

tags$li(
p("Drug information options: These options refer to drugs listed
in the Therapeutic Target Database (TTD), separated by indications.
One option filters genes targeted by drugs for any kidney disease,
and the other filters genes targeted by drugs for other diseases
or with unspecified indications.")),

tags$li(
p("If multiple options are selected, genes must only satisfy one
of the chosen criteria."))
))),
br(),

h4("The GPS Table:"),
p("By default, the displayed GPS Table is a full and comprehensive
summary of the annotation for all 5,906 genes and 594 eGFRcrea
signals. After applying “Signal-Filtering”, “Variant-to-gene
mapping” or “Gene-to-phenotype mapping”, the displayed GPS Table
shows the results of the respective data filtering. Blue and orange
columns contain numbers representing the count of credible set
variants targeting the respective gene through that specific
feature. If a gene was located in a locus with multiple independent
signals or was mapped by credible set variants from different
association signals, multiple rows for that gene may be
included."),
br(),
p("The number of genes meeting the filter criteria is displayed
below the GPS Table."),
br(),

h4("Step-by-step guide"),
p("Find here a step-by-step guide with three
examples:",tags$a(href="https://homepages.uni-
regensburg.de/~wit59712/gps/step-by-step_guide_for_app.pdf",
"step-by-step_guide_for_app.pdf"))
),
```

```
tabPanel("Gene-, Variant-, and Region-Search - User Guide",
h3("Gene Search:"),
p("The 'Genes' tab allows to search your gene(s) of interest.
Information is available for the 5906 genes overlapping any
eGFRcrea locus."),
h4("Search panel"),
p("The left panel on the 'Genes' tab is the search panel. There
are three options to start your search request"),

tags$ol(
tags$li(
p("The option 'Gene Name' allows to enter one gene name. KidneyGPS
uses the official HGNC gene names and does only support a few
synonyms.")),

tags$li(
p("The option 'Paste a list of genes' allows to enter a bulk of
gene names to search for. Allowed separators are spaces and all
non-letter symbols except for hyphens ('-') as those can be part
of a gene name.")),

tags$li(
p("The option 'Choose txt file with a list of genes' allows to
upload a txt-file with a list of gene names to search for. After
selecting a file via the 'Browse..' button, the separator must be
selected. This can be comma, semicolon, or tab. The 'reset' button
deletes the uploaded file.")),
),

p("The last clicked input field is automatically selected and only
gene names in this field will be included in the search."),
br(),

h4("Specification panel"),
p("Beside searching for all available information for a gene in
KidneyGPS, the search results can be restricted or extended using
the different options in the 'Specification' panel. The colored
options are selected by default, so that each information
supporting a genes relevance for kidney function will be shown.
Unselect options if these should not be shown in the results.
Additionally, you can restrict the shown resultsto genes that map
to credible set variants above a user-define PPA threshold. The
higher a variant's PPA, the more likely it is the causal variant
for the association signal. "),
br(),

h4("Search results"),
p("After clicking 'GO' or pressing 'Enter' the search will be
performed. First, an excerpt of the GPS Table will be shown for
the searched genes that are located in an eGFR locus. A text will
inform how many of the searched genes are included in KidneyGPS.
How many functionally or regulatory relevant credible variants map
to the searched gene is shown in the GPS Table. Details to these
variants can be found below the GPS Table. If a PPA filter was
selected only credible set variants meeting this criterion will be
counted in the GPS Table."),
```

```
  br(),
  hr(),

  h3("Variant Search:"),
  h4("Search panel"),
  p("The left panel on the 'Variants' tab is the search panel. There
  are three options to start your search request. KidneyGPS supports
  RS-identifiers and genetic positions (format chr:position) from
  genomic build hg19."),

  tags$ol(
  tags$li(
  p("The option 'Single SNP search' allows to enter one variant.")),

  tags$li(
  p("The option 'Paste a list of RSIDs' allows to enter a bulk of
  variants to search for. Allowed separators are spaces and all non-
  letter symbols.")),

  tags$li(
  p("The option 'Choose txt file with a list of RSIDs' allows to
  upload a txt-file with a list of variants to search for. After
  selecting a file via the 'Browse..' button, the separator must be
  selected. This can be comma, semicolon, or tab. The 'reset' button
  deletes the uploaded file.")),
  ),
  br(),

  h4("Search options"),
  p("Similar to the 'Gene search', search results can be restricted
  or extended using the search options. There is the possibility to
  search for eGFR association statistics in 'all ancestries', which
  will be displayed when a variant is genome-wide significant (in
  the all ancestry meta-analysis). In general, variants are searched
  for overlap with 99% credible set variants (selected from European-
  only meta-analysis; do not necessarily meet the genome-wide
  significane criterion). Additional functional information is
  available for these variants (coloured options) as well as
  association statistics separated by diabetes status and a regional
  association diagram for the locus containing the variant(s) being
  searched for."),
  br(),

  h4("Search results"),
  p("After clicking 'GO' or pressing 'Enter' the search will be
  performed. If the default options are selected and if the searched
  variants are genome-wide significant associated or credible set
  variants, the first two tables will display the association
  statistics. If the searched variant has functional consequences
  for a gene, this will be displayed below."),
  br(),
  hr(),

  h3("Region Search:"),
  h4("Search panel"),
  p("The tab 'Region' allows to search a region for overlapping eGFR
```

```
association signals. The region can be defined by selection the
chromosome and entering start- and end-postion of the region."),
br(),

h4("Search results"),
p("After clicking 'GO' or pressing 'Enter' the search will be
performed. If the searched region overlaps at least partly an eGFR
locus, two results tables will be displayed. The first table shows
the independent signals included in the overlapping eGFR locus.
The second table is the GPS Table excerpt for those signals
including all genes in the eGFR locus."),
br(),
hr()
),

tabPanel("Data Sources",
h3("Data Sources:"),
br(),
h4("Association with eGFRcrea:"),
p("Genetic loci and genes within these loci are based on a
",tags$abbr(title="genome-wide association studies","GWAS"),
meta-analysis for eGFRcrea of ",
tags$a(href="https://www.ukbiobank.ac.uk/", target="_blank",
"UK Biobank data"),
"and ", tags$a(href="https://ckdgen.imbi.uni-freiburg.de",
target="_blank", "CKDGen consortium"), "data",
tags$b("(n=1,201,909)."),"  Detailed information on the selection
process can be found "
,tags$a(href="https://www.nature.com/articles/s41467-021-24491-
0", target="_blank", "here."),"  A GWAS-meta-analysis restricted to
individuals of European-ancestry (n=1,004,040) was used to
identify independent association signals and to calculate
posterior probabilities of association (PPA) for all variants in
each signal. The 99% credible set of variants contains the causal
variant with 99% probability, under the assumption that there is
one causal variant per association signal and that this variant is
included in the analysis."),
br(),

h4("Association with other phenotypes:"),
h5("eGFRcys & BUN",style="font-weight:bold"),
p("GWAS meta-analyses were also performed for eGFR estimated from
serum cystatin C (eGFRcys, n=460,826) and blood urea nitrogen (BUN,
n=852,678). KidneyGPS provides the information if the locus lead
variant (variant with the smallest association p-value in a locus)
is nominal significantly associated with eGFRcys or BUN with
concordant effect directions. Summary statistics of these analyses
can be downloaded ", tags$a(href="https://www.uni-
regensburg.de/medizin/epidemiologie-praeventivmedizin/genetische-
epidemiologie/gwas-summary-statistics/index.html",
target="_blank", "here.")),

h5("Interaction with diabetes status",style="font-weight:bold"),
p(list("Diabetes mellitus (DM) is a risk factor for kidney failure.
A GWAS meta-analysis for eGFRcrea conducted separatly for
individuals with or without DM ( "
```

```
,HTML(paste("n",tags$sub("DM"),sep="")),
"=178,691,",
HTML(paste("n",tags$sub("noDM"),sep="")),"=1,296,113) by Winkler
et al. identified 7 loci with significant DM/noDM difference. 5 of
these locis showed a more pronounced effect on eGFR in DM versus
noDM (DM>NoDM), one locus had a DM-only effect and one locus a
noDM-only effect. Further information on the impact of diabetes
status on the genetic eGFRcrea effect sizes can be found in the
original publication: ",
tags$a(href="https://www.nature.com/articles/s42003-022-03448-z",
target="_blank", "Winkler et al. Commun. Biol. 2022"), ". Variants
identified by this study were mapped to eGFRcrea signals in
KidneyGPS via overlap, or strong correlation with the signal index
variant identified by Stanzick et al.")),

h5("Association with eGFRcrea decline",style="font-weight:bold"),
p(list("Progressive eGFR-decline can lead to kidney failure,
necessitating dialysis or transplantation. Hence, ",
tags$a(href="https://www.kidney-international.org/article/S0085-
2538(22)00454-9/fulltext", target="_blank", "Gorski et al. [Kidney
Int. 2022]"), " searched for genetic association with annual eGFR-
decline using 62 longitudinal studies in 343,339 individuals.
Associated variants were identified by three approaches: First, a
genome-wide screen on eGFR-decline unadjusted for eGFR-baseline
revealed two significantly (",
HTML(paste("(P",tags$sub("decline"),sep="")),
HTML(paste("&#60;", " 5 x 10",tags$sup('-8'),sep="")),")
associated variants within the ", tags$em("UMOD-PDILT"),
" locus. Second, a candidate approach among the 263 lead variants
for eGFRcrea from ",
tags$a(href="https://www.nature.com/articles/s41588-019-0407-x",
target="_blank", "Wuttke et al. [Nat. Genet. 2019]"),
" identified two associated variants (Bonferroni corrected: ",
HTML(paste("P",tags$sub("decline"),sep="")),
HTML(paste("&#60;", " 0.05/263 = 1.90 x 10",tags$sup('-4')
,sep="")), "). Third, a genome-wide screen for association with
eGFR-decline adjusted for eGFRcrea at baseline revealed five
variants, that were also associated (Bonferroni corrected: ",
HTML(paste("P",tags$sub("decline"),sep="")),
HTML(paste("&#60;" ," 0.05/12 = 4.17 x 10",tags$sup("-3")
,sep="")),
") with eGFRcrea decline unadjusted. The identified C15orf54 signal
maps to a second signal in this locus and is thus not included in
our GPS. We integrated these identified variants, when they resided
in eGFRcrea signal or showed strong correlation with the signal
index variant identified by Stanzick et al.")),
br(),

h4("CADD:"),
p("The combined annotation dependend depletion (CADD) score is a
measurement of the deleteriousness of a genetic variant [Rentzsch
et al. 2018]. By integrating multiple annotations, it contrasts
variants that survived natural selection with simulated mutations.
CADD evaluated ~8.6 billion SNPs and the CADD-Phred Score used on
this website represents the rank of variant compared to all
annotated variants. Variants with the coding and non-coding
```

```
consequences \"stop-gained\", \"stop-lost\", \"missense\",
\"canonical splice\", \"noncoding change\", \"synonymous\" or
\"splice-site\" are not restricted regarding their CADD-Phred
Score. Variants with \"other\" consequences are filtered for a
CADD-Phred Score", HTML("<span>&#8805;</span>"),"15, which
restricts our analysis to the 3.2% most deleterious variants.
Further, the analysis is restricted to variants within the affected
gene as overlap with eQTLs and sQTLs should be minimized to avoid
overscoring particular genes and variants. For additional
information regarding CADD, please vistit the ",
tags$a(href="https://cadd.gs.washington.edu/info",
target="_blank", "CADD website.")),
p("Used version: v1.6 [2020-03-23]"),
br(),

h4("eQTL and sQTL data:"),
p("All credible variants were searched in expression quantitative
trait loci (eQTL) databases. Three sources for eQTL data were
used:"),
h5("NEPTUNE",style="font-weight:bold"),
p("eQTL data from the NEPTUNE study [Gillies et al. 2018] includes
", tags$em("cis-"),"eQTLs, which are variants that influence
expression of genes within a 1Mb region centred around the variant.
The association between a variant and the expression of a gene was
deemed to be significant if the false dicovery rate (FDR) was
<0.05. This eQTL data was obtained from glomerular and tubulo-
interstitial tissue. Further information about the NEPTUNE study
can be found on the webpage of the ",
tags$a(href="https://www.neptune-study.org/about",
target="_blank", "study"), " and on the ",
tags$a(href="http://nephqtl.org/about", target="_blank", "NephQTL
browser.")),
p("Version from [2017-09-25]"),

h5("Susztaklab (Sheng et al.)", style="font-weight:bold"),
p("The Susztaklab also provides comprehensive kidney omics data.
We integrated the eQTL data from glomerular und tubulo-interstitial
tissue published by Sheng et al.",
tags$a(href="https://www.nature.com/articles/s41588-021-00909-9",
target="_blank", " (Sheng, X. et al., Nature Genetics, 2021)")),

h5("GTEx", style="font-weight:bold"),
p("In contrast to the other two eQTL sources, the GTEx project is
not restricted to kidney tissue. Furthermore, additional splicing
altering variants (sQTLs) were investigated. Thus, the here
integrated GTEx data includes ",tags$em("cis-"),"eQTL and -sQTL
information from 48 different tissues with a mapping window of 1Mb
up- and downstream of the transcription start site [Auget et al.
2021]. Further information about GTEx can be found
",tags$a(href="https://www.gtexportal.org/home/",
target="_blank", "here.")),
p("Used version: GTEx Release v7 [2017-09-05]"),
br(),

h4("Mouse phenotypes:"),
p("Information on genes with kidney-relevant phenotypes in mice
```

```
origin from the Mouse Genome Informatics database (MGI,[Bult el
al. 2018]). This includes all phenotypes subordinate to \"abnormal
kidney morphology\" (MP:0002135) and \"abnormal kidney
physiology\" (MP:0002136). Further information how this data was
collected can be found on the ",
tags$a(href="http://www.informatics.jax.org/", target="_blank",
"MGI webpage.")),
p("Version from [2020-06-03]"),
br(),

h4("Human phenotypes:"),
p("We used three sources to identify genes causing genetic
disorders with kidney phenotype in human:"),
h5("OMIM", style="font-weight:bold"),
p("The Online Mendelian Inheritance in Man (OMIM) database [Hamosh
et al. 2000] was queried for phenotype entries subordinate to the
clinical synopsis class \"kidney\". Diseases with \"kidney\"-
phenotype entries being: \"normal kidneys\", \"normal renal
ultrasound at ages 4 and 7 (in two family)\", \"no kidney
disease\", \"no renal disease; normal renal function\", \"normal
renal function; no kidney disease\" and \"no renal findings\" were
manually excluded. Be aware that OMIM entries missing a clinical
synopsis entry are not included in kidneyGPS regardless of a
potential kidney involvement. Further information on the diseases
can be found at the ",tags$a(href="https://www.omim.org/",
target="_blank", "OMIM webpage.")),
p("Version from [2020-08-07]"),

h5("Groopman et al.", style="font-weight:bold"),
p(tags$a(href="http://www.columbiamedicine.org/divisions/gharavi/
files/Kidney_Gene_List_625.xlsx", target="_blank","A list of 625
genes "), "associated with Mendelian forms of kidney and
genitourinary disease was published by Groopman et al. in 2019 in
the New England Journal of Medicine. The original article
\"Diagnostic Utility of Exome Sequencing for Kidney Disease\" can
be found ",
tags$a(href="https://www.nejm.org/doi/10.1056/NEJMoa1806891",
target="_blank", "here."),"  Please notice that not all 625 genes
are included in any eGFRcrea locus and thus cannot be found in
kidneyGPS."),

h5("Wopperer al.", style="font-weight:bold"),
p("Autosomal Dominant Tubulointerstitial Kidney Disease (ADTKD) is
a heriditary kidney-disease normaly caused by mutations in at least
one of five genes (", tags$em("UMOD, MUC1, REN, HNF1B, SEC61A1"),
") and leads to kidney failure in midadulthood. However, Wopperer
et al. (2022) identified 27 putative novel ADTKD genes, of which
9 are located within an eGFRcrea associated locus. Disease type of
known ADTKD genes is stated as \"confirmed ADTKD\" in the \"Kidney
phenotypes in human\" section and as \"putative ADTKD\" for the
novel genes. The original publication can be found ",
tags$a(href="https://www.kidney-international.org/article/S0085-
2538(22)00384-2/fulltext", target="_blank","here.")),
br(),

h4("Drug information:"),
```

```
  p("Information on weather a gene, it's mRNA or the respective
  protein is a known drug target or interacts with a drug (e.g. as
  transporter) was downloaded from the Thearepeutic Target Database
  (TTD). \"Highest drug status\" and \"disease/indication\" refer to
  the drug and not necessarily to the shown target-drug pair.
  Additional information on TTD can be found in the related
  publication from ",
  tags$a(href="https://doi.org/10.1093/nar/gkab953",
  target="_blank","Ying Zhou et al. 2022."),"  Drugs referring to
  kidney related indications were obtained using ICD-11 codes GB4'X'
  to GB9'X'"),
  hr()
  ),

  tabPanel("Privacy, data security an License",
  h3("Privacy, data security an License"),
  div(
  p("We take the privacy and security of your data very seriously.
  Your input data is only used within the application to provide the
  intended functionality; your input data is not stored.."),
  p("Data Collection: Data Collection: We do not collect any user
  data, including IP addresses, searched genes, or variants within
  the RShiny application."),
  p("Data Logging: We do not log any data queries or actions
  performed by users within the application."),
  p("Data Sharing: We do not share any user data or information with
  third parties."),
  p("Data Access: Only authorized personnel have access to the
  application's underlying infrastructure."),
  br(),

  p("This work is licensed under a Creative Commons Attribution 4.0
 International License."),
  p("This RShiny web-server is operated by the University of
  Regensburg. The full privacy policy can be viewed here:
  https://www.uni-regensburg.de/datenschutz/")
  ),
  hr()
  ),

  tabPanel("Citation",
  h3("Citation"),
  p("Did you use KidneyGPS for a publication? We would appreciate if
  you cite us: \"KidneyGPS: an easily accessible web application to
  prioritize kidney function genes and variants based on evidence
  from genome-wide association studies\"  and the original data
  sources appropriately."),
  hr()
  ),

  tabPanel("Contact",h3("Contact"),
  p("If you have any questions not answered by the \"Documentation
  & Help\" section, please feel free to ask your question in this
  user forum: ",
  tags$a(href="https://github.com/kjstanzick/KidneyGPS/discussions/
  categories/q-a", "KidneyGPS at GitHub") ,"or contact: Kira
```

```
  Stanzick ", tags$a(href="mailto:kira-julia.stanzick@klinik.uni-
  regensburg.de",
  "(kira-julia.stanzick@klinik.uni-regensburg.de)")),
  hr()
  ),
  widths = c(2,8)
  )))
)


################################################
#   server part
#
#
################################################
server <- function(input, output, session) {

  #overall definitions:
  # definition of some table headers, if there is a double-row table header needed, or if
  there should be additional information available with mouse hover

  sketchcadd = htmltools::withTags(table(
   class = 'display',
   thead( # start of table header
    tr( # first table header row
     th('RSID'), # first cell
     th(class="PPA hover", 'PPA', p(class="PPAcomment", "Posterior
      probability of association of the SNP (probability of driving
      the association of the respective signal)")), # cell with additional
      information
     th('Affected gene'),
     th('Locus ID'),
     th(class="Signalidwithcomment hover",'Signal ID',
      span(class="Signalidcomment", "ID of independent signals
      within     a locus")),
     th('Functional consequence'),
     th(class="REF hover",'Reference allele',
      span(class="REFcomment", "Allele for reference aminoacid")),
     th(class="ALT hover",'Alternative allele',
      span(class="ALTcomment", "Allele for altered aminoacid")),
     th('Aminoacid position'),
     th('Reference aminoacid'),
     th('Alternative aminoacid'),
     th(class="CADDscore hover",'CADD Phred-Score',
      span(class="CADDscorecomment", "Scaled score for the
      deleteriousness of this allelic exchange"))
    )
   )
  ))

  sketchcredvar = htmltools::withTags(table(
   class = 'display',
   thead(
    tr(
     th( rowspan=2, 'RSID*'), # cell spans two rows
     th(rowspan=2,'Locus ID'),
```

```
    th(rowspan=2,class="Signalidwithcomment hover",'Signal ID',
     span(class="Signalidcomment", "ID of independent signals
     within a locus")),
    th(rowspan=2,class="PPA hover", 'PPA', p(class="PPAcomment",
     "Posterior probability of association of the SNP (probability
      of driving the association of the respective signal)")),
    th(rowspan=2,'N'),
    th(rowspan=2,'Effect allele'),
    th(rowspan=2,'Other allele'),
    th(rowspan=2,'EAF'),
    th(colspan=3, 'Association with log(eGFRcrea) conditioned on
     other signal index variants'), # cell spans three columns
    th(colspan=3, 'Association with log(eGFRcrea) unconditioned'),
    th(rowspan=2,'Chr'),
    th(rowspan=2, class="pos hover",'Pos',
     span(class="poscomment","SNP position from GRCh37")),
   ),
   tr( # second row of this header
    lapply(c('beta', 'StdErr', 'P-value','beta', 'StdErr',
     'P-value'), th)
   )
  )
 ))

 sketchcredvargene = htmltools::withTags(table(
  class = 'display',
  thead(
   tr(
    th( rowspan=2, 'RSID*'),
    th(rowspan=2,'Locus ID'),
    th(rowspan=2,class="Signalidwithcomment hover",'Signal ID',
     span(class="Signalidcomment", "ID of independent signals
     within a locus")),
    th(rowspan=2,class="PPA hover", 'PPA', p(class="PPAcomment",
     "Posterior probability of association of the SNP (probability
     of driving the association of the respective signal)")),
    th(rowspan=2, 'Functional impact on the searched gene'),
    th(rowspan=2,'N'),
    th(rowspan=2,'Effect allele'),
    th(rowspan=2,'Other allele'),
    th(rowspan=2,'EAF'),
    th(colspan=3, 'Association with log(eGFRcrea) conditioned on
     other signal index variants'),
    th(colspan=3, 'Association with log(eGFRcrea) unconditioned'),
    th(rowspan=2,'Chr'),
    th(rowspan=2, class="pos hover",'Pos',
     span(class="poscomment","SNP position from GRCh37")),
   ),
   tr(
    lapply(c('beta', 'StdErr', 'P-value','beta', 'StdErr',
    'P-value'), th)
   )
  )
 ))
```

```
sketchdmstatgene = htmltools::withTags(table(
 class = 'display',
 thead(
  tr(
   th(rowspan=2, 'RSID'),
   th(rowspan=2,'Effect allele'),
   th(rowspan=2,'Other allele'),
   th(colspan=5, 'Association of eGFRcrea in individuals with
    Diabetes'),
   th(colspan=5, 'Association of eGFRcrea in individuals without
    Diabetes'),
   th(rowspan=2,'P-Value of the difference in association between
    Diabetes and no Diabetes')
  ),
  tr(
   lapply(c('EAF', 'beta', 'StdErr','P-value', 'N', 'EAF', 'beta',
   'StdErr','P-value', 'N'), th)
  )
 )
))

sketchallsig = htmltools::withTags(table(
 class = 'display',
 thead(
  tr(
   th( rowspan=2, 'RSID*'),
   th(rowspan=2,'Chr'),
   th(rowspan=2,'Pos'),
   th(rowspan=2,'Effect allele'),
   th(rowspan=2,'Other allele'),
   th(rowspan=2,'N'),
   th(rowspan=2,'EAF'),
   th(colspan=3,'Association with log(eGFRcrea)
    unconditioned**'),
   th(rowspan=2,'Nearest gene'),
   th(rowspan=2,'Distance to nearest gene'),
   th(rowspan=2,'Locus ID')
  ),
  tr(
   lapply(c('beta', 'StdErr', 'P-value'), th)
  )
 )
))

# GPS tab ------------------------------------------------------
# header of GPS table
sketch = htmltools::withTags(table(
 class = 'display',
 thead(
  tr(
   th(rowspan = 2, class="Geneofficial hover", 'Gene*',
    span(class="Genecomment", "Official gene name from HGNC")),
   th(rowspan = 2, 'Locus name**'),
   th(rowspan = 2, class="Locusidwithcomment hover",'Locus ID',
    span(class="Locusidcomment", "k: known locus from Wuttke et
    al. Nat.commun.2019, n: novel locus in Stanzick et al.
```

```
      Nat.commun. 2021")),
     th(rowspan = 2, class="Signalidwithcomment hover",'Signal ID',
      span(class="Signalidcomment", "ID of independent signals
      within a locus")),
     th(rowspan = 2, class="validationwithcomment hover",'eGFRcys or
      BUN validation', span(class="validationcomment",
      list( "Information whether the locus lead variant is nominal
      significant (P<0.05) associated with concordent effect
      direction with ", tags$abbr(title="glomerular filtration rate
      estimated from serum cystatin C","eGFRcys"),"or
      ",tags$abbr(title="blood urea nitrogen","BUN")))),
     th(rowspan = 2, class="DMwithcomment hover",'Signal association
      depends on DM', span(class="DMcomment",list( "Information
      whether the signal index variant (or a correlated variant)
      shows significant interaction with ",
      tags$abbr(title="Diabetes mellitus","DM"), "-status, Winkler
      et al. 2022"))),
     th(rowspan = 2, class="declinewithcomment hover",'Signal
      association with eGFRcrea decline',
      span(class="declinecomment",list( "Information whether the
      signal index variant (or a correlated variant) was established
      with annual eGFR-decline, Gorski et al. 2022"))),
     th(rowspan = 2, '# credible variants in signal'),
     th(colspan = 3, class="CADD", span('# protein-relevant credible
      variants in the gene')),
     th(colspan = 6, class="eqtl", '# credible variants that modulate
      gene expression (eQTLs) or splicing (sQTLs), FDR < 5%'),
     th(rowspan = 2, class="eqtl", 'Coloc in NEPTUNE tissue'),
     th(rowspan = 2, class="pheno",'# kidney phenotypes in mouse'),
     th(rowspan = 2, class="pheno", '# kidney phenotypes in human'),
     th(rowspan = 2, class ="drug", 'Gene is known as drug target or
      for drug interaction'),
     th(rowspan = 2, 'Distance to locus lead variant'),
     th(rowspan = 2, 'Chromosome'),
     th(rowspan = 2, 'Position gene start'),
     th(rowspan = 2, 'Position gene end')
    ),
    tr(
     th('stop-gained, stop-lost, non-synonymous'),
     th('canonical splice, noncoding change, synonymous, splice
      site'),
     th(span('other deleterious variant (CADD-
      Phred',HTML('<span>&#8805;</span>'),'15)')),
     th(class='eQTLglowithcomment hover', 'eQTL glomerulus',
      span(class='eQTLglocomment','NEPTUNE, or Sheng et al.
      [Nat.Genet. 2021]')),
     th(class='eQTLtubwithcomment hover','eQTL tubulo-
      interstitium', span(class='eQTLtubcomment','NEPTUNE, or Sheng
      et al. [Nat.Genet. 2021]')),
     th('eQTL kidney cortex (GTEx)'),
     th('eQTL other tissue (GTEx)'),
     th('sQTL kidney cortex (GTEx)'),
     th('sQTL other tissue (GTEx)')
    )
   )
  ))
```

```
# color definition of header:
headerCallback <- "function( thead, data, start, end, display ) {
 $(thead).closest('thead').find('th').eq(8).css('background-
  color', '#ccffff');
 $(thead).closest('thead').find('th').eq(9).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(10).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(11).css('background-
  color', '#b3ffb3');
 $(thead).closest('thead').find('th').eq(12).css('background-
  color', '#b3ffb3');
 $(thead).closest('thead').find('th').eq(13).css('background-
  color', '#d9b3ff');
 $(thead).closest('thead').find('th').eq(18).css('background-
  color', '#ccffff');
 $(thead).closest('thead').find('th').eq(19).css('background-
  color', '#ccffff');
 $(thead).closest('thead').find('th').eq(20).css('background-
  color', '#ccffff');
 $(thead).closest('thead').find('th').eq(21).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(22).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(23).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(24).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(25).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(26).css('background-
  color', '#ffdb99');
}"

initComplete <- "function(settings, json) {
 $('.dt-button.buttons-collection').css('width', 'auto');
 $('.dt-button.buttons-collection select').css('width', '300px');
 $('.dt-button.buttons-collection').css('min-width', '300px');
}"

# render active radio buttons when input$signalppafilter is selected, else disable buttons:
observe({
 if(input$signalppafilter){
  output$signalppafilterdetail <- renderUI(
   radioButtons(inputId = "signalhigh",
    choiceNames = c("PPA >99% (most likely causal variants)", "PPA
     >50% (variants with high probability of being causal)", "PPA
     >10% (excluding variants with little probability of being
     causal)"),
    choiceValues = c(0.99, 0.5, 0.1),
    label =NULL,
    selected = 0.5)
  )
 }else{
  output$signalppafilterdetail <- renderUI(
```

```
    disabled(radioButtons(inputId = "signalhigh",
     choiceNames = c("PPA >99% (most likely causal variants)", "PPA
      >50% (variants with high probability of being causal)", "PPA
      >10% (excluding variants with little probability of being
      causal)"),
     choiceValues = c(0.99, 0.5, 0.1),
     label =NULL,
     selected = 0.5))
   )
  }
})

# update no signalnofilter button when input$signalppafilter changes
observeEvent(input$signalppafilter,{
 if(!(input$signalppafilter | input$signalsmall)){
  updateCheckboxInput(session, "signalnofilter", value=TRUE,
   label = "no filter (show all signals)") # selected because the other
   options are not selected
 }
 req(input$signalppafilter)
 updateCheckboxInput(session, "signalnofilter", value=FALSE,
  label = "no filter (show all signals)") # not selected because
  input$signalppafilter is selected
})

# update no signalnofilter button when input$signalsmall changes
observeEvent(input$signalsmall,{
 if(!(input$signalppafilter | input$signalsmall)){
  updateCheckboxInput(session, "signalnofilter", value=TRUE,
   label = "no filter (show all signals)")
 }
 req(input$signalsmall)
 updateCheckboxInput(session, "signalnofilter", value=FALSE,
  label = "no filter (show all signals)")
})

# update input$signalsmall and input$signalppafilter to FALSE when input$signalnofilter
is selected
observeEvent(input$signalnofilter,{
 req(input$signalnofilter)
 updateCheckboxInput(session, "signalppafilter", value=FALSE,
  label = "The signal's credible set contains a variant with
   posterior probability of association (PPA) of:")
 updateCheckboxInput(session, "signalsmall", value=FALSE,
  label = "signal has a small credible set (1-5 variants)")
})

# enable/disable detailed mapping options in dependence of the selection of protein or
expression regulating variant:
observe({
# check if something is selected:
 if (isTruthy(input$genemapprot) | isTruthy(input$genemapeqtl)) {
  updateCheckboxInput(session, "mapnofilter",
   label = "no filter (show all genes in selected signals)",
   value = FALSE) # update input$mapnofilter to FALSE because there is some
   mapping criterium selected
```

```
    output$detail.genemap.ppa <- renderUI( # render additional options:
     div(
      h4("Restrict mapping to credible set variants with any of
       the following properties regarding strength of statistical
       support:"),
      checkboxInput(inputId = "mapppafilter",
       label ="Posterior probability of association (PPA):",
       value = TRUE),
      div(id="gpsmapppafilter",
       uiOutput("mapppafilterdetail") # this is renderd below in dependence
       of input$mapppafilter
      ),
      checkboxInput(inputId = "mapsmall",
       label ="variant is contained in small credible set (1-5
       variants)", value = TRUE),
      div((checkboxInput(inputId = "nofiltermapppa",
       label =p("no filter (show all genes mapped by any credible
       set variant)"), value = FALSE)))
     )
    )
   } else {
    updateCheckboxInput(session, "mapnofilter",
     label = "no filter (show all genes in selected signals)",
     value = TRUE)
    output$detail.genemap.ppa <- renderUI(
     div(
      h4("Restrict mapping to credible set variants with any of the
       following properties regarding strength of statistical
       support:"),
      disabled(checkboxInput(inputId = "mapppafilter",
       label ="Posterior probability of association (PPA):",
       value = FALSE)),
      div(id="gpsmapppafilter",
       uiOutput("mapppafilterdetail")
      ),
      disabled(checkboxInput(inputId = "mapsmall",
       label ="variant is contained in small credible set (1-5
       variants)", value = FALSE)),
      div(disabled(checkboxInput(inputId = "nofiltermapppa",
       label =p("no filter (show all genes mapped by any credible
       set variant)"), value = TRUE)))
     )
    )
   }
  })

  #render mapppafilterdetail:
  observe({
   output$mapppafilterdetail <- renderUI(
   # first render disabled. When input$mapppafilter is disabled it has no value, thus it can
   not be used as if-criterion
    disabled(radioButtons(inputId = "maphigh",
     choiceNames = c("PPA >99%", "PPA >50%", "PPA >10%"),
     choiceValues = c(0.99, 0.5, 0.1),
     label =NULL,
     selected = 0.5))
```

```
  )
  req(input$mapppafilter) # check if input$mapppafilter has a value
  if(input$mapppafilter){ #if input$mapppafilter is TRUE render radio buttons
   output$mapppafilterdetail <- renderUI(
    radioButtons(inputId = "maphigh",
     choiceNames = c("PPA >99%", "PPA >50%", "PPA >10%"),
     choiceValues = c(0.99, 0.5, 0.1),
     label =NULL,
     selected = 0.5)
   )
  }else{ # else disable buttons
   output$mapppafilterdetail <- renderUI(
    disabled(radioButtons(inputId = "maphigh",
     choiceNames = c("PPA >99%", "PPA >50%", "PPA >10%"),
     choiceValues = c(0.99, 0.5, 0.1),
     label =NULL,
     selected = 0.5 ))
   )
  }
 })

 observeEvent(input$mapnofilter,{ # update other options when
input$mapnofilter is selected
  req(input$mapnofilter)
  updateCheckboxGroupInput(session,"genemapprot",
   selected = NULL,
   choiceNames = c("stop-gained, stop-lost, non-synonymus",
    "canonical splice, noncoding change, synonymous, splice site",
    "other functional consequence with high predicted
     deleteriousness (CADD-Phred \u2265 15)"),
   choiceValues = c(11:13)
  )
  updateCheckboxGroupInput(session, "genemapeqtl",
   selected = NULL,
   choiceNames = c("eQTL in glomerular tissue (NEPTUNE, or Sheng et
    al [Nat.Genet. 2021])","eQTL in tubulo-interstitial tissue
    (NEPTUNE, or Sheng et al [Nat.Genet. 2021])","eQTL in kidney
    cortex tissue (GTEx)","eQTL in other tissue (GTEx)","sQTL in
    kidney cortex tissue (GTEx)","sQTL in other tissue (GTEx)"),
   choiceValues = c(14:19)
  )
 })

 observeEvent(input$nofiltermapppa,{ # update other options when
input$nofiltermapppa is selected
  if(input$nofiltermapppa){
   updateCheckboxInput(session,"mapppafilter", value = FALSE)
   updateCheckboxInput(session, "mapsmall",value=FALSE)
  }
 })

 observe({ # update input$nofiltermapppa to FALSE when one of the other options is
selected
  if(isTruthy(input$mapsmall)| isTruthy(input$mapppafilter)){
   updateCheckboxInput(session, "nofiltermapppa",value=FALSE)
  }else{
```

```
   updateCheckboxInput(session,"nofiltermapppa",value=TRUE)
  }
 })

 observe({ # if nothing is selected set input$genelistnofilter to TRUE
  if (!isTruthy(input$genelistpheno) &&
      !isTruthy(input$genelistdrug)) {
   updateCheckboxInput(session, "genelistnofilter", value = TRUE)
  } else {
   updateCheckboxInput(session, "genelistnofilter", value = FALSE)
  }
 })

 observeEvent(input$genelistpheno, { # update input$genelistnofilter to FALSE
when input$genelistpheno is selected
  if (isTruthy(input$genelistpheno)) {
   updateCheckboxInput(session, "genelistnofilter", value = FALSE)
  }
 })

 observeEvent(input$genelistdrug, {# update input$genelistnofilter to FALSE
when input$genelistdrug is selected
  if ( isTruthy(input$genelistdrug)) {
   updateCheckboxInput(session, "genelistnofilter", value = FALSE)
  }
 })

 observe({ #update the other options when input$genelistnofilter is selected
  if (input$genelistnofilter) {
   updateCheckboxGroupInput(session, "genelistpheno",
    choiceNames = c("Gene has known kidney phenotypes in mouse
     models (Mouse Genome Informatics, MGI)","Gene is described to
     cause a human disease with kidney phenotype (online mendelian
     inheritance in man, OMIM, Groopman et al. [N.Engl.J.Med,2019],
     or Wopperer et al. [Kidney Int.,2022])"),
    choiceValues = c("mouse","human"),
    selected = NULL)
   updateCheckboxGroupInput(session,"genelistdrug",
    choiceNames = c("Gene is a known drug target for kidney diseases
     (Therapeutic Drug Database)","Gene is a known drug target for
     any other diseases or with unspecific indication (Therapeutic
     Drug Database)"),
    choiceValues = c("kidney", "other"),
    selected = NULL)
  }
 })


 ##### check if all required options are selected:
 go <- reactiveValues("gps_correct" = FALSE) #go$gps_correct is FALSE
as long as the options are not correctly selected

 observeEvent(input$gpsgo,{
  if(
   (!input$signalppafilter
    &!input$signalsmall&!input$signalnofilter)
```

```
    |(!isTruthy(input$genemapprot)&!isTruthy(input$genemapeqtl)
     &!input$mapnofilter)|(!isTruthy(input$genelistpheno)
     &!isTruthy(input$genelistdrug)&!input$genelistnofilter)
   ){
    go$gps_correct = FALSE
   }else{
    go$gps_correct = TRUE
   }
  })

  observeEvent(input$gpsgo,{ # when the "Go Priotize" Button is clicked do the
  following
   if(!go$gps_correct){ # when the filter options are not correct
    showModal(modalDialog( # Show additional window with warning message
     titel="Missing GPS filter-options",
     tags$h4("Error: There are options missing for gene-
      prioritisation!", style="color: red;"),
     p("Please check your selections. Did you select 'no filter' in
      section 1, 2 and 3 if you do not want to restrict your search
      results to any of the available options?"),
     easyClose = TRUE, # one can close the window by clicking outside the window
     footer=NULL,
     size = "m" # size of the window
    ))
   }
  })


  # introduce three reactive values which contain indices of the GPS Table. This will be
  reduced to those meeting the filter criteria in the following.
  gps_gps_rows <- reactiveValues("signalfilter" = c(1:nrow(GPS)),
   "mapfilter"=c(1:nrow(GPS)), "genelistfilter"=c(1:nrow(GPS)))

  observeEvent(input$gpsgo,{ # always waiting for the "Go Priotize"-Button
   if(go$gps_correct){ # and only doing something when the filter options are
   correct
    if(isTruthy(input$signalppafilter)
       &isTruthy(input$signalsmall)){ #PPA Filter AND small-set filter are
       selected
     gps_gps_rows$signalfilter=which(GPS$max_PPA>=input$signalhigh
     |GPS$credible_variants_per_locus_signal<=5) # reduce the indices to
      those meeting the signal filter criteria (high PPA OR small set)
    }else{
     if(isTruthy(input$signalppafilter)
        &!isTruthy(input$signalsmall)){ # only PPA filter
      gps_gps_rows$signalfilter=
       which(GPS$max_PPA>=input$signalhigh)
     }else if(!isTruthy(input$signalppafilter)
              &isTruthy(input$signalsmall)){ # only small-set filter
      gps_gps_rows$signalfilter=
       which(GPS$credible_variants_per_locus_signal<=5)
     }else if(!isTruthy(input$signalppafilter)
              &!isTruthy(input$signalsmall)){ # no filter
      gps_gps_rows$signalfilter=c(1:nrow(GPS))
     }
```

```
   }
   # restrict bun/cys signals:
   if(input$signalcysbun==TRUE){
    gps_gps_rows$signalfilter=
     intersect(gps_gps_rows$signalfilter,
     which(GPS$Cys.BUN%in%c("bun","cys","cys/bun")))
   }
   # restrict DM signals:
   if(input$signalDM==TRUE){
    gps_gps_rows$signalfilter=
     intersect(gps_gps_rows$signalfilter,
     which(GPS$DM_effect!="-"))
   }
   ## restrict decline signals:
   if(input$signaldecline==TRUE){
    gps_gps_rows$signalfilter=
     intersect(gps_gps_rows$signalfilter,
     which(GPS$decline_effect!="-"))
   }
  }
})

#reduce all GPS versions to the rows meeting the signal-filter criteria using the above defined indices
gps_version_all_show <- eventReactive(input$gpsgo,{
 GPS_show[gps_gps_rows$signalfilter,]
})

gps_version_10_show <- eventReactive(input$gpsgo,{
 GPS_10_show[gps_gps_rows$signalfilter,]
})

gps_version_50_show <- eventReactive(input$gpsgo,{
 GPS_50_show[gps_gps_rows$signalfilter,]
})


# map filter: combine all mapping filter options to one vector
map <- eventReactive(input$gpsgo,{
 if(go$gps_correct){
  if(isTruthy(input$genemapprot)& isTruthy(input$genemapeqtl)){
   c(input$genemapprot,input$genemapeqtl)
  }else if(isTruthy(input$genemapprot)
           &!isTruthy(input$genemapeqtl)){
   c(input$genemapprot)
  }else if(!isTruthy(input$genemapprot)
           &isTruthy(input$genemapeqtl)){
   c(input$genemapeqtl)
  }else if(isTruthy(input$mapnofilter)){
   NULL
  }
 }
})
```

```
# combine the additional mapping filter options (PPA, small set) to one variable
ppa_filter_gps <- eventReactive(input$gpsgo,{
 if(go$gps_correct){
  if(!is.null(map())){
   if(isTruthy(input$nofiltermapppa)){
    "no_filter"
   }else{
    if(isTruthy(input$mapppafilter)&isTruthy(input$mapsmall)){
     paste("small_", input$maphigh, sep="")
    }else{
     if(isTruthy(input$mapppafilter)&!isTruthy(input$mapsmall)){
      input$maphigh
     }else if(!isTruthy(input$mapppafilter)
              &isTruthy(input$mapsmall)){
      "small"
     }
    }
   }
  }
 }
})


#apply mapping filter to each GPS version in dependence of the additional mapping filter
"ppa_filter_gps()":
# the full GPS version is needed for small signals and single signals (PPA>99%)
gps_version_all_show_map <- eventReactive(input$gpsgo,{
 if(go$gps_correct){
  if(!is.null(map())){
   if(any(ppa_filter_gps()%in%c(0.5, 0.1))){
    NULL
   }else{
    if(any(ppa_filter_gps()
       %in%c("small", "small_0.99", "small_0.5","small_0.1"))){
     if(length(map())==1){
      gps_version_all_show()[which(gps_version_all_show()
      [,as.numeric(map())]!=0
      & gps_version_all_show()[,10] <=5),]
     }else{
      gps_version_all_show()[which(rowSums(gps_version_all_show()
      [,as.numeric(map())])!=0
      & gps_version_all_show()[,10] <=5),]
     }
    }else if(ppa_filter_gps()=="no_filter"){
     if(length(map())==1){
      gps_version_all_show()[which(gps_version_all_show()
      [,as.numeric(map())]!=0 ),]
     }else{
      gps_version_all_show()[which(rowSums(gps_version_all_show()
      [,as.numeric(map())])!=0 ),]
     }
    }else if(ppa_filter_gps()==0.99){
     if(length(map())==1){
      gps_version_all_show()[which(gps_version_all_show()
      [,as.numeric(map())]!=0
      & gps_version_all_show()[,10] ==1),]
```

```
      }else{
       gps_version_all_show()[which(rowSums(gps_version_all_show()
       [,as.numeric(map())])!=0
       & gps_version_all_show()[,10] ==1),]
      }
     }
    }
   }else{
    gps_version_all_show()
   }
  }
})

#PPA 10 version is needed for all mapping including a PPA >10% filtering (only PPA>10%
and PPA>10% OR small set)
gps_version_10_show_map <- eventReactive(input$gpsgo,{
 if(go$gps_correct) {
  if(!is.null(map())){
   if(ppa_filter_gps()%in%c("no_filter", "small_0.5",
                            "small","small_0.99", 0.99,0.5)){
    NULL
   }else{
    if(length(map())==1){
     gps_version_10_show()[which(gps_version_10_show()
     [,as.numeric(map())]!=0),]
    }else{
     gps_version_10_show()[which(rowSums(gps_version_10_show()
     [,as.numeric(map())])!=0),]
    }
   }
  }else{
   NULL
  }
 }
})

#PPA 50 version is needed for all mapping including a PPA >50% filtering (only PPA>50%
and PPA>50% OR small set)
gps_version_50_show_map <- eventReactive(input$gpsgo,{
 if(go$gps_correct){
  if(!is.null(map())){
   if(ppa_filter_gps()%in%c("no_filter", "small_0.1",
                            "small","small_0.99", 0.99, 0.1)){
    NULL
   }else{
    if(length(map())==1){
     gps_version_50_show()[which(gps_version_50_show()
     [,as.numeric(map())]!=0),]
    }else{
     gps_version_50_show()[which(rowSums(gps_version_50_show()
     [,as.numeric(map())])!=0),]
    }
   }
  }else{
   NULL
  }
```

```
  }
})

# combine gps_versions

gps_version_mapped_show <- eventReactive(input$gpsgo,{
 if(go$gps_correct){
  if(is.null(gps_version_10_show_map())
     &is.null(gps_version_50_show_map())){
   gps_version_all_show_map()
  }else if(is.null(gps_version_10_show_map())
           &is.null(gps_version_all_show_map())){
   gps_version_50_show_map()
  }else if(is.null(gps_version_all_show_map())
           &is.null(gps_version_50_show_map())){
   gps_version_10_show_map()
  }else if(!is.null(gps_version_all_show_map())
           &!is.null(gps_version_50_show_map())){
   GPS_tbl = rbind(gps_version_all_show_map(),
             gps_version_50_show_map() )
   if(any(duplicated(GPS_tbl[,c("Gene*","Signal ID")]))){
    GPS_tbl[which(!duplicated(GPS_tbl
                  [,c("Gene*","Signal ID")])),]
   }
  }else if(!is.null(gps_version_all_show_map())
           &!is.null(gps_version_10_show_map())){
   GPS_tbl = rbind(gps_version_all_show_map(),
             gps_version_10_show_map() )
   if(any(duplicated(GPS_tbl[,c("Gene*","Signal ID")]))){
    GPS_tbl[which(!duplicated(GPS_tbl
                 [,c("Gene*","Signal ID")])),]
   }
  }
 }
})



# pheno and drug filtering based on the before filtered and combined GPS versions
gps_version_mapped_pheno_show <- eventReactive(input$gpsgo,{
 if(go$gps_correct){
  if(!isTruthy(input$genelistpheno)
     &!isTruthy(input$genelistdrug)){
   gps_version_mapped_show()
  }else{
   pheno = NULL
   if(isTruthy(input$genelistpheno)){
    if(length(input$genelistpheno)==2){
     pheno = c(20,21)
    }else{
     pheno = ifelse(input$genelistpheno=="mouse",20,21)
    }
   }
   drug=NULL
   if(isTruthy(input$genelistdrug)){
    if(length(input$genelistdrug)==2){
```

```
      drug = c("yes for kidney disease", "yes for other disease")
     }else{
      drug = ifelse(input$genelistdrug=="kidney",
      "yes for kidney disease", "yes for other disease")
     }
    }
    if(is.null(pheno)){
     gps_version_mapped_show()[which(gps_version_mapped_show()
     [,"known_drug_target"] %in% drug),]
    }else if(is.null(drug)){
     if(length(pheno)==1){
      gps_version_mapped_show()[which(gps_version_mapped_show()
      [,pheno]!=0),]
     }else{
      gps_version_mapped_show()[which(rowSums
     (gps_version_mapped_show()[,pheno])!=0),]
     }
    }else if(!is.null(pheno) & ! is.null(drug)){
     if(length(pheno)==1){
      gps_version_mapped_show()[which(gps_version_mapped_show()
      [,pheno]!=0 | gps_version_mapped_show()
      [,"known_drug_target"] %in% drug),]
     }else{
      gps_version_mapped_show()[which(rowSums(
      gps_version_mapped_show()[,pheno])!=0
      |gps_version_mapped_show()[,"known_drug_target"] %in%drug),]
     }
    }
   }
  }
 })


 # additional reactive value. Introduced to handle the shown GPS version before
 "Go Prioritize" was clicked:
 go_correct <- reactiveVal(FALSE)

 observeEvent(input$gpsgo, { # when "Go Prioritize" was clicked
  if(go$gps_correct){ # and filter criteria are ok
   go_correct(TRUE) #this changes to TRUE
  }
 })

 gps_show_final <- reactive({
  if (go_correct()) {
   gps_version_mapped_pheno_show()
  }else{
   GPS_show # as long as go_correct() is FALSE the whole GPS is shown
  }
 })


 output$downloadGPSTable <- downloadHandler(
  filename = function() {
   paste("results_KidneyGPS_", Sys.Date(), ".xlsx", sep = "")
  },
```

```
  content = function(file) {
   data_to_export <- gps_show_final()[,GPS_columns]
   data_to_export[,"Gene*"] <-
    gsub('<a href=".*">|</a>', '', data_to_export[,"Gene*"])
    # excludes link from GPS Genes
   write_xlsx(data_to_export, file)
  }
 )


 GPS_columns <- match(c('Gene*','Locus name**','Locus ID',
 'Signal ID','eGFRcys or BUN validation','DM_effect',
 'decline_effect','# credible variants in signal','stop-gained,
 stop-lost, non-synonymus','canonical splice, noncoding change,
 synonymous, splice site','other deleterious variant',
 'eQTL glomerulus (NEPTUNE, or Sheng et al [Nat Genet, 2021])',
 'eQTL tubulo-interstitium (NEPTUNE, or Sheng et al [Nat Genet,
 2021])','eQTL kidney cortex (GTEx)','eQTL other tissue
 (GTEx)','sQTL kidney cortex (GTEx)','sQTL other tissue
 (GTEx)','Coloc in NEPTUNE tissue','# kidney phenotypes in
 mouse','# kidney phenotypes in human',
'known_drug_target','Distance to locus lead variant',
'Chromosome','Position gene start','Position gene end')
,names(GPS_show))


 output$GPS <- renderDataTable({ # This renders the shown GPS Table
  datatable(gps_show_final()[,GPS_columns],
   rownames = FALSE,
   options = list(
    columnDefs = list(list(className = 'dt-left',
     targets = "_all")),
    initComplete=JS(initComplete),
    headerCallback = JS(headerCallback), # define header
    dom = 'rltip' # Define the table control elements to appear on the page and in
    what order. r: processing display element, l: length changing menu, t: table, i: Table
    Information summary, P: pagination
   ),
   class ='display',
   filter = list(
    position = 'top', clear = TRUE, plain = TRUE # filter fields above the
    table columns
   ),
   escape = FALSE,
   caption = htmltools::tags$caption( # text displayed below the table
    style = 'caption-side: bottom; text-align: left;
    font-weight: normal;',
    '* Link to GeneCards, **nearest gene to locus lead-variant'),
   container = sketch
  )
 })


 output$GPS_Genes1 <- renderText({ # counts number of genes and signals in
 the displayed GPS Table
  ngene_unique <- length(unique(gps_show_final()[,'Gene*']))
```

```
 nsignal <- length(unique(paste(gps_show_final()[,'Locus ID'],
  gps_show_final()[,'Signal ID'],sep = ".")))
 paste(ngene_unique, "genes from",nsignal, "signals meet the
  applied filter criteria.")
})


observeEvent(input$question_GPS,{ # Window when the ? is clicked
 showModal(modalDialog(
  titel="Help GPS filter-options",
  tags$h4("GPS filter-options - Help:"),
  p("This tab allows to generate customized gene lists. Select
   filter options from all three sections and click 'Go prioritze'
   to generate your gene list. When all 5906 genes in all 594
   signals should be shown, the 'no filter' options have to be
   selected."),
  easyClose = TRUE,
  footer=NULL,
  size = "m"
  )
 )
})


# region page ----------------------------------------------------
# definition of GPS table header:
sketch2 = htmltools::withTags(table(
 class = 'display',
  thead(
   tr(
    th(rowspan = 2, class="Geneofficial hover", 'Gene*',
     span(class="Genecomment", "Official gene name from HGNC")),
    th(rowspan = 2, 'Locus name**'),
    th(rowspan = 2, class="Locusidwithcomment hover",'Locus ID',
     span(class="Locusidcomment", "k: known locus from Wuttke et
     al. Nat.commun.2019, n: novel locus in Stanzick et al.
     Nat.commun. 2021")),
    th(rowspan = 2, class="Signalidwithcomment hover",'Signal ID',
     span(class="Signalidcomment", "ID of independent signals
     within a locus")),
    th(rowspan = 2, class="validationwithcomment hover",'eGFRcys
     or BUN validation', span(class="validationcomment",
     list( "Information whether the locus lead variant is nominal
      significant (P<0.05) associated with concordent effect
      direction with ", tags$abbr(title="glomerular filtration
      rate estimated from serum cystatin C","eGFRcys"),
      "or ",tags$abbr(title="blood urea nitrogen","BUN")))),
    th(rowspan = 2, class="DMwithcomment hover", 'Signal
     association depends on DM', span(class="DMcomment",
     list( "Information whether the signal index variant (or a
      correlated variant) shows different effect on eGFR in
      individuals with ", tags$abbr(title="Diabetes
      mellitus","DM"), "-status, Winkler et al. 2022"))),
    th(rowspan = 2, class="declinewithcomment hover",
     'Signal association with eGFRcrea decline',
     span(class="declinecomment",list( "Information whether the
```

```
        signal index variant (or a correlated variant) was
        established with annual eGFR-decline, Gorski et al.
        2022"))),
      th(rowspan = 2, '# credible variants in signal'),
      th(rowspan = 2, class="maxPPA hover",'max PPA',
       span(class="maxPPAcomment", "max. probability of a credible
       variant in this signal to be causal")),
      th(colspan = 3, class="CADD", span('# protein-relevant
       credible variants in the gene')),
      th(colspan = 6, class="eqtl", '# credible variants that
       modulate gene expression (eQTLs) or splicing (sQTLs), FDR <
       5%'),
      th(rowspan = 2, class="eqtl", 'Coloc in NEPTUNE tissue'),
      th(rowspan = 2, class="pheno",'# kidney phenotypes in mouse'),
      th(rowspan = 2, class="pheno", '# kidney phenotypes in human'),
      th(rowspan = 2, class ="drug", 'Gene is known as drug target
       or for drug interaction'),
      th(rowspan = 2, 'Distance to locus lead variant'),
      th(rowspan = 2, 'Chromosome'),
      th(rowspan = 2, 'Position gene start'),
      th(rowspan = 2, 'Position gene end')
     ),
     tr(
      th('stop-gained, stop-lost, non-synonymous'),
      th('canonical splice, noncoding change, synonymous,
       splice site'),
      th(span('other deleterious variant (CADD-Phred',
       HTML('<span>&#8805;</span>'),'15)')),
      th(class='eQTLglowithcomment hover', 'eQTL glomerulus',
       span(class='eQTLglocomment','NEPTUNE, or Sheng et al.
       [Nat.Genet. 2021]')),
      th(class='eQTLtubwithcomment hover', 'eQTL tubulo-
       interstitium', span(class='eQTLtubcomment','NEPTUNE, or
       Sheng et al. [Nat.Genet. 2021]')),
      th('eQTL kidney cortex (GTEx)'),
      th('eQTL other tissue (GTEx)'),
      th('sQTL kidney cortex (GTEx)'),
      th('sQTL other tissue (GTEx)')
     )
    )
))


headerCallback2 <- "function( thead, data, start, end, display) {
 $(thead).closest('thead').find('th').eq(9).css('background-
  color', '#ccffff');
 $(thead).closest('thead').find('th').eq(10).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(11).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(12).css('background-
  color', '#b3ffb3');
 $(thead).closest('thead').find('th').eq(13).css('background-
  color', '#b3ffb3');
 $(thead).closest('thead').find('th').eq(14).css('background-
  color', '#d9b3ff');
```

```
 $(thead).closest('thead').find('th').eq(19).css('background-
  color', '#ccffff');
 $(thead).closest('thead').find('th').eq(20).css('background-
  color', '#ccffff');
 $(thead).closest('thead').find('th').eq(21).css('background-
  color', '#ccffff');
 $(thead).closest('thead').find('th').eq(22).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(23).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(24).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(25).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(26).css('background-
  color', '#ffdb99');
 $(thead).closest('thead').find('th').eq(27).css('background-
  color', '#ffdb99');
}"


observe(
 if(input$pos_start!=0&input$pos_end!=0){ # only when there is input show
  the output section
  shinyjs::showElement(id="region_go_div")
 }
)

#combine ENTER and "go" click to one reactive value only reacting if there is region input
go <- reactiveValues('region'=c())

observeEvent(input$region_go, {
 if(input$pos_start!=0&input$pos_end!=0){
  if(is.null(go$region)){go$region=1}
  else{go$region <- go$region+1} #count +1 when the go button is clicked
 }
})

observeEvent(input$keyPressed, {
 if(input$pos_start!=0&input$pos_end!=0){
  if(is.null(go$region)){go$region=1}
  else{go$region <- go$region+3} #count +3 when the when enter is clicked
 }
})

region_start <- eventReactive(go$region,{input$pos_start})
region_end <- eventReactive(go$region,{input$pos_end})

valid_reg <- eventReactive( go$region,{( # check if a valid input was made
 if(!is.na(region_start())& !is.na(region_end())){
  (is.integer(region_start()) & is.integer(region_end())&
   input$pos_start>0 & input$pos_end>0 &
   input$pos_end>input$pos_start &
   input$pos_start!=input$pos_end)}
 else{FALSE}
)})
```

```
chr <- eventReactive(go$region,{input$Chromosome})

output$valid_gene_test <- renderText({
 if(!valid_reg()){
  if(is.na(region_start())){
   "Error: invalid region start position"}
  else{
   if(!is.integer(region_start())|region_start()<0) {
    "Error: invalid region start position"}
   else{
    if(is.na(region_end())){"Error: invalid region end position"}
    else{
     if(!is.integer(region_end())| region_end()<0) {
      "Error: invalid region end position"}
     else{
      if(region_end()<region_start()) {"Error: region start must
       be smaller than region end position"}
      else{
       if(region_end()==region_start()){
        "Error: region start and end must not be the same position"
       }else{"undefined error"}
      }
     }
    }
   }
  }
 }
})


ov_loci<- reactive({ #search for overlapping loci
 if(valid_reg()){
  region_table_show[which(chr()==as.numeric(region_table$chr)
   &(region_start()<=region_table$region_start_250kb
   & region_end()>=region_table$region_start_250kb)
   |(region_start()>=region_table$region_start_250kb
   & region_start()<=region_table$region_end_250kb))),]
 }else{0}
})

overlap_loci <- eventReactive(go$region,{nrow(ov_loci())!=0})
# TRUE or FALSE if there are overlapping loci

output$ov_loci <- DT::renderDT(server=FALSE, {
# rendering the region table of the overlapping loci
 if(valid_reg() & overlap_loci()){
  datatable(ov_loci(),rownames=FALSE,
   options = list(
    columnDefs = list(list(className = 'dt-left',
     targets = "_all")),
    scrollX=TRUE, scrollCollapse = TRUE,
    dom = 'lfrtBip',
    buttons= list(c('copy', 'excel','csv'))
   ),
   extensions = 'Buttons',
```

```
    caption = htmltools::tags$caption(
     style = 'caption-side: bottom; text-align: left;
     font-weight: normal;',
     '* SNPs, which are associated with log(eGFRcrea) in the all-
     ancestry GWAS meta-analysis (unconditioned) with P<5E-8')
   )
  }
 })

 output$ov_loci_test <- renderText({
  if(valid_reg()& !overlap_loci()) {
   "no eGFRcrea associated loci found in this region"
  }else{
   if(valid_reg() & overlap_loci()){
    paste(length(unique(ov_loci()[,1])),"  loci (including
     ",nrow(ov_loci()),"  signals) have been found in your searched
     region.", sep="")
   }
  }
 })

 GPS_columns_region <- match(c('Gene*','Locus name**','Locus ID',
  'Signal ID', 'eGFRcys or BUN validation',
  'DM_effect','decline_effect','# credible variants in signal',
  'max PPA', 'stop-gained, stop-lost, non-synonymus',
  'canonical splice, noncoding change, synonymous, splice site',
  'other deleterious variant','eQTL glomerulus (NEPTUNE, or Sheng
  et al [Nat Genet, 2021])','eQTL tubulo-interstitium (NEPTUNE, or
  Sheng et al [Nat Genet, 2021])','eQTL kidney cortex (GTEx)','eQTL
  other tissue (GTEx)','sQTL kidney cortex (GTEx)','sQTL other
  tissue (GTEx)','Coloc in NEPTUNE tissue','# kidney phenotypes in
  mouse','# kidney phenotypes in human', 'known_drug_target',
  'Distance to locus lead variant','Chromosome',
  'Position gene start', 'Position gene end'),names(GPS_show))

 GPS_region_search <- reactive({
  if(valid_reg()&overlap_loci()){
   x = GPS_show[which(
    GPS$locus_id%in%ov_loci()[,1]),GPS_columns_region]
   y = GPS[which(GPS$locus_id%in%ov_loci()[,1]),]
   v=c()
   for(i in 1:nrow(y)){
    # check how many non zero annotation columns per gene there are
    v[i] = 11-length(which(y[i,c(9:19)]==0))
   }
   # order GPS Table that Genes with most entries are at the top
   x[order(v,decreasing=T),]
  }else{NULL}
 })

 output$GPS_region_search <-
 DT::renderDT(server=FALSE, {  #render GPS Table
  if(valid_reg()& !is.null(GPS_region_search())){
   datatable(GPS_region_search(),rownames=FALSE,
    options = list(
     columnDefs = list(list(className = 'dt-left',
```

```
      targets = "_all")),
     scrollX=TRUE, scrollCollapse = TRUE,
     headerCallback = JS(headerCallback2),
     dom = 'lfrtBip',
     buttons= list(c('copy', 'excel','csv'))
    ),
    extensions= 'Buttons',
    class ='display',
    filter = list(
     position = 'top', clear = TRUE, plain = TRUE),
    escape = FALSE,
    caption = htmltools::tags$caption(
     style = 'caption-side: bottom; text-align: left;
      font-weight: normal;',
      '* Link to GeneCards, **nearest gene to locus lead-variant'),
    container = sketch2
   )%>% formatPercentage('max PPA') #format number to percentage
  }
})

# show results section when search is started
observeEvent(go$region,{shinyjs::showElement(id="region_div")})

observeEvent(go$region, ({
 if(valid_reg()){
  updateCollapse(session, "regionCollapse", open = c("Overlapping
   genetic eGFRcrea signals")) # open region search section
  if(!is.null(GPS_region_search())){
   updateCollapse(session, "regionCollapse", open = c("Overlapping
    genetic eGFRcrea signals", "GPS entries for genes in
    overlapping eGFRcrea loci")) # open GPS section, when there is a
    overlapping locus
  }
 }
}))


observe({
 shinyjs::toggleElement(id="ov_genes_down",
 condition = (!is.null(GPS_region_search())) ) #make GPS Table visible
})

#help window handling:
observeEvent(input$question_region_search,{
 showModal(modalDialog(
  titel="Help region search",
  tags$h4("Region search - Help:"),
  p("Select a chromosome and enter a start/end position [bp] of a
   region you are interested in. The results show you a list of
   overlapping eGFRcrea loci and the GPS entries for the respective
   genes if there are any. Attation: The region end must be larger
   than region start and region start can't be 0. For further
   information check the Documentation page."),
  easyClose = TRUE,
  footer=NULL,
  size = "m"
```

```
 ))
})

# snp page ----------------------------------------------------

#define input type based on click for further analyses
inputtype <- reactiveValues('snp'=c())
onclick("snp", (inputtype$snp=c("single")), add=T)
onclick("SNP_batch", (inputtype$snp=c("list")), add=T)
onclick("snp-upload-section", (inputtype$snp=c("upload")), add=T)

#show search button:

observe(
 if(isTruthy(input$file1)|isTruthy(input$snp)
    |isTruthy(input$SNP_batch)){
  shinyjs::showElement(id="snp_go_div")
 }
)


#combine ENTER and "go" click to one reactive value only reacting if there is snp input
go <- reactiveValues('snp'=c())
#update the value of snp$go when go or enter are clicked:
observeEvent(input$snp_go, {
 if(isTruthy(input$file1)|isTruthy(input$snp)
    |isTruthy(input$SNP_batch)){
  if(is.null(go$snp)){go$snp=1}
  else{go$snp <- go$snp+1}
 }
})

observeEvent(input$keyPressed, {
 if(isTruthy(input$file1)|isTruthy(input$snp)
    |isTruthy(input$SNP_batch)){
  if(is.null(go$snp)){go$snp=1}
  else{go$snp <- go$snp+3}
 }
})

#upload options visible when there is an input file
observe({
 shinyjs::toggleElement(id="SNP_batch_upload_options",
  condition = (isTruthy(input$file1)) )
})

#disable credvar only functions, when not searched for cred vars
observe({
 if(any(input$`variant-search-options`=="credvar")){
  output$detail.evidence.snp <- renderUI(
   div(class="snpinputbox",
    checkboxGroupInput(inputId="detail_evidence_variant",
     label=span(class="snpsearch","Specification of additional
      functional SNP information",
      span(class="snpsearchcomment","functional information only
       available for credible variants")),
```

```
       selected = c("CADD","NEPTUNE_glo","NEPTUNE_tub",
                    "GTEx_eQTL","GTEx_sQTL"),
       inline = FALSE, width = NULL,
       choiceNames = c("deleteriousness and functional annotation
        (CADD)","eQTL in glomerulus (NEPTUNE, or Sheng et al
        [Nat.Genet. 2021])","eQTL in tubulo-interstitium (NEPTUNE,
        or Sheng et al [Nat.Genet. 2021])","eQTL in kidney cortex
        (GTEx)","sQTL in kidney cortex (GTEx)","eQTL in other
        tissues (GTEx)","sQTL in other tissues (GTEx)", "eGFR
        association statistics separated by diabetes status"),
       choiceValues = c("CADD","NEPTUNE_glo","NEPTUNE_tub",
        "GTEx_eQTL","GTEx_sQTL","GTEx_wo_kidney_eQTL",
        "GTEx_wo_kidney_sQTL","dm_stat")),
 ))}else{
  output$detail.evidence.snp <- renderUI(
   div(class="snpinputbox",disabled(
    checkboxGroupInput(inputId="detail_evidence_variant",
     label=span(class="snpsearch","Specification of additional
      functional SNP information or locus based
      information",span(class="snpsearchcomment","functional
      information only available for credible variants")),
     inline = FALSE, width = NULL,
     choiceNames = c("deleteriousness and functional annotation
      (CADD)","eQTL in glomerulus (NEPTUNE, or Sheng et al
      [Nat.Genet. 2021])","eQTL in tubulo-interstitium (NEPTUNE,
      or Sheng et al [Nat.Genet. 2021])","eQTL in kidney cortex
      (GTEx)","sQTL in kidney cortex (GTEx)","eQTL in other
      tissues (GTEx)","sQTL in other tissues (GTEx)","eGFR
      association statistics separated by diabetes status"),
     choiceValues =  c("CADD","NEPTUNE_glo","NEPTUNE_tub",
      "GTEx_eQTL","GTEx_sQTL",
      "GTEx_wo_kidney_eQTL","GTEx_wo_kidney_sQTL","dm_stat"))),
  ))
 }
})

###first snp input processing:
single_snp <- eventReactive(go$snp, {
 gsub(" ","",input$snp) # exclude spaces
})

#build vector of snps from text input
input_snp_batch_list <- eventReactive(go$snp,{
 req(input$SNP_batch) #only do the following when there is input
 # excludes all non-word characteres except ":" that might be used as separators:
 ivbl=unlist(strsplit(input$SNP_batch,
  "[^[:alnum:]^:]",perl=TRUE))
 if(any(ivbl=="")){
  ivbl[-which(ivbl=="")]
 }else{ivbl}
})


#build vector of snps from file input
input_snp_batch_upload <- reactiveValues('input'= c())
```

```
observe({
 req(input$file1)
 input_snp_batch_upload$input <- scan(file=input$file1$datapath,
                                      sep = input$sep_snp,
                                      what="character")
})


#reset file input, when reset button is clicked
observeEvent(input$reset_variant,{
 reset("file1")
 input_snp_batch_upload$input <- NULL
})

#final selection and save searched snps in "snp()":
snp <- eventReactive(go$snp, {
 if(inputtype$snp=="single"){
  single_snp()
 }else{
  if(inputtype$snp=="list"){
   input_snp_batch_list()
  }else{
   if(inputtype$snp=="upload"){
    input_snp_batch_upload$input
   }else{NULL}
  }
 }
})

inputtype <- reactiveValues('snp_final'=c())

observeEvent(go$snp,{
 inputtype$snp_final = inputtype$snp
})

#snp page - output generation:
#search for overlap with credible variants
cred_vars <- eventReactive(go$snp, {
 if(!is.null(snp())){
  snp1 = snp()
  if(any(grepl(":", snp1))){ # do this when chr:pos is used as input
   for(i in grep(":", snp1)){
    chr=unlist(strsplit(snp1[i],":"))[1]
    pos=unlist(strsplit(snp1[i],":"))[2]
    if(any(cred_var$chr==chr & cred_var$pos==pos)){
     snp1[i] = cred_var$rsid[which(cred_var$chr==chr
      & cred_var$pos==pos)] # replace searched input (chr:pos) with rsids
    }
   }
  }
  if(any(snp1%in%cred_var$rsid)){
   snp_in_cred=(snp1%in%cred_var$rsid)
   # restrict to SNPs included in 99% credible set variants
   snp1[which(snp_in_cred)]
  }else{NULL}
 }else{NULL}
```

```
})

valid <- eventReactive(go$snp, {!is.null(cred_vars())})

detail_evidence_variant <- eventReactive(go$snp, {
 input$detail_evidence_variant})

gws_snps <- eventReactive(go$snp,{
 if(!is.null(snp())){
  if(any(snp()%in%all_sig$RSID)){
   snp_in_gws = (snp()%in%all_sig$RSID)
   snp()[which(snp_in_gws)] # search for overlap with genome-wide significant
   variants
  }else{NULL}
 }else{NULL}
})

gws<-eventReactive(go$snp, {!is.null(gws_snps())})

# open all relevant divs:
observeEvent(go$snp,{shinyjs::showElement(id="SNP_div")})
observeEvent(go$snp,{runjs('
 document.getElementById("SNP_div").scrollIntoView({ left: 0,
 block: "end", behavior: "smooth" });
 ')}) # this java-script command leads to automatic scrolling

observeEvent(go$snp,{shinyjs::toggleElement(id="gws_div",
 condition = any(input$`variant-search-options`=="gws"))})

observeEvent(go$snp,{shinyjs::toggleElement(id="cred_var_div",
 condition = any(input$`variant-search-options`=="credvar"))})

observeEvent(go$snp,{shinyjs::toggleElement(id="div_cadd_snp",
 condition= (any(detail_evidence_variant()=="CADD")&valid()))})

observeEvent(go$snp,{shinyjs::toggleElement(id="div_glo_snp",
 condition= (any(detail_evidence_variant()=="NEPTUNE_glo")
  &valid()))})

observeEvent(go$snp,{shinyjs::toggleElement(id="div_tub_snp",
 condition= (any(detail_evidence_variant()=="NEPTUNE_tub")
  &valid()))})

observeEvent(go$snp,{shinyjs::toggleElement(
 id="div_GTEx_eqtl_snp",
 condition= (any(detail_evidence_variant()=="GTEx_eQTL")
  &valid()))})

observeEvent(go$snp,{shinyjs::toggleElement(
 id="div_GTEx_sqtl_snp",
 condition= (any(detail_evidence_variant()=="GTEx_sQTL")
  &valid()))})

observeEvent(go$snp,{shinyjs::toggleElement(
 id="div_GTEx_eqtl_wo_kidney_snp",
 condition= (any(detail_evidence_variant()==
```

```
  "GTEx_wo_kidney_eQTL")&valid())})

 observeEvent(go$snp,{shinyjs::toggleElement(
  id="div_GTEx_sqtl_wo_kidney_snp",
  condition= (any(detail_evidence_variant()==
   "GTEx_wo_kidney_sQTL")&valid()))})

 observeEvent(go$snp,{shinyjs::toggleElement(id="div_dm_stat_snp",
  condition= (any(detail_evidence_variant()=="dm_stat")
  &valid()))})

 observeEvent(go$snp,{shinyjs::toggleElement(
  id="div_locus_zoom_snp",
  condition= (any(input$snp.locuszoom=="locus_zoom")&valid()))})


 #generate text output how many searched snps are genome-wide significant associated
 output$namecheck <- renderText({
  if(!is.null(snp())){
   if(inputtype$snp_final=="single"){
    if(gws()){
     paste(snp(), "is associated with log(eGFRcrea) in at least one
      of the 424 loci at P<5E-8 (all-ancestry GWAS meta-analysis,
      unconditioned):", sep=" ")
    }else{
     paste(snp(), "is not included in the analysis, or not
      associated with log(eGFRcrea) in one of the 424 loci at
      P<5E-8 (all-ancestry GWAS meta-analysis, unconditioned)."
      , sep=" ")
    }
   }else{
    if(gws()){
     if(length(gws_snps())>1){
      paste("Of your ",length(snp()), " queried SNPs, ",
       length(gws_snps()), " SNPs are associated with log(eGFRcrea)
       in at least one of the 424 loci at P<5E-8 (all-ancestry GWAS
       meta-analysis, unconditioned):", sep="")
     }else{
      paste("Of your ",length(snp()), " queried SNPs, ",
       length(gws_snps()), " SNP (", gws_snps(),") is associated
       with log(eGFRcrea) in at least one of the 424 loci at
       P<5E-8 (all-ancestry GWAS meta-analysis, unconditioned):",
       sep="")
     }
    }else{
     paste("None of your ",length(snp()), " queried SNPs is
      associated with log(eGFRcrea) in one of the 424 loci at
      P<5E-8 (all-ancestry GWAS meta-analysis, unconditioned).",
      sep="")
    }
   }
  }else{
   "Please check if you correctly entered or uploaded your SNPs and
   if the right input field is selected."
  }
 })
```

```
#render table of all-ancestry statistics
output$all_var <- DT::renderDT(server=FALSE, {
 if(gws()){
  datatable(all_sig_show[which(all_sig$RSID%in%gws_snps()),
   c(15,11,12,2:8,22,23,24)], rownames=FALSE,
   options = list(
    columnDefs = list(list(className = 'dt-left',
     targets = "_all")),
    scrollX=TRUE, scrollCollapse = TRUE,
    dom = 'lfrtBip',
    buttons= list(c('copy', 'excel','csv'))
   ),
   extensions = 'Buttons',
   escape=FALSE,
   caption = htmltools::tags$caption(
    style = 'caption-side: bottom; text-align: left;
     font-weight: normal;',
    htmltools::tags$p('* Link to dbSNP'),
    htmltools::tags$p('** adjusted for age, sex and other
     study-specific covariates'),
    htmltools::tags$p('Effect allele: eGFRcrea lowering allele,
     N: Sample size of the all-ancestry GWAS meta-analysis for
     this SNP, EAF: Effect allele frequency'),
    htmltools::tags$p('beta: Effectsize of the effect allele on
     log(eGFRcrea)')
   ),
   class='display',
   container = sketchallsig
  )
 }
})

#generate text output of how many searched SNPs are credible set variants
output$namecheck2 <- renderText({
 if(!is.null(snp())){
  if(inputtype$snp_final=="single"){
   if(valid()){
    paste(snp(), "is a 99% credible variant in at least one of the
     594 eGFRcrea signals (EUR). Shown are results from GWAS meta-
     analyses in European ancestry:", sep=" ")
   }else{
    paste(snp(), "is not a 99% credible variant in any of the 594
     eGFRcrea signals (EUR).", sep=" ")
   }
  }else{
   if(gws()){
    if(length(cred_vars())>1){
     paste("Of your ",length(snp()), " queried SNPs, ",
      length(cred_vars()), " SNPs are 99% credible variants in at
      least one of the 594 eGFRcrea signals (EUR). Shown are
      results from GWAS meta-analyses in European ancestry:",
      sep="")
    }else{
     paste("Of your ",length(snp()), " queried SNPs, ",
      length(cred_vars()), " SNP (", cred_vars(),") is a 99%
```

```
        credible variant in at least one of the 594 eGFRcrea signals
        (EUR). Shown are results from GWAS meta-analyses in European
        ancestry:", sep="")
      }
     }else{
      paste("None of your ",length(snp()), " queried SNPs is a 99%
       credible variant in any of the 594 eGFRcrea signals (EUR).",
       sep="")
     }
    }
   }
  })

  # render credible set table (EUR association statistics conditioned and unconditioned)
  output$cred_var <- DT::renderDT(server=FALSE, {
   if(valid()){
    datatable(cred_var_show[which(cred_var$rsid%in%cred_vars()),],
     rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
      dom = 'lfrtBip',
      buttons= list(c('copy', 'excel','csv'))
     ),
     extensions = 'Buttons',
     escape=FALSE,
     caption = htmltools::tags$caption(
      style = 'caption-side: bottom; text-align: left;
       font-weight: normal;',
      htmltools::tags$p('* Link to dbSNP'),
      htmltools::tags$p('PPA: posterior probability of association
       of the SNP (probability of driving the association of the
       respective signal)'),
      htmltools::tags$p('N: Sample size of the European-ancestry
       GWAS meta-analysis for this SNP, Effect allele: eGFRcrea
       lowering allele, EAF: Effect allele frequency'),
      htmltools::tags$p('Unconditioned association statistics
       origin from a european ancestry only GWAS meta-analysis for
       eGFRcrea. For loci with multiple independent signals, the
       conditioned association statistics origin from conditioning
       on the signal index variants of the other signals in the
       locus.')
     ),
     class='display',
     container = sketchcredvar
    )%>% formatPercentage('PPA')
   }
  })

  # Text if variants have functional consequences
  output$extras <-  renderText({
   if(valid()){
    if(length(detail_evidence_variant())!=0){
     paste("The following functional evidence exists for the ",
      length(cred_vars())," 99% credible variant(s) included in your
```

```
      search:",sep="")
    }
   }else{
    "Your request didn't include any 99% credible variant so no
    search for functional evidence can be performed."
   }
  })


#output CADD
output$CADD_text <- renderText({
 if(any(detail_evidence_variant()=="CADD")){
  if(any(CADD$RSID%in%cred_vars())){
   if(length(cred_vars())==1){
    paste("Of your ",length(cred_vars())," searched SNP that is a
     credible variant, ",length(which(cred_vars()%in%CADD$RSID)),
     " SNP resides in a gene and is predicted deleterious:",sep="")
   }else{
    paste("Of your",length(cred_vars()) ," searched SNPs that are
     credible variants, ",
     length(which(cred_vars()%in%CADD$RSID))," SNP(s) reside(s)
     in a gene and is/are predicted deleterious:",sep="")
   }
  }else{
   if(length(cred_vars())==1){
    paste("Your ",length(cred_vars())," searched SNP, that is a
     credible variant, does not reside in a gene or is not
     predicted deleterious.",sep="")
   }else{
    paste("None of your ",length(cred_vars())," searched SNPs,
     that are credible variants, resides in a gene and is predicted
     deleterious.",sep="")
   }
  }
 }
})

output$CADD <- DT::renderDT(server=FALSE, {
 if(valid()){
  if(any(detail_evidence_variant()=="CADD")){
   if(any(CADD$RSID%in%cred_vars())){
    x = CADD_show[which(CADD$RSID%in%cred_vars()),]
    if(any(duplicated(x))) x = x[-which(duplicated(x)),]
    datatable(x, rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
      dom = 'lfrtBip',
      buttons= list(c('copy', 'excel','csv'))
     ),
     extensions = 'Buttons',
     caption = htmltools::tags$caption(
      style = 'caption-side: bottom; text-align: left;
       font-weight: normal;',
      htmltools::tags$p('PPA: posterior probability of association
```

```
        of the SNP (probability of driving the association of the
        respective signal)'),
       htmltools::tags$p(' Reference and alternative allele from
        human reference genome (GRCh37)')
      ),
      class ='display',
      container = sketchcadd
     )%>% formatPercentage('PPA')
    }
   }
  }
})

# Neptune + Susztak glomerulus
output$glo_text <- renderText({
 if(any(detail_evidence_variant()=="NEPTUNE_glo")){
  if(any(glo_show$RSID%in%cred_vars())){
   if(length(cred_vars())==1){
    paste("Of your ",length(cred_vars())," searched SNP that is a
     credible variant, ",
     length(which(cred_vars()%in%glo_show$RSID))," SNP modulates
     gene expression in glomerular tissue:",sep="")
   }else{
    paste("Of your",length(cred_vars()) ," searched SNPs that are
     credible variants, ",
     length(which(cred_vars()%in%glo_show$RSID))," SNP(s)
     modulate(s) gene expression in glomerular tissue:",sep="")
   }
  }else{
   if(length(cred_vars())==1){
    paste("Your ",length(cred_vars())," searched SNP, that is a
     credible variant, does not modulate gene expression in
     glomerular tissue.",sep="")
   }else{
    paste("None of your ",length(cred_vars())," searched SNPs,
    that are credible variants, modulates gene expression in
    glomerular tissue.",sep="")
   }
  }
 }
})

output$glo <- DT::renderDT(server=FALSE, {
 if(valid()){
  if(any(detail_evidence_variant()=="NEPTUNE_glo")){
   if(any(glo_show$RSID%in%cred_vars())){
    datatable(glo_show[which(glo_show$RSID%in%cred_vars()),],
     rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
      dom = 'lfrtBip',
      buttons= list(c('copy', 'excel','csv'))
     ),
     extensions = 'Buttons',
```

```
      caption = htmltools::tags$caption(
       style = 'caption-side: bottom; text-align: left;
        font-weight: normal;',
       htmltools::tags$p('* eQTL effect allele is the eGFRcrea
        lowering allele'),
       htmltools::tags$p('PPA: posterior probability of association
        of the SNP (probability of driving the association of the
        respective signal), EAF: Effect allele frequency'),
       htmltools::tags$p('** NEPTUNE, or Sheng et al, Nat.Genet.
        2021')
      )
     )%>% formatPercentage('PPA')
    }
   }
  }
})

# tubulo-interstitial
output$tub_text <- renderText({
 if(any(detail_evidence_variant()=="NEPTUNE_tub")){
  if(any(tub_show$RSID%in%cred_vars())){
   if(length(cred_vars())==1){
    paste("Of your ",length(cred_vars())," searched SNP that is a
     credible variant, ",
     length(which(cred_vars()%in%tub_show$RSID))," SNP modulates
     gene expression in tubulo-interstitial tissue:",sep="")
   }else{
    paste("Of your",length(cred_vars()) ," searched SNPs that are
     credible variants, ",
     length(which(cred_vars()%in%tub_show$RSID))," SNP(s)
     modulate(s) gene expression in tubulo-interstitial
     tissue:",sep="")
   }
  }else{
   if(length(cred_vars())==1){
    paste("Your ",length(cred_vars())," searched SNP, that is a
     credible variant, does not modulate gene expression in tubulo-
     interstitial tissue.",sep="")
   }else{
    paste("None of your ",length(cred_vars())," searched SNPs,
     that are credible variants, modulates gene expression in
     tubulo-interstitial tissue.",sep="")
   }
  }
 }
})

output$tub <- DT::renderDT(server=FALSE, {
 if(valid()){
  if(any(detail_evidence_variant()=="NEPTUNE_tub")){
   if(any(tub_show$RSID%in%cred_vars())){
    datatable(tub_show[which(tub_show$RSID%in%cred_vars()),],
     rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
```

```
        scrollX=TRUE, scrollCollapse = TRUE,
        dom = 'lfrtBip',
        buttons= list(c('copy', 'excel','csv'))
       ),
       extensions = 'Buttons',
       caption = htmltools::tags$caption(
        style = 'caption-side: bottom; text-align: left;
         font-weight: normal;',
        htmltools::tags$p('* eQTL effect allele is the eGFRcrea
         lowering allele'),
        htmltools::tags$p('PPA: posterior probability of association
         of the SNP (probability of driving the association of the
         respective signal), EAF: Effect allele frequency'),
        htmltools::tags$p('** NEPTUNE, or Sheng et al, Nat.Genet.
         2021')
       )
      )%>% formatPercentage('PPA')
    }
   }
  }
 })

 #GTEx eQTL kidney
 output$eqtl_text <- renderText({
  if(any(detail_evidence_variant()=="GTEx_eQTL")){
   if(any(GTEx_eQTL$rs_id_dbSNP151_GRCh38p7%in%cred_vars())){
    if(length(cred_vars())==1){
     paste("Of your ",length(cred_vars())," searched SNP that is a
      credible variant, ",length(which(cred_vars()
      %in%GTEx_eQTL$rs_id_dbSNP151_GRCh38p7))," SNP modulates gene
      expression in kidney cortex tissue:",sep="")
    }else{
     paste("Of your",length(cred_vars()) ," searched SNPs that are
      credible variants, ",length(which(cred_vars()
      %in%GTEx_eQTL$rs_id_dbSNP151_GRCh38p7))," SNP(s) modulate(s)
      gene expression in kidney cortex tissue:",sep="")
    }
   }else{
    if(length(cred_vars())==1){
     paste("Your ",length(cred_vars())," searched SNP, that is a
      credible variant, does not modulate gene expression in kidney
      cortex tissue.",sep="")
    }else{
     paste("None of your ",length(cred_vars())," searched SNPs,
     that are credible variants, modulates gene expression in kidney
     cortex tissue.",sep="")
    }
   }
  }
 })

 output$eqtl <- DT::renderDT(server=FALSE, {
  if(valid()){
   if(any(detail_evidence_variant()=="GTEx_eQTL")){
    if(any(GTEx_eQTL$rs_id_dbSNP151_GRCh38p7%in%cred_vars())){
     datatable(GTEx_eQTL_show[
```

```
      which(GTEx_eQTL$rs_id_dbSNP151_GRCh38p7%in%cred_vars()),],
      rownames=FALSE,
      options = list(
       columnDefs = list(list(className = 'dt-left',
        targets = "_all")),
       scrollX=TRUE, scrollCollapse = TRUE,
       dom = 'lfrtBip',
       buttons= list(c('copy', 'excel','csv'))
      ),
      extensions = 'Buttons',
      caption = htmltools::tags$caption(
       style = 'caption-side: bottom; text-align: left;
        font-weight: normal;',
       htmltools::tags$p('* eQTL effect allele is the eGFRcrea
        lowering allele'),
       htmltools::tags$p('PPA: posterior probability of association
        of the SNP (probability of driving the association of the
        respective signal), EAF: Effect allele frequency'),
      )
     )%>% formatPercentage('PPA')
    }
   }
  }
 })

 #GTEx sQTL kidney
 output$sqtl_text <- renderText({
  if(any(detail_evidence_variant()=="GTEx_sQTL")){
   if(any(GTEx_sQTL$rs_id_dbSNP151_GRCh38p7%in%cred_vars())){
    if(length(cred_vars())==1){
     paste("Of your ",length(cred_vars())," searched SNP that is a
      credible variant, ",       length(which(cred_vars()
      %in%GTEx_sQTL$rs_id_dbSNP151_GRCh38p7))," SNP modulates gene
      splicing in kidney cortex tissue:",sep="")
    }else{
     paste("Of your",length(cred_vars()) ," searched SNPs that are
      credible variants, ",length(which(cred_vars()
      %in%GTEx_sQTL$rs_id_dbSNP151_GRCh38p7))," SNP(s) modulate(s)
      gene splicing in kidney cortex tissue:",sep="")
    }
   }else{
    if(length(cred_vars())==1){
     paste("Your ",length(cred_vars())," searched SNP, that is a
      credible variant, does not modulate gene splicing in kidney
      cortex tissue.",sep="")
    }else{
     paste("None of your ",length(cred_vars())," searched SNPs,
      that are credible variants, modulates gene splicing in kidney
      cortex tissue.",sep="")
    }
   }
  }
 })

 output$sqtl <- DT::renderDT(server=FALSE, {
  if(valid()){
```

```
   if(any(detail_evidence_variant()=="GTEx_sQTL")){
    if(any(GTEx_sQTL$rs_id_dbSNP151_GRCh38p7%in%cred_vars())){
     datatable(GTEx_sQTL_show[
      which(GTEx_sQTL$rs_id_dbSNP151_GRCh38p7%in%cred_vars()),],
      rownames=FALSE,
      options = list(
       columnDefs = list(list(className = 'dt-left',
        targets = "_all")),
       scrollX=TRUE, scrollCollapse = TRUE,
       dom = 'lfrtBip',
       buttons= list(c('copy', 'excel','csv'))
      ),
      extensions = 'Buttons',
      caption = htmltools::tags$caption(
       style = 'caption-side: bottom; text-align: left;
        font-weight: normal;',
       htmltools::tags$p('* sQTL effect allele is the eGFRcrea
        lowering allele'),
       htmltools::tags$p('PPA: posterior probability of association
        of the SNP (probability of driving the association of the
        respective signal), EAF: Effect allele frequency'),
      )
     )%>% formatPercentage('PPA')
    }
   }
  }
 })

 #GTEx eQTL other tissue
 output$eqtl_wo_kidney_text <- renderText({
  if(any(detail_evidence_variant()=="GTEx_wo_kidney_eQTL")){
   if(any(gtex_without_kidney$rs_id_dbSNP151_GRCh38p7
    %in%cred_vars())){
    if(length(cred_vars())==1){
     paste("Of your ",length(cred_vars())," searched SNP that is a
      credible variant, ",length(which(cred_vars()
      %in%gtex_without_kidney$rs_id_dbSNP151_GRCh38p7))," SNP
      modulates gene expression in other tissues:",sep="")
    }else{
     paste("Of your",length(cred_vars()) ," searched SNPs that are
      credible variants, ",length(which(cred_vars()
      %in%gtex_without_kidney$rs_id_dbSNP151_GRCh38p7))," SNP(s)
      modulate(s) gene expression in other tissues:",sep="")
    }
   }else{
    if(length(cred_vars())==1){
     paste("Your ",length(cred_vars())," searched SNP, that is a
      credible variant, does not modulate gene expression in other
      tissues.",sep="")
    }else{
     paste("None of your ",length(cred_vars())," searched SNPs,
      that are credible variants, modulates gene expression in other
      tissues.",sep="")
    }
   }
  }
```

```
})

output$eqtl_wo_kidney <- DT::renderDT(server=FALSE, {
 if(valid()){
  if(any(detail_evidence_variant()=="GTEx_wo_kidney_eQTL")){
   if(any(gtex_without_kidney$rs_id_dbSNP151_GRCh38p7
    %in%cred_vars())){
    datatable(gtex_without_kidney_show[
     which(gtex_without_kidney$rs_id_dbSNP151_GRCh38p7
     %in%cred_vars()),c(1:13)], rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
      dom = 'lfrtBip',
      buttons= list(c('copy', 'excel','csv'))
     ),
     extensions = 'Buttons',
     caption = htmltools::tags$caption(
      style = 'caption-side: bottom; text-align: left;
       font-weight: normal;',
      htmltools::tags$p('* eQTL effect allele is the eGFRcrea
       lowering allele'),
      htmltools::tags$p('PPA: posterior probability of association
       of the SNP (probability of driving the association of the
       respective signal), EAF: Effect allele frequency'),
     )
    )%>% formatPercentage('PPA')
   }
  }
 }
})

#GTEx sQTL other tissue
output$sqtl_wo_kidney_text <- renderText({
 if(any(detail_evidence_variant()=="GTEx_wo_kidney_sQTL")){
  if(any(sqtl_without_kidney$rs_id_dbSNP151_GRCh38p7
   %in%cred_vars())){
   if(length(cred_vars())==1){
    paste("Of your ",length(cred_vars())," searched SNP that is a
     credible variant, ",length(which(cred_vars()
     %in%sqtl_without_kidney$rs_id_dbSNP151_GRCh38p7))," SNP
     modulates gene splicing in other tissues:",sep="")
   }else{
    paste("Of your",length(cred_vars()) ," searched SNPs that are
     credible variants, ",length(which(cred_vars()
     %in%sqtl_without_kidney$rs_id_dbSNP151_GRCh38p7))," SNP(s)
     modulate(s) gene splicing in other tissues:",sep="")
   }
  }else{
   if(length(cred_vars())==1){
    paste("Your ",length(cred_vars())," searched SNP, that is a
     credible variant, does not modulate gene splicing in other
     tissues.",sep="")
   }else{
    paste("None of your ",length(cred_vars())," searched SNPs,
```

```
      that are credible variants, modulates gene splicing in other
      tissues.",sep="")
    }
   }
  }
 })

 output$sqtl_wo_kidney <- DT::renderDT(server=FALSE, {
  if(valid()){
   if(any(detail_evidence_variant()=="GTEx_wo_kidney_sQTL")){
    if(any(sqtl_without_kidney$rs_id_dbSNP151_GRCh38p7
     %in%cred_vars())){
     datatable(sqtl_without_kidney_show[
      which(sqtl_without_kidney$rs_id_dbSNP151_GRCh38p7
      %in%cred_vars()),c(1:12)], rownames=FALSE,
      options = list(
       columnDefs = list(list(className = 'dt-left',
        targets = "_all")),
       scrollX=TRUE, scrollCollapse = TRUE,
       dom = 'lfrtBip',
       buttons= list(c('copy', 'excel','csv'))
      ),
      extensions = 'Buttons',
      caption = htmltools::tags$caption(
       style = 'caption-side: bottom; text-align: left;
        font-weight: normal;',
       htmltools::tags$p('* sQTL effect allele is the eGFRcrea
        lowering allele'),
       htmltools::tags$p('PPA: posterior probability of association
        of the SNP (probability of driving the association of the
        respective signal), EAF: Effect allele frequency'),
      )
     )%>% formatPercentage('PPA')
    }
   }
  }
 })

 # statistics separated by DM status
 output$dm_stat_snp <- DT::renderDT(server=FALSE, {
  if(valid()){
   if(any(detail_evidence_variant()=="dm_stat")){
    if(any(DM_stats$rsid%in%cred_vars())){
     datatable(DM_stats_show[
      which(DM_stats$rsid%in%cred_vars()),], rownames=FALSE,
      options = list(
       columnDefs = list(list(className = 'dt-left',
        targets = "_all")),
       scrollX=TRUE, scrollCollapse = TRUE,
       dom = 'lfrtBip',
       buttons= list(c('copy', 'excel','csv'))
      ),
      extensions = 'Buttons',
      caption = htmltools::tags$caption(
       style = 'caption-side: bottom; text-align: left;
        font-weight: normal;',
```

```
        htmltools::tags$p('N: Sample size of European-ancestry GWAS
         meta-analysis for this SNP, Effect allele: eGFRcrea
         lowering allele in the combined analysis, EAF: Effect allele
         frequency')
       ),
       class='display',
       container = sketchdmstatgene
      )
     }
    }
   }
})

# help window SNP-search:
observeEvent(input$question_SNP,{
 showModal(modalDialog(
  titel="Help SNP search",
  tags$h4("Variant search - Help:"),
  p("Enter a RSid or genomic position in format of
   chromosome:position (GRCh37) of a SNP you are interested in. If
   you search without any additional options the association
   statistics for unconditioned eGFRcrea (if the SNP has an
   association p-value <5E-8) are displayed. If the SNP is a 99%
   credible variant for any eGFRcrea signal also conditioned
   results (if there are other independent signals in the
   respective locus) and the posterior propabitliy of association
   (PPA) is shown in a second table. For further information check
   the Documentation & Help page."),
  easyClose = TRUE,
  footer=NULL,
  size = "m"
 ))
})

#Locus Zoom
Locus_ID_snp<-reactive({
 if(valid()){
  cred_var$`Locus Id`[which(cred_var$rsid==snp())]}
})

#create HTML syntax for link to plots
snp_plots <- reactive({
 if(valid()){
  ids=which(links_id%in%Locus_ID_snp())[1]
  paste("<p><a href=\"",links[ids],"\"target=\"_blank\">Regional
   association plot of locus",links_id[ids],"</a></p>")
 }
})

#render links
output$plot_download_links_snp <- renderUI({
 list(lapply(snp_plots(),HTML))
})
```

```
# gene page ----------------------------------------------------


#define input type based on click for further analyses
inputtype <- reactiveValues('gene'=c())

onclick("Genename", (inputtype$gene=c("single")), add=T)

onclick("gene_batch", (inputtype$gene=c("list")), add=T)

onclick("gene-upload-section", (inputtype$gene=c("upload")),
add=T)

observe({
 shinyjs::toggleElement(id="gene_batch_upload_options",
  condition = (isTruthy(input$file2)) )
})

#show search button:
observe(
 if(isTruthy(input$file2)|isTruthy(input$Genename)
    |isTruthy(input$gene_batch)){
  shinyjs::showElement(id="gene_go_div")
 }
)


#combine ENTER and "go" click to one reactive value only reacting if there is gene input
and change that value each time "go" or ENTER are clicked
go <- reactiveValues('gene'=c(c()))

observeEvent(input$gene_go, {
 if(isTruthy(input$file2)|isTruthy(input$Genename)
    |isTruthy(input$gene_batch)){
  if(is.null(go$gene)){go$gene=1}
  else{go$gene <- go$gene+1}
 }
})
observeEvent(input$keyPressed, {
 if(isTruthy(input$file2)|isTruthy(input$Genename)
    |isTruthy(input$gene_batch)){
  if(is.null(go$gene)){go$gene=1}
  else{go$gene <- go$gene+3}
 }
})


#first gene input processing:
single_gene <- eventReactive(go$gene, {input$Genename})

#build vector of genes from text input
input_gene_batch_list <- eventReactive(go$gene,{
 req(input$gene_batch)
 igbl=unlist(strsplit(input$gene_batch,
  "[^[:alnum:]^-]",perl=TRUE)) # excludes all non-word characteres except
                               "-" that might be used as separators
```

```
 if(any(igbl=="")){
  igbl[-which(igbl=="")]
 }else{igbl}
})

#build vector of genes from file input
input_gene_batch_upload <- reactiveValues('input'= c())

observe({
 req(input$file2)
 input_gene_batch_upload$input <- scan(file=input$file2$datapath,
                                       sep = input$sep_gene,
                                       what="character")
})

#reset file input, when reset button is clicked
observeEvent(input$reset_gene,{
 reset("file2")
 input_gene_batch_upload$input <- NULL
})

#final selection:
gene <- eventReactive(go$gene, {
 if(inputtype$gene=="single"){
  toupper(single_gene())
 }else{
  if(inputtype$gene=="list"){
   toupper(input_gene_batch_list())
  }else{
   if(inputtype$gene=="upload"){
    toupper(input_gene_batch_upload$input)
   }else{NULL}
  }
 }
})

inputtype <- reactiveValues('gene_final'=c())

observeEvent(go$gene,{
  inputtype$gene_final = inputtype$gene
})


corrected_genenames <- eventReactive(go$gene,{ # indices vector if input
                                               genenames are not HGNC names
 if(!is.null(gene())){
  if(any(gene()%in%toupper(real_genes$genes)
     &!gene()%in%toupper(real_genes$Approved.symbol) )){
   which(gene()%in%toupper(real_genes$genes)
       &!gene()%in%toupper(real_genes$Approved.symbol))
  }else{NULL}
 }else{NULL}
})


correct_genenames <- eventReactive(go$gene,{
```

```
 if(!is.null(gene())){
  gene1 = gene()
  if(!is.null(corrected_genenames())){
   for (i in corrected_genenames()){
   gene1[i]=toupper(real_genes$Approved.symbol[
    which(real_genes$genes==gene1[i])]) # correct the input gene names to
                                        HGNC gene names
   }
  }
  if(any(gene1%in%toupper(genes_sorted$Gene))){
   gene1[which(gene1%in%toupper(genes_sorted$Gene))]
   #check for overlap with gene-list that was used to generate KidneyGPS
  }else{NULL}
 }else{NULL}
})

#Render text for synonym-check:
output$testforme <-  renderText({
 if(!is.null(gene())){
  if(!is.null(corrected_genenames())){
   if(length(corrected_genenames())==1){
    paste("Of your ",length(gene()), " gene names(s), ",
     length(corrected_genenames()), " was a synonym and was
     matched to its  official HGNC gene name
     [",correct_genenames(),"].")
   }else{
    paste("Of your ",length(gene()), " gene names(s), ",
     length(corrected_genenames()), " were a synonyms and were
     matched to their  official HGNC gene names.")
   }
  }
 }
})

#check for overlap of correct gene-names with GPS-Genes:
genes_in_gps <- eventReactive(go$gene,{
 if(!is.null(correct_genenames())){
  if(any(correct_genenames()%in%toupper(GPS$Gene))){
   correct_genenames()[which(correct_genenames()
   %in%toupper(GPS$Gene))]
  }else{NULL}
 }else{NULL}
})

valid_gene <- eventReactive(go$gene, {!is.null(genes_in_gps())})

# combine mapping input options to one value. Contains the indices of GPS columns
referring to the selected options:
columns <- eventReactive(go$gene,{
 if(isTruthy(input$columns1)|isTruthy(input$columns2)
  |isTruthy(input$columns3)|isTruthy(input$columns4)){
   as.numeric(c(input$columns1,input$columns2,input$columns3,
   input$columns4))
 }else{NULL}
})
```

```
detail_evidence_locus_gene <- eventReactive(go$gene, {
 input$detail_evidence_locus_gene}) # save additional options in this value

ppa <- eventReactive(go$gene, {input$ppa}) # save ppa input

incorrect_genenames <- eventReactive(go$gene,{ # calculate number of
                                               incorrect gene-names
 length(gene())-length(correct_genenames())
})

# render text for gene search depending of how many input genes are real gene-names
and how many overlap KidneyGPS genes:
output$namecheck_gene <- renderText({
 if(inputtype$gene_final=="single"){
  if(is.null(correct_genenames())){
   paste(gene()," is no valid/official gene name, please check
    your input.",sep="")
  }else{
   if(is.null(genes_in_gps())){
    paste(correct_genenames()," is not included in any eGFRcrea
     locus.", sep="")
   }else{
    if(!is.null(corrected_genenames())){
     paste("Following information has been found for ",
      genes_in_gps(),"[",gene(),"]",":",sep="")
    }else{
     paste("Following information has been found for ",
      genes_in_gps(),":",sep="")
    }
   }
  }
 }else{
  if(is.null(correct_genenames)){
   if(incorrect_genenames()==1){
    paste("You queried ",length(gene())," gene name. ",gene()," 
     is no official/valid gene name. No further searches were
     performed. Please check your input.",sep="")
   }else{
    paste("You queried ",length(gene())," gene names. All of these
    are no official/valid gene names. No further searches were
    performed. Please check your input.",sep="")
   }
  }else{
   if(is.null(genes_in_gps())){
    if(incorrect_genenames()==1){
     paste("You queried ",length(gene())," genes. ",
      incorrect_genenames()," of these is no official/valid gene
      name. Additionally, none of your ",
      length(correct_genenames()), " queried official genes is not
      included in any of the 424 eGFRcrea loci. Thus, no GPS
      entries are available.",sep="")
    }else{
     if(incorrect_genenames()==0){
      paste("You queried ",length(gene())," genes. All of these
       are official/valid gene names. However, none of your ",
```

```
          length(correct_genenames()), " queried official genes is
          included in any of the 424 eGFRcrea loci. Thus, no GPS
          entries are available.",sep="")
        }else{
         paste("You queried ",length(gene()),," genes. ",
          incorrect_genenames()," of these are no official/valid gene
          names. Additionally, none of your ",
          length(correct_genenames()), " queried official genes is
          included in any of the 424 eGFRcrea loci. Thus, no GPS
          entries are available.",sep="")
        }
       }
      }else{
       if(length(correct_genenames())==1){
        if(incorrect_genenames()==1){
         paste("You queried ",length(gene()),," genes. ",
          incorrect_genenames()," of these is no official/valid gene
          names. ",length(genes_in_gps()), " (",genes_in_gps(),")","
          of your ", length(correct_genenames()), " queried official
          genes is included in any of the 424 eGFRcrea loci and thus
          included in the GPS.",sep="")
        }else{
         if(incorrect_genenames()==0){
          paste("You queried ",length(gene()),," genes. All of these
           are official/valid gene names from HGNC. ",
           length(genes_in_gps()),," (",genes_in_gps(),")"," of your
           ", length(correct_genenames()), " queried official genes
           is included in any of the 424 eGFRcrea loci and thus
           included in the GPS.",sep="")
         }else{
          paste("You queried ",length(gene()),," genes. ",
           incorrect_genenames()," of these are no official/valid
           gene names. ",length(genes_in_gps()),," (",genes_in_gps(),
           ")"," of your ", length(correct_genenames()), " queried
           official genes is included in any of the 424 eGFRcrea loci
           and thus included in the GPS.",sep="")
         }
        }
       }else{
        if(incorrect_genenames()==1){
         paste("You queried ",length(gene()),," genes. ",
          incorrect_genenames()," of these is no official/valid gene
          names. ",length(genes_in_gps()),," of your ",
          length(correct_genenames()), " queried official genes are
          included in any of the 424 eGFRcrea loci and thus included
          in the GPS.",sep="")
        }else{
         if(incorrect_genenames()==0){
          paste("You queried ",length(gene()),," genes. All of these
           are official/valid gene names from HGNC. ",
           length(genes_in_gps()),," of your ",
           length(correct_genenames()), " queried official genes are
           included in any of the 424 eGFRcrea loci and thus included
           in the GPS.",sep="")
         }else{
          paste("You queried ",length(gene()),," genes. ",
```

```
          incorrect_genenames()," of these are no official/valid
          gene names. ",length(genes_in_gps())," of your ",
          length(correct_genenames()), " queried official genes are
          included in any of the 424 eGFRcrea loci and thus included
          in the GPS.",sep="")
        }
       }
      }
     }
    }
   }
  })
```

render CADD output-table

```
output$CADD_gene <- DT::renderDT(server=FALSE, {
 if(valid_gene()){
  if(any(columns()%in%c(11:13))){
   if(any(toupper(CADD$GeneName)%in%genes_in_gps())){
    x = CADD_show[which(toupper(CADD$GeneName)
                  %in%genes_in_gps()&CADD$PPA>=ppa()),]
    if(any(duplicated(x))) x = x[-which(duplicated(x)),]
    datatable(x, rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
      dom = 'lfrtBip',
      buttons= list(c('copy', 'excel','csv'))
     ),
     extensions = 'Buttons',
     caption = htmltools::tags$caption(
      style = 'caption-side: bottom; text-align: left;
       font-weight: normal;',
      htmltools::tags$p('PPA: posterior probability of association
       of the SNP (probability of driving the association of the
       respective signal)'),
      htmltools::tags$p('Reference and alternative allele from
       human reference genome (GRCh37)')
     ),
     class ='display',
     container = sketchcadd
    )%>% formatPercentage('PPA')
   }
  }
 }
})
```

#store excerpt of CADD table overlapping searched genes:

```
cadd_gene <- reactive({
 if(valid_gene()){
  if(any(columns()%in%c(11:13))){
   if(any(toupper(CADD$GeneName)%in%genes_in_gps())){
```

```
     x = CADD[which(toupper(CADD$GeneName)
              %in%genes_in_gps()&CADD$PPA>=ppa()),]
     if(any(duplicated(x))){
      l = which(duplicated(x))
      x = x[-l,]
     }x
    }else{0}
   }else{0}
  }else{0}
})

#render cadd text:
output$CADD_gene_text <- renderText({
 if(inputtype$gene_final=="single"){
  if(is.data.frame(cadd_gene())){
   paste(genes_in_gps()," contains at least one credible variant
    that is predicted deleterious or has a functional consequence
    for ",genes_in_gps() ,":",sep="")
  }else{
   paste(genes_in_gps()," does not contain a credible variant that
    is predicted deleterious or has a functional consequence for
    ",genes_in_gps(),".",sep="")
  }
 }else{
  if(is.data.frame(cadd_gene())){
   if(length(genes_in_gps())==1){
    paste(genes_in_gps()," contains at least one credible variant
     that is predicted deleterious or has a functional consequence
     for ",genes_in_gps() ,":",sep="")
   }else{
    if(length(unique(cadd_gene()[,15])==1)){
     paste("Of your ",length(genes_in_gps()), " searched genes
      included in the GPS, ",length(unique(cadd_gene()[,15])),"
      gene contains at least one credible variant that is predicted
      deleterious or has a functional consequence for this
      gene:",sep="")
    }else{
     paste("Of your ",length(genes_in_gps()), " searched genes
      included in the GPS, ",length(unique(cadd_gene()[,15])),"
      genes contain at least one credible variant that is predicted
      deleterious or has a functional consequence for the
      respective gene:",sep="")
    }
   }
  }else{
   paste("None of your ",length(genes_in_gps()), " searched genes
    included in the GPS contains at least one credible variant
    that is predicted deleterious or has a functional consequence
    for the respective gene.",sep="")
  }
 }
})

#Neptune & Susztak glomerulus:
output$glo_gene <- DT::renderDT(server=FALSE, {
 if(valid_gene()){
```

```
    if(any(columns()==14)){
     if(any(toupper(glo_show$`Affected gene`)%in%genes_in_gps())){
      datatable(glo_show[
       which(toupper(glo_show$`Affected gene`)%in%genes_in_gps()
       & glo_show$PPA>=ppa()),], rownames=FALSE,
       options = list(
        columnDefs = list(list(className = 'dt-left',
         targets = "_all")),
        scrollX=TRUE, scrollCollapse = TRUE,
        dom = 'lfrtBip',
        buttons= list(c('copy', 'excel','csv'))
       ),
       extensions = 'Buttons',
       caption = htmltools::tags$caption(
        style = 'caption-side: bottom; text-align: left;
         font-weight: normal;',
        htmltools::tags$p('* eQTL effect allele is the eGFRcrea
         lowering allele'),
        htmltools::tags$p('PPA: posterior probability of association
         of the SNP (probability of driving the association of the
         respective signal), EAF: Effect allele frequency'),
        htmltools::tags$p('** NEPTUNE, or Sheng et al, Nat.Genet.
         2021')
       )
      )%>% formatPercentage('PPA')
     }
    }
   }
  })

  glo_gene <- reactive({
   if(valid_gene()){
    if(any(columns()==14)){
     if(any(toupper(glo$Approved.symbol)%in%genes_in_gps())){
      glo[which(toupper(glo$Approved.symbol)%in%genes_in_gps()
                & glo$ppa>=ppa()),]
     }else{0}
    }else{0}
   }else{0}
  })

  output$glo_gene_text <- renderText({
   if(inputtype$gene_final=="single"){
    if(is.data.frame(glo_gene())){
     paste("Expression of ",genes_in_gps()," is modulated by a
      credible variant in glomerular tissue:",sep="")
    }else{
     paste("No credible variant modulates expression of ",
      genes_in_gps()," in glomerular tissue.",sep="")
    }
   }else{
    if(is.data.frame(glo_gene())){
     if(length(genes_in_gps())==1){
      paste("Expression of ",genes_in_gps()," is modulated by a
       credible variant in glomerular tissue:",sep="")
     }else{
```

```
     if(length(unique(glo_gene()[,5]))==1){
      paste("Of your ",length(genes_in_gps()), " searched genes
       included in the GPS, "," expression of ",
       length(unique(glo_gene()[,5])),"gene is modulated by a
       credible variant in glomerular tissue:",sep="")
     }else{
      paste("Of your ",length(genes_in_gps()), " searched genes
       included in the GPS, "," expression of ",
       length(unique(glo_gene()[,5])),"genes is modulated by a
       credible variant in glomerular tissue:",sep="")
     }
    }
   }else{
    paste("Expression of none of your ",length(genes_in_gps()), "
     searched genes included in the GPS is modulated by a credible
     variant in glomerular tissue.",sep="")
   }
  }
 })
```

#Neptune & Susztak tubulo-interstitium：

```
output$tub_gene <- DT::renderDT(server=FALSE, {
 if(valid_gene()){
  if(any(columns()==15)){
   if(any(toupper(tub_show$`Affected gene`)%in%genes_in_gps())){
    datatable(tub_show[
     which(toupper(tub_show$`Affected gene`)
     %in%genes_in_gps()&tub_show$PPA>=ppa()),],rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
      dom = 'lfrtBip',
      buttons= list(c('copy', 'excel','csv'))
     ),
     extensions = 'Buttons',
     caption = htmltools::tags$caption(
      style = 'caption-side: bottom; text-align: left;
       font-weight: normal;',
      htmltools::tags$p('* eQTL effect allele is the eGFRcrea
       lowering allele'),
      htmltools::tags$p('PPA: posterior probability of association
       of the SNP (probability of driving the association of the
       respective signal), EAF: Effect allele frequency'),
      htmltools::tags$p('** NEPTUNE, or Sheng et al, Nat.Genet.
       2021')
     )
    )%>% formatPercentage('PPA')
   }
  }
 }
})

tub_gene <- reactive({
 if(valid_gene()){
  if(any(columns()==15)){
```

```
    if(any(toupper(tub$Approved.symbol)%in%genes_in_gps())){
     tub[which(toupper(tub$Approved.symbol)
      %in%genes_in_gps()&tub$ppa>=ppa()),]
    }else{0}
   }else{0}
  }else{0}
})

output$tub_gene_text <- renderText({
 if(inputtype$gene_final=="single"){
  if(is.data.frame(tub_gene())){
   paste("Expression of ",genes_in_gps()," is modulated by a
    credible variant in tubulo-interstitial tissue:",sep="")
  }else{
   paste("No credible variant modulates expression of ",
    genes_in_gps()," in tubulo-interstitial tissue.",sep="")
  }
 }else{
  if(is.data.frame(tub_gene())){
   if(length(genes_in_gps())==1){
    paste("Expression of ",genes_in_gps()," is modulated by a
     credible variant in tubulo-interstitial tissue:",sep="")
   }else{
    if(length(unique(tub_gene()[,'Approved.symbol']))==1){
     paste("Of your ",length(genes_in_gps()), " searched genes
      included in the GPS, "," expression of ",
      length(unique(tub_gene()[,'Approved.symbol']))," gene is
      modulated by a credible variant in tubulo-interstitial
      tissue:",sep="")
    }else{
     paste("Of your ",length(genes_in_gps()), " searched genes
      included in the GPS, "," expression of ",
      length(unique(tub_gene()[,'Approved.symbol']))," genes is
      modulated by a credible variant in tubulo-interstitial
      tissue:",sep="")
    }
   }
  }else{
   paste("Expression of none of your ",length(genes_in_gps()), "
    searched genes included in the GPS is modulated by a credible
    variant in tubulo-interstitial tissue.",sep="")
  }
 }
})

#GTEx eQTL kidney
output$eqtl_gene <- DT::renderDT(server=FALSE, {
 if(valid_gene()){
  if(any(columns()==16)){
   if(any(toupper(GTEx_eQTL$gene)%in%genes_in_gps())){
    datatable(GTEx_eQTL_show[which(toupper(GTEx_eQTL$gene)
     %in%genes_in_gps()&GTEx_eQTL$ppa>=ppa()),],rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
```

```
        dom = 'lfrtBip',
        buttons= list(c('copy', 'excel','csv'))
       ),
       extensions = 'Buttons',
       caption = htmltools::tags$caption(
        style = 'caption-side: bottom; text-align: left;
         font-weight: normal;',
        htmltools::tags$p('* eQTL effect allele is the eGFRcrea
         lowering allele'),
        htmltools::tags$p('PPA: posterior probability of
         association of the SNP (probability of driving the
         association of the respective signal), EAF: Effect allele
         frequency'),
       )
      )%>% formatPercentage('PPA')
     }
    }
   }
  })

  eqtl_gene <- reactive({
   if(valid_gene()){
    if(any(columns()==16)){
     if(any(toupper(GTEx_eQTL$gene)%in%genes_in_gps())){
      GTEx_eQTL[which(toupper(GTEx_eQTL$gene)
                %in%genes_in_gps()&GTEx_eQTL$ppa>=ppa()),]
     }else{0}
    }else{0}
   }else{0}
  })

  output$eqtl_gene_text <- renderText({
   if(inputtype$gene_final=="single"){
    if(is.data.frame(eqtl_gene())){
     paste("Expression of ",genes_in_gps()," is modulated by a
      credible variant in kidney cortex tissue:",sep="")
    }else{
     paste("No credible variant modulates expression of ",
      genes_in_gps()," in kidney cortex tissue.",sep="")
    }
   }else{
    if(is.data.frame(eqtl_gene())){
     if(length(genes_in_gps())==1){
      paste("Expression of ",genes_in_gps()," is modulated by a
       credible variant in kidney cortex tissue:",sep="")
     }else{
      if(length(unique(eqtl_gene()[,'gene']))==1){
       paste("Of your ",length(genes_in_gps()), " searched genes
        included in the GPS, "," expression of ",
        length(unique(eqtl_gene()[,'gene']))," gene is modulated by
        a credible variant in kidney cortex tissue:",sep="")
      }else{
       paste("Of your ",length(genes_in_gps()), " searched genes
        included in the GPS, "," expression of ",
        length(unique(eqtl_gene()[,'gene']))," genes is modulated
        by a credible variant in kidney cortex tissue:",sep="")
```

```
     }
    }
   }else{
    paste("Expression of none of your ",length(genes_in_gps()),
     " searched genes included in the GPS is modulated by a credible
     variant in kidney cortex tissue.",sep="")
   }
  }
})
```

#GTEx other tissue eQTL

```
output$eqtl_wo_kidney_gene <- DT::renderDT(server=FALSE, {
 if(valid_gene()){
  if(any(columns()==19)){
   if(any(toupper(gtex_without_kidney_show[,3])
    %in%genes_in_gps())){
    datatable(gtex_without_kidney_show[
     which(toupper(gtex_without_kidney_show[,3])
     %in%genes_in_gps()& gtex_without_kidney_show$PPA
     >= ppa()),c(1:13)],rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
      dom = 'lfrtBip',
      buttons= list(c('copy', 'excel','csv'))
     ),
     extensions='Buttons',
     caption = htmltools::tags$caption(
      style = 'caption-side: bottom; text-align: left;
       font-weight: normal;',
      htmltools::tags$p('* eQTL effect allele is the eGFRcrea
       lowering allele'),
      htmltools::tags$p('PPA: posterior probability of
       association of the SNP (probability of driving the
       association of the respective signal), EAF: Effect allele
       frequency'),
     )
    )%>% formatPercentage('PPA')
   }
  }
 }
})

eqtl_wo_kidney_gene <- reactive({
 if(valid_gene()){
  if(any(columns()==19)){
   if(any(toupper(gtex_without_kidney_show[,3])
                  %in%genes_in_gps())){
    gtex_without_kidney_show[
     which(toupper(gtex_without_kidney_show[,3])
     %in%genes_in_gps()& gtex_without_kidney_show$PPA >= ppa()),]
   }else{0}
  }else{0}
 }else{0}
```

```
})

output$eqtl_wo_kidney_gene_text <- renderText({
 if(inputtype$gene_final=="single"){
  if(is.data.frame(eqtl_wo_kidney_gene())){
   paste("Expression of ",genes_in_gps()," is modulated by a
    credible variant in other tissues:",sep="")
  }else{
   paste("No credible variant modulates expression of ",
    genes_in_gps()," in other tissues.",sep="")
  }
 }else{
  if(is.data.frame(eqtl_wo_kidney_gene())){
   if(length(genes_in_gps())==1){
    paste("Expression of ",genes_in_gps()," is modulated by a
     credible variant in other tissues:",sep="")
   }else{
    if(length(unique(eqtl_wo_kidney_gene()[,3]))==1){
     paste("Of your ",length(genes_in_gps()), " searched genes
      included in the GPS, "," expression of ",
      length(unique(eqtl_wo_kidney_gene()[,3])),"gene is
      modulated by a credible variant in other tissues:",sep="")
    }else{
     paste("Of your ",length(genes_in_gps()), " searched genes
      included in the GPS, "," expression of ",
      length(unique(eqtl_wo_kidney_gene()[,3])),"genes is
      modulated by a credible variant in other tissues:",sep="")
    }
   }
  }else{
   paste("Expression of none of your ",length(genes_in_gps()), "
    searched genes included in the GPS is modulated by a credible
    variant in other tissues.",sep="")
  }
 }
})


# GTEx sQTL kidney
output$sqtl_gene <- DT::renderDT(server=FALSE, {
 if(valid_gene()){
  if(any(columns()==18)){
   if(any(toupper(GTEx_sQTL$gene)%in%genes_in_gps())){
    datatable(GTEx_sQTL_show[
     which(toupper(GTEx_sQTL$gene)%in%genes_in_gps()
     &GTEx_sQTL$ppa >= ppa()),],rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
      dom = 'lfrtBip',
      buttons= list(c('copy', 'excel','csv'))
     ),
     extensions='Buttons',
     caption = htmltools::tags$caption(
      style = 'caption-side: bottom; text-align: left;
```

```
          font-weight: normal;',
        htmltools::tags$p('* sQTL effect allele is the eGFRcrea
          lowering allele'),
        htmltools::tags$p('PPA: posterior probability of
          association of the SNP (probability of driving the
          association of the respective signal), EAF: Effect allele
          frequency'),
       )
      )%>% formatPercentage('PPA')
     }
    }
   }
  })

  sqtl_gene <- reactive({
   if(valid_gene()){
    if(any(columns()==18)){
     if(any(toupper(GTEx_sQTL$gene)%in%genes_in_gps())){
      GTEx_sQTL[which(toupper(GTEx_sQTL$gene)%in%genes_in_gps()
      &GTEx_sQTL$ppa >= ppa()),]
     }else{0}
    }else{0}
   }else{0}
  })

  output$sqtl_gene_text <- renderText({
   if(inputtype$gene_final=="single"){
    if(is.data.frame(sqtl_gene())){
     paste("Splicing of ",genes_in_gps()," is modulated by a
      credible variant in kidney cortex tissue:",sep="")
    }else{
     paste("No credible variant modulates splicing of ",
      genes_in_gps()," in kidney cortex tissue.",sep="")
    }
   }else{
    if(is.data.frame(sqtl_gene())){
     if(length(genes_in_gps())==1){
      paste("Splicing of ",genes_in_gps()," is modulated by a
       credible variant in kidney cortex tissue:",sep="")
     }else{
      if(length(unique(sqtl_gene()[,'gene']))==1){
       paste("Of your ",length(genes_in_gps()), " searched genes
        included in the GPS, "," splicing of ",
        length(unique(sqtl_gene()[,'gene']))," gene is modulated by
        a credible variant in kidney cortex tissue:",sep="")
      }else{
       paste("Of your ",length(genes_in_gps()), " searched genes
        included in the GPS, "," splicing of ",
        length(unique(sqtl_gene()[,'gene']))," genes is modulated
        by a credible variant in kidney cortex tissue:",sep="")
      }
     }
    }else{
     paste("Splicing of none of your ",length(genes_in_gps()), "
      searched genes included in the GPS is modulated by a credible
      variant in kidney cortex tissue.",sep="")
```

```
   }
  }
 })

 # GTEx sQTL other tissues
 output$sqtl_wo_kidney_gene <- DT::renderDT(server=FALSE, {
  if(valid_gene()){
   if(any(columns()==19)){
    if(any(toupper(sqtl_without_kidney_show[,3])
                   %in%genes_in_gps())){
     datatable(sqtl_without_kidney_show[
     which(toupper(sqtl_without_kidney_show[,3])
     %in%genes_in_gps()
     &sqtl_without_kidney_show$PPA >= ppa()),c(1:13)],
      rownames=FALSE,
      options = list(
       columnDefs = list(list(className = 'dt-left',
        targets = "_all")),
       scrollX=TRUE, scrollCollapse = TRUE,
       dom = 'lfrtBip',
       buttons= list(c('copy', 'excel','csv'))
      ),
      extensions='Buttons',
      caption = htmltools::tags$caption(
       style = 'caption-side: bottom; text-align: left;
        font-weight: normal;',
       htmltools::tags$p('* sQTL effect allele is the eGFRcrea
        lowering allele'),
       htmltools::tags$p('PPA: posterior probability of
        association of the SNP (probability of driving the
        association of the respective signal), EAF: Effect allele
        frequency'),
      )
     )%>% formatPercentage('PPA')
    }
   }
   }
 })

 sqtl_wo_kidney_gene <- reactive({
  if(valid_gene()){
   if(any(columns()==19)){
    if(any(toupper(sqtl_without_kidney_show[,3])
                   %in%genes_in_gps())){
     sqtl_without_kidney_show[
     which(toupper(sqtl_without_kidney_show[,3])
     %in%genes_in_gps()& sqtl_without_kidney_show$PPA >= ppa()),]
    }else{0}
   }else{0}
  }else{0}
 })

 output$sqtl_wo_kidney_gene_text <- renderText({
  if(inputtype$gene_final=="single"){
   if(is.data.frame(sqtl_wo_kidney_gene())){
    paste("Splicing of ",genes_in_gps()," is modulated by a
```

```
      credible variant in other tissues:",sep="")
    }else{
     paste("No credible variant modulates splicing of ",
      genes_in_gps()," in other tissues.",sep="")
    }
   }else{
    if(is.data.frame(sqtl_wo_kidney_gene())){
     if(length(genes_in_gps())==1){
      paste("Splicing of ",genes_in_gps()," is modulated by a
       credible variant in other tissues:",sep="")
     }else{
      if(length(unique(sqtl_wo_kidney_gene()[,3]))==1){
       paste("Of your ",length(genes_in_gps()), " searched genes
        included in the GPS, "," splicing of ",
        length(unique(sqtl_wo_kidney_gene()[,3])),"gene is
        modulated by a credible variant in other tissues:",sep="")
      }else{
       paste("Of your ",length(genes_in_gps()), " searched genes
        included in the GPS, "," splicing of ",
        length(unique(sqtl_wo_kidney_gene()[,3])),"genes is
        modulated by a credible variant in other tissues:",sep="")
      }
     }
    }else{
     paste("Splicing of none of your ",length(genes_in_gps()), "
      searched genes included in the GPS is modulated by a credible
      variant in other tissues.",sep="")
    }
   }
  })

  #MGI
  output$mgi_gene <- DT::renderDT(server=FALSE, {
   if(valid_gene()){
    if(any(columns()==20)){
     if(any(toupper(MGI$human_symbol)%in%genes_in_gps())){
      datatable(MGI_show[which(toupper(MGI$human_symbol)
       %in%genes_in_gps()),],rownames=FALSE,
       options = list(
        columnDefs = list(list(className = 'dt-left',
         targets = "_all")),
        scrollX=TRUE, scrollCollapse = TRUE,
        dom = 'lfrtBip',
        buttons= list(c('copy', 'excel','csv'))
       ),
       extensions = 'Buttons'
      )
     }
    }
   }
  })

  mgi_gene<-reactive({
   if(valid_gene()){
    if(any(columns()==20)){
     if(any(toupper(MGI$human_symbol)%in%genes_in_gps())){
```

```
     (MGI_show[which(toupper(MGI$human_symbol)
                     %in%genes_in_gps()),]
     )
    }else{0}
   }else{0}
  }else{0}
 })

 output$mgi_gene_text <- renderText({
  if(inputtype$gene_final=="single"){
   if(is.data.frame(mgi_gene())){
    paste("Following kidney phenotypes in mice are described
     for ",genes_in_gps()," :",sep="")
   }else{
    paste("No kidney phenotypes in mice were described for ",
     genes_in_gps()," .",sep="")
   }
  }else{
   if(is.data.frame(mgi_gene())){
    if(length(genes_in_gps())==1){
     paste("Following kidney phenotypes in mice are described for
      ",genes_in_gps()," :",sep="")
    }else{
     if(length(unique(mgi_gene()[,1]))==1){
      paste("Of your ", length(genes_in_gps())," searched genes
       included in the GPS, ",length(unique(mgi_gene()[,1])),"
       gene is described with kidney phenotypes in mice:",sep="")
     }else{
      paste("Of your ", length(genes_in_gps())," searched genes
       included in the GPS, ",length(unique(mgi_gene()[,1])),"
       genes are described with kidney phenotypes in
       mice:",sep="")
     }
    }
   }else{
    paste("No kidney phenotypes in mice were described for any of
     your ",length(genes_in_gps()), " searched genes included in
     the GPS.",sep="")
   }
  }
 })

 #OMIM, Groopman, Wopperer
 output$omim_gene <- DT::renderDT(server=FALSE, {
  if(valid_gene()){
   if(any(columns()==21)){
    if(any(toupper(OMIM$Gene)%in%genes_in_gps())){
     datatable(OMIM_MIM[which(toupper(OMIM$Gene)
      %in%genes_in_gps()),c(1,4,2,3)],rownames=FALSE,
      options = list(
       columnDefs = list(list(className = 'dt-left',
        targets = "_all")),
       scrollX=TRUE, scrollCollapse = TRUE,
       dom = 'lfrtBip',
       buttons= list(c('copy', 'excel','csv'))
      ),
```

```
      extensions = 'Buttons',
      escape = FALSE,
      caption = htmltools::tags$caption(
       style = 'caption-side: bottom; text-align: left;
        font-weight: normal;',
       htmltools::tags$p('* Link to OMIM entry'),
       htmltools::tags$p('** Online Mendelian Inheritance in Man,
        OMIM; Groopman et al, N.Engl.J.Med. 2019 or Wopperer et
        al, Kidney Int. 2022')
      )
     )
    }
   }
  }
 })


 omim_gene <- reactive({
  if(valid_gene()){
   if(any(columns()==21)){
    if(any(toupper(OMIM$Gene)%in%genes_in_gps())){
     OMIM_MIM[which(toupper(OMIM$Gene)
                    %in%genes_in_gps()),c(1,4,2,3)]
    }else{0}
   }else{0}
  }else{0}
 })

 output$omim_gene_text <- renderText({
  if(inputtype$gene_final=="single"){
   if(is.data.frame(omim_gene())){
    paste("Following genetic disorders with kidney phenotypes in
     human are described for ",genes_in_gps()," :",sep="")
   }else{
    paste("No genetic disorders with kidney phenotypes in human
     were described for ",genes_in_gps()," .",sep="")
   }
  }else{
   if(is.data.frame(omim_gene())){
    if(length(genes_in_gps())==1){
     paste("Following genetic disorders with kidney phenotypes in
      human are described for ",genes_in_gps()," :",sep="")
    }else{
     if(length(unique(omim_gene()[,1]))==1){
      paste("Of your ", length(genes_in_gps())," searched genes
       included in the GPS, ",length(unique(omim_gene()[,1])),"
       gene is described with genetic disorders with kidney
       phenotypes in human:",sep="")
     }else{
      paste("Of your ", length(genes_in_gps())," searched genes
       included in the GPS, ",length(unique(omim_gene()[,1])),"
       genes are described with genetic disorders with kidney
       phenotypes in human:",sep="")
     }
    }
   }else{
```

```
    paste("No genetic disorders with kidney phenotypes in human
     were described for any of your ",length(genes_in_gps()), "
     searched genes included in the GPS.",sep="")
   }
  }
 })

 #drugability
 #kidney drugs
 output$drug_gene_kid <- DT::renderDT(server=FALSE, {
  if(valid_gene()){
   if(any(columns()==25)){
    if(any(toupper(drugability$gps_gene[
    which(drugability$includes_kideny_disease)])
    %in%genes_in_gps())){
     datatable(drugability_show[
      which(toupper(drugability$gps_gene)%in%genes_in_gps()
      &drugability$includes_kideny_disease),],rownames=FALSE,
      options = list(
       columnDefs = list(list(className = 'dt-left',
        targets = "_all")),
       scrollX=TRUE, scrollCollapse = TRUE,
       dom = 'lfrtBip',
       buttons= list(c('copy', 'excel','csv'))
      ),
      extensions = 'Buttons',
      escape = FALSE,
      caption = htmltools::tags$caption(
       style = 'caption-side: bottom; text-align: left;
        font-weight: normal;',
       htmltools::tags$p('* Link to Therapeutic Target Database
        (TTD)'),
       htmltools::tags$p('** This status is the highest status for
        any target of this drug. This does not necessarily need
        to be the queried gene. Check TTD for more information')
      )
     )
    }
   }
  }
 })



 drug_gene_kid <- reactive({
  if(valid_gene()){
   if(any(columns()==25)){
    if(any(toupper(drugability$gps_gene[
    which(drugability$includes_kideny_disease)])
    %in%genes_in_gps())){
     drugability_show[which(toupper(drugability$gps_gene)
      %in%genes_in_gps() & drugability$includes_kideny_disease),]
    }else{0}
   }else{0}
  }else{0}
 })
```

```
output$drug_gene_kid_text <- renderText({
 if(inputtype$gene_final=="single"){
  if(is.data.frame(drug_gene_kid())){
   paste("Following drug information with indication for kidney
    diseases is described for ",genes_in_gps()," :",sep="")
  }else{
   paste(genes_in_gps()," is not described as known drug target
    or to show a drug interaction for drugs with indication for
    kidney
    diseases.",sep="")
  }
 }else{
  if(is.data.frame(drug_gene_kid())){
   if(length(genes_in_gps())==1){
    paste("Following drug information with indication for kidney
     diseases is described for ",genes_in_gps()," :",sep="")
   }else{
    if(length(unique(drug_gene_kid()[,1]))==1){
     paste("Of your ", length(genes_in_gps())," searched genes
      included in the GPS, ",
      length(unique(drug_gene_kid()[,1])),"
      gene is described as a known drug target or to show a drug
      interaction for drugs with indication for kidney
      diseases:",sep="")
    }else{
     paste("Of your ", length(genes_in_gps())," searched genes
      included in the GPS, ",
      length(unique(drug_gene_kid()[,1])),"genes are described
      as a known drug target or to show a drug
      interaction for drugs with indication for kidney
      diseases:",sep="")
    }
   }
  }else{
   paste("None of your ",length(genes_in_gps()), " searched genes
    included in the GPS is described as a known drug target or
    to show a drug interaction for drugs with indication for
    kidney diseases.",sep="")
  }
 }
})

#other drugs

output$drug_gene_oth <- DT::renderDT(server=FALSE, {
 if(valid_gene()){
  if(any(columns()==26)){
   if(any(toupper(drugability$gps_gene[
   which(!drugability$includes_kideny_disease)])
   %in%genes_in_gps())){
    datatable(drugability_show[
     which(toupper(drugability$gps_gene)%in%genes_in_gps()
     &!drugability$includes_kideny_disease),],rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
```

```
          targets = "_all")),
        scrollX=TRUE, scrollCollapse = TRUE,
        dom = 'lfrtBip',
        buttons= list(c('copy', 'excel','csv'))
       ),
       extensions = 'Buttons',
       escape = FALSE,
       caption = htmltools::tags$caption(
        style = 'caption-side: bottom; text-align: left;
         font-weight: normal;',
        htmltools::tags$p('* Link to Therapeutic Target Database
         (TTD)'),
        htmltools::tags$p('** This status is the highest status for
         any target of this drug. This does not necessarily need
         to be the queried gene. Check TTD for more information')
       )
      )
     }
    }
   }
  })



  drug_gene_oth <- reactive({
   if(valid_gene()){
    if(any(columns()==26)){
     if(any(toupper(drugability$gps_gene[
     which(!drugability$includes_kideny_disease)])
     %in%genes_in_gps())){
      drugability_show[which(toupper(drugability$gps_gene)
      %in%genes_in_gps() & !drugability$includes_kideny_disease),]
     }else{0}
    }else{0}
   }else{0}
  })

  output$drug_gene_oth_text <- renderText({
   if(inputtype$gene_final=="single"){
    if(is.data.frame(drug_gene_oth())){
     paste("Following drug information with indication for other
      diseases or missing indication is described for ",
      genes_in_gps()," :",sep="")
    }else{
     paste(genes_in_gps()," is not described as known drug target
      or to show a drug interaction for drugs with missing
      indication or indication for other diseases.",sep="")
    }
   }else{
    if(is.data.frame(drug_gene_oth())){
     if(length(genes_in_gps())==1){
      paste("Following drug information with indication for other
       diseases or missing indication is described for ",
       genes_in_gps()," :",sep="")
     }else{
      if(length(unique(drug_gene_oth()[,1]))==1){
```

```
       paste("Of your ", length(genes_in_gps()),"  searched genes
        included in the GPS, ",
        length(unique(drug_gene_oth()[,1])),"
        gene is described as a known drug target or to show a drug
        interaction for drugs with missing indication or indication
        for other diseases:",sep="")
      }else{
       paste("Of your ", length(genes_in_gps()),"  searched genes
        included in the GPS, ",
        length(unique(drug_gene_oth()[,1])),"
        genes are described as a known drug target or to show a
        drug interaction for drugs with missing indication or
        indication for other diseases:",sep="")
      }
     }
    }else{
     paste("None of your ",length(genes_in_gps()), " searched genes
      included in the GPS is described as a known drug target or
      to show a drug interaction for drugs with missing indication
      or indication for other diseases.",sep="")
    }
   }
  })


  #credible variants
  output$cred_var_gene <- DT::renderDT(server=FALSE, {
   if(valid_gene()){
    if(any(detail_evidence_locus_gene()=="cred_var")){
     # extract Locus IDs the searched genes belong to
     a=unique(GPS$locus_id[which(toupper(GPS$Gene)
                           %in%genes_in_gps())])
     # reduce cred_var_show table to variants belonging to those loci
     z=cred_var_show[which(cred_var$`Locus Id`%in%a
                     &cred_var$ppa >= ppa()),]
     # make vector with rsids
     snps=cred_var$rsid[which(cred_var$`Locus Id`
                       %in%a &cred_var$ppa >= ppa())]
     #add functional consequence if there is any:
     eqtl=rep("-",length(snps))
     if(is.data.frame(tub_gene())){
      eqtl[which(snps%in%tub_gene()[,1])]="eQTL"}
     if(is.data.frame(glo_gene())){
      eqtl[which(snps%in%glo_gene()[,1])]="eQTL"}
     if(is.data.frame(eqtl_gene())){
      eqtl[which(snps%in%eqtl_gene()[,1])]="eQTL"}
     sqtl=rep("-",length(snps))
     if(is.data.frame(sqtl_gene())){
      sqtl[which(snps%in%sqtl_gene()[,1])]="sQTL"}
     cadd=rep("-",length(snps))
     if(is.data.frame(cadd_gene())){
      cadd[which(snps%in%cadd_gene()[,1])]="protein-relevant SNP"}
     z['functional.consequence'] =
      ifelse(eqtl=="-",
       ifelse(sqtl=="-",
        ifelse(cadd=="-", "-", cadd),
```

```
        ifelse(cadd=="-", sqtl, paste(sqtl,cadd,sep=", "))),
       ifelse(sqtl=="-",
        ifelse(cadd=="-", eqtl, cadd),
        ifelse(cadd=="-", paste(eqtl,sqtlsep=", "),
       paste(eqtl,sqtl,cadd, sep=", "))))
    z=z[,c(1:4,17,5:16)]
    #render the final table:
    datatable(z, rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
      dom = 'lfrtBip',
      buttons= list(c('copy', 'excel','csv'))
     ),
     extensions = 'Buttons',
     escape=FALSE,
     caption = htmltools::tags$caption(
      style = 'caption-side: bottom; text-align: left;
       font-weight: normal;',
      htmltools::tags$p('* Link to dbSNP'),
      htmltools::tags$p('PPA: posterior probability of
       association of the SNP (probability of driving the
       association of the respective signal)'),
      htmltools::tags$p('N: Sample size of European-ancestry GWAS
       meta-analysis for this SNP, Effect allele: eGFRcrea
       lowering allele, EAF: Effect allele frequency'),
      htmltools::tags$p('For loci with multiple independent
       signals, the association results are provided conditioned
       and unconditioned on the signal index variants of the other
       signals in the locus.')
     ),
     class='display',
     container = sketchcredvargene
    )%>% formatPercentage('PPA')
   }
  }
 })

 #dm statistics
 output$dm_stat_gene <- DT::renderDT(server=FALSE, {
  if(valid_gene()){
   if(any(detail_evidence_locus_gene()=="dm_stat")){
    a=unique(GPS$locus_id[which(toupper(GPS$Gene)
                          %in%genes_in_gps())])
    snps=cred_var$rsid[which(cred_var$`Locus Id`
                       %in%a &cred_var$ppa >= ppa())]
    dm_gene = DM_stats_show[which(DM_stats$rsid%in%snps),]
    datatable(dm_gene, rownames=FALSE,
     options = list(
      columnDefs = list(list(className = 'dt-left',
       targets = "_all")),
      scrollX=TRUE, scrollCollapse = TRUE,
      dom = 'lfrtBip',
      buttons= list(c('copy', 'excel','csv'))
     ),
```

```
      extensions = 'Buttons',
      escape=FALSE,
      caption = htmltools::tags$caption(
       style = 'caption-side: bottom; text-align: left;
        font-weight: normal;',
       htmltools::tags$p('N: Sample size of European-ancestry GWAS
        meta-analysis for this SNP, Effect allele: eGFRcrea
        lowering allele in the combined analysis, EAF: Effect
        allele frequency')
      ),
      class='display',
      container = sketchdmstatgene
     )
    }
   }
  })
```

#generate GPS table from all the above excerpts of functional tables (eQTL, sQTL, CADD)

```
GPS_gene <- reactive({
 if(valid_gene()){
  x=GPS_show[which(toupper(GPS$Gene)%in%genes_in_gps()),]
  y=GPS[which(toupper(GPS$Gene)%in%genes_in_gps()),]
  rows=nrow(x)
  cadd1=rep(0,rows)
  cadd2=rep(0,rows)
  cadd3=rep(0,rows)
  tub_g=rep(0,rows)
  glo_g=rep(0,rows)
  eqtl_kid=rep(0,rows)
  sqtl_kid=rep(0,rows)
  eqtl_all=rep(0,rows)
  sqtl_all=rep(0,rows)
  for(i in 1:rows){
   if(is.data.frame(cadd_gene())){
    cadd1[i]=nrow(cadd_gene()[
     which(cadd_gene()['signal_id']==y[i,'signal_id']
     & cadd_gene()['ConsScore']>6
     & cadd_gene()['GeneName']==y[i,'Gene']
     & cadd_gene()['region_id']==y[i,'locus_id']),])
    cadd2[i]=nrow(cadd_gene()[
     which(cadd_gene()['signal_id']==y[i,'signal_id']
     & cadd_gene()['ConsScore']>4& cadd_gene()['ConsScore']<=6
     & cadd_gene()['GeneName']==y[i,'Gene']
     & cadd_gene()['region_id']==y[i,'locus_id']),])
    cadd3[i]=nrow(cadd_gene()[
     which(cadd_gene()['signal_id']==y[i,'signal_id']
     & cadd_gene()['ConsScore']<=4
     & cadd_gene()['GeneName']==y[i,'Gene']
     & cadd_gene()['region_id']==y[i,'locus_id']),])
   }
   if(is.data.frame(tub_gene())){
    tub_g[i]=length(unique(tub_gene()[
     which(tub_gene()['signal_id']==y[i,'signal_id']
     & tub_gene()['Approved.symbol']==y[i,'Gene']
```

```
      & tub_gene()['region_id']==y[i,'locus_id']),1]))
    }
    if(is.data.frame(glo_gene())){
     glo_g[i]=length(unique(glo_gene()[
      which(glo_gene()['signal_id']==y[i,'signal_id']&
      glo_gene()['Approved.symbol']==y[i,'Gene']
      & glo_gene()['region_id']==y[i,'locus_id']),1]))
    }
    if(is.data.frame(eqtl_gene())){
     eqtl_kid[i]=nrow(eqtl_gene()[
      which(eqtl_gene()['signal_id']==y[i,'signal_id']
      & eqtl_gene()['gene']==y[i,'Gene']
      & eqtl_gene()['region_id']==y[i,'locus_id']),])
    }
    if(is.data.frame(eqtl_wo_kidney_gene())){
     eqtl_all[i]=length(unique(eqtl_wo_kidney_gene()[
      which(eqtl_wo_kidney_gene()['signal_id']==y[i,'signal_id']
      & eqtl_wo_kidney_gene()['Affected gene']==y[i,'Gene']
      & eqtl_wo_kidney_gene()['region_id']==y[i,'locus_id']),1]))
    }
    if(is.data.frame(sqtl_gene())){
     sqtl_kid[i]=nrow(sqtl_gene()[
      which(sqtl_gene()['signal_id']==y[i,'signal_id']
      & sqtl_gene()['gene']==y[i,'Gene']
      & sqtl_gene()['region_id']==y[i,'locus_id']),])
    }
    if(is.data.frame(sqtl_wo_kidney_gene())){
     sqtl_all[i]=length(unique(sqtl_wo_kidney_gene()[
      which(sqtl_wo_kidney_gene()['signal_id']==y[i,'signal_id']
      & sqtl_wo_kidney_gene()['Affected gene']==y[i,'Gene']
      & sqtl_wo_kidney_gene()['region_id']==y[i,'locus_id']),1]))
    }
   }
   x[,'stop-gained, stop-lost, non-synonymus']=cadd1
   x[,'canonical splice, noncoding change, synonymous, splice
    site']=cadd2
   x[,'other deleterious variant']=cadd3
   x[,'eQTL glomerulus (NEPTUNE, or Sheng et al [Nat Genet,
    2021])']=glo_g
   x[,'eQTL tubulo-interstitium (NEPTUNE, or Sheng et al [Nat
    Genet, 2021])']=tub_g
   x[,'eQTL kidney cortex (GTEx)']=eqtl_kid
   x[,'eQTL other tissue (GTEx)']=eqtl_all
   x[,'sQTL kidney cortex (GTEx)']=sqtl_kid
   x[,'sQTL other tissue (GTEx)']=sqtl_all
   x=x[,c('Gene*','Locus name**','Locus ID','Signal ID','eGFRcys
    or BUN validation','DM_effect','decline_effect','# credible
    variants in signal','max PPA', names(GPS_show)[columns()],
    'Distance to locus lead variant','Chromosome','Position gene
     start','Position gene end')]
  }else{0}
 })


 # count how long the GPS table is:
 rowsGPS <- reactive({
```

```
 if(valid_gene()){nrow(GPS_gene())}
 else{0}
})

#header GPS
sketch_gene <- reactive({
 htmltools::withTags((table(
  class='display',
  thead(
   tr(
    th(rowspan = 2, class="Geneofficial hover", 'Gene*',
     span(class="Genecomment", "Official gene name from HGNC")),
    th(rowspan = 2, 'Locus name**'),
    th(rowspan = 2, class="Locusidwithcomment hover",'Locus ID',
     span(class="Locusidcomment", "k: known locus from Wuttke et
     al. Nat.commun.2019, n: novel locus in Stanzick et al.
     Nat.commun. 2021")),
    th(rowspan =2,class="Signalidwithcomment hover",'Signal ID',
     span(class="Signalidcomment", "ID of independent signals
     within a locus")),
    th(rowspan = 2, class="validationwithcomment hover",'eGFRcys
     or BUN validation', span(class="validationcomment",list(
     "Information whether the locus lead variant is nominal
     significant (P<0.05) associated with concordent effect
     direction with ",tags$abbr(title="glomerular filtration
     rate estimated from serum cystatin C","eGFRcys"),"or ",
     tags$abbr(title="blood urea nitrogen","BUN")))),
    th(rowspan = 2, class="DMwithcomment hover",'Signal
     association depends on DM', span(class="DMcomment",list(
     "Information whether the signal index variant (or a
     correlated variant) shows significant interaction with ",
     tags$abbr(title="Diabetes mellitus","DM"),"-status, Winkler
     et al. 2022"))),
    th(rowspan = 2, class="declinewithcomment hover",'Signal
     association with eGFRcrea decline',
     span(class="declinecomment",list( "Information whether the
     signal index variant (or a correlated variant) was
     established with annual eGFR-decline, Gorski et al.
     2022"))),
    th(rowspan = 2, '# credible variants in signal'),
    th(rowspan = 2, class="maxPPA hover",'max PPA',
     span(class="maxPPAcomment", "max. probability of a credible
     variant in this signal to be causal")),
    if(any(columns()%in%c(11:13))){
     th(colspan = length(which(columns()%in%c(11:13))),
      class="CADD",span('# protein-relevant credible variants in
      the gene'))},
    if(any(columns()%in%c(14:19))){
     th(colspan = length(which(columns()%in%c(14:19))),
      class="eqtl", '# credible variants that modulate gene
      expression (eQTLs) or splicing (sQTLs), FDR < 5%')},
    if(any(columns()==20)){
     th(rowspan = 2, class="eqtl", 'Coloc in NEPTUNE tissue')},
    if(any(columns()==21)){
     th(rowspan = 2, class="pheno",'# kidney phenotypes in
      mouse')},
```

```
      if(any(columns()==22)) {
       th(rowspan = 2, class="pheno", '# kidney phenotypes in
        human')},
      if(any(columns()%in%c(25,26))){
       th(rowspan = 2, class="drug", 'Gene is known as drug target
        or for drug interaction')},
      th(rowspan = 2, 'Distance to locus lead variant'),
      th(rowspan = 2, 'Chromosome'),
      th(rowspan = 2, 'Position gene start'),
      th(rowspan = 2, 'Position gene end')
     ),
     tr(
      if(any(columns()==11)){
       th(class="CADD",'stop-gained, stop-lost, non-synonymous')},
      if(any(columns()==12)) {
       th(class="CADD",'canonical splice, noncoding change,
        synonymous, splice site')},
      if(any(columns()==13)) {
       th(class="CADD", span('other deleterious variant (CADD-
        Phred',HTML('<span>&#8805;</span>'),'15)'))},
      if(any(columns()==14)) {
       th(class='eQTLglowithcomment hover eqtl','eQTL glomerulus',
        span(class='eQTLglocomment','NEPTUNE, or Sheng et al.
        [Nat.Genet. 2021]'))},
      if(any(columns()==15)) {
       th(class='eQTLtubwithcomment hover eqtl','eQTL tubulo-
        interstitium', span(class='eQTLtubcomment','NEPTUNE, or
        Sheng et al. [Nat.Genet. 2021]'))},
      if(any(columns()==16)) {
       th(class="eqtl",'eQTL kidney cortex (GTEx)')},
      if(any(columns()==17)) {
       th(class="eqtl",'eQTL other tissue (GTEx)')},
      if(any(columns()==18)) {
       th(class="eqtl",'sQTL kidney cortex (GTEx)')},
      if(any(columns()==19)) {
       th(class="eqtl",'sQTL other tissue (GTEx)')}
     )
    )
   )))
  })


  # render GPS table:
  output$GPSrows <-  DT::renderDT( server =FALSE, {
   datatable(GPS_gene(), rownames=FALSE,
    options = list(
     columnDefs = list(list(className = 'dt-left',
      targets = "_all")),
     scrollX=TRUE, scrollCollapse = TRUE,
     lengthChange=TRUE,
     dom = 'lfrtBip',
     buttons= list(c('copy', 'excel','csv'))
    ),
    extensions = 'Buttons',
    escape=FALSE,
    caption = htmltools::tags$caption(
```

```
    style = 'caption-side: bottom; text-align: left;
     font-weight: normal;',
    '* Link to GeneCards, **nearest gene to locus lead-variant'),
   container=sketch_gene()
 )%>% formatPercentage('max PPA')
})

#Locus summary
output$summary <- renderDataTable({
 if(valid_gene()){
  if(any(detail_evidence_locus_gene()=="summary")){
   a=unique(GPS$locus_id[
    which(toupper(GPS$Gene)%in%genes_in_gps())])
   b=region_table_show[which(region_table$Locus_id%in%a),]
   datatable(b[order(b[,1],b[,2]),], rownames=F,
    options = list(
     columnDefs = list(list(className = 'dt-left',
      targets = "_all")),
     scrollX=TRUE, scrollCollapse = TRUE,
     dom = 'lfrtBip',
     buttons= list(c('copy', 'excel','csv'))
    ),
    extensions = 'Buttons',
    caption = htmltools::tags$caption(
     style = 'caption-side: bottom; text-align: left;
      font-weight: normal;',
     '* SNPs, which are associated with log(eGFRcrea) in the all-
     ancestry GWAS meta-analysis (unconditioned) with P<5E-8')
   )
  }
 }
})

#Locus Zoom
Locus_ID<-reactive({
 if(valid_gene()){
  GPS$locus_id[which(GPS$Gene%in%genes_in_gps())]}
})

#create HTML syntax for link to plots
gene_plots <- reactive({
 if(valid_gene()){
  ids=which(links_id%in%Locus_ID())[1]
  paste("<p><a href=\"",links[ids],"\"target=\"_blank\">
   Regional association plot of locus",links_id[ids],"</a></p>")
 }
})

#render links
output$plot_download_links_gene <- renderUI({
 list(lapply(gene_plots(),HTML))
})

# program general gene-search help window:
observeEvent(input$question_GPS_gene,{
 showModal(modalDialog(
```

```
    titel="GPS table",
    tags$h4("GPS table help:"),
    p("Enter a gene name and choose every evidence you want to see
     in the GPS table. Numbers within the CADD-, eQTL- and sQTL-
     columns display the number of credible variants in the
     association signal with the respective characteristics.
     Numbers in the MGI and OMIM column represent the number of
     different phenotypes/diseases found for the respective gene.
     For further information check the Documentation page."),
    easyClose = TRUE,
    footer=NULL,
    size = "m"
  ))
 })

 # program gene-search further options help window:
 observeEvent(input$question_details_gene,{
  showModal(modalDialog(
   titel="Details GPS Table",
   tags$h4("Details on GPS table - Help:"),
   p("Click here for details on the entries you find in the GPS
     table for a gene. If a gene has no entry in the respective
     column the selected option will not give you any result. For
     further information check the Documentation page."),
   easyClose = TRUE,
   footer=NULL,
   size = "m"
  ))
 })

 #show relevant sections:
 observe({shinyjs::toggleElement(
  id="GPS_gene", condition = (rowsGPS()!=0)) # show results section when
  there is at least one searched gene that is included in KidneyGPS
 })

 observeEvent(go$gene,{shinyjs::showElement(id="Gene_div")})

 observeEvent(go$gene,{runjs('
  document.getElementById("Gene_div").scrollIntoView({ left: 0,
  block: "end", behavior: "smooth" });')
 })

 # show the other sections when they are selected:
 observeEvent(go$gene,{shinyjs::toggleElement(
  id="div_cadd_gene",
  condition= (any(columns()%in%c(11:13))&valid_gene()))})

 observeEvent(go$gene,{shinyjs::toggleElement(id="div_glo_gene",
  condition= (any(columns()==14)&valid_gene()))})

 observeEvent(go$gene,{shinyjs::toggleElement(id="div_tub_gene",
  condition= (any(columns()==15)&valid_gene()))})

 observeEvent(go$gene,{shinyjs::toggleElement(
  id="div_GTEx_eqtl_gene",
```

```
    condition= (any(columns()==16)&valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(
    id="div_GTEx_sqtl_gene",
    condition= (any(columns()==18)&valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(
    id="div_GTEx_wo_kidney_eqtl_gene",
    condition= (any(columns()==17)&valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(
    id="div_GTEx_wo_kidney_sqtl_gene",
    condition= (any(columns()==19)&valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(id="div_mgi_gene",
    condition= (any(columns()==20)&valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(
    id="div_omim_gene",
    condition= (any(columns()==21)&valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(
    id="div_drug_gene_kid",
    condition= (any(columns()==25)&valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(
    id="div_drug_gene_oth",
    condition= (any(columns()==26)&valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(id="div_summary",
    condition= any(detail_evidence_locus_gene()=="summary"
               &valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(
    id="div_cred_var_gene",
    condition= any(detail_evidence_locus_gene()=="cred_var"
               &valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(
    id="div_dm_stat_gene",
    condition= any(detail_evidence_locus_gene()=="dm_stat"
               &valid_gene())})

  observeEvent(go$gene,{shinyjs::toggleElement(
    id="div_locus_zoom_gene",
    condition= any(input$gene.locuszoom=="locus_zoom"
              &valid_gene())})
}#End server part

# start Shiny App:
shinyApp(ui = ui, server = server)
```

e) Complete CSS script for KidneyGPS

```
h3{
  padding-right: 50px;
  padding-bottom: 0px;
  max-width: 50%;
}

a:visited {
  color: purple;
}

.question{
  color:grey;
  height:20px;
  width: 20px;
  padding:1px ;
  font-size: 10px;
  border:2px solid grey;
  border-radius: 50%;
  margin-bottom: 6px;
}
.go{
  position: relative;
  float: right;
}

#gpsgo {
  font-weight: bold;
  font-size: large;
  border-width: thick;
  border-color: #428c20;
  border-radius: 10px;
}

#reset_variant,#reset_gene{
  margin-top: -30px;
}

.Output_header{
  font-size: 16px;
  font-weight: bold;
}

#detail_evidence_locus_gene{
  margin-top: -10px;
  padding-top: -15px;
}

.CADD, #CADD_gene th, #CADD th, #CADD_snp_batch th,
#CADD_gene_batch th{
  background-color: #ccffff;
}

.CADD{
  padding: 0px 4px;
}
```

```
.eqtl, #glo_gene th, #glo th, #glo_snp_batch th, #glo_gene_batch
th, #tub_gene th , #tub th, #tub_snp_batch th, #tub_gene_batch
th, #eqtl_gene th, #etql th, #eqtl_snp_batch th,
#eqtl_gene_batch th, #sqtl_gene th, #sqtl th, #sqtl_snp_batch
th, #sqtl_gene_batch th{
  background-color: #ffdb99;
}

.eqtl{
  padding: 0px 4px;
}

.pheno, #div_mgi_gene th, #mgi_gene_batch th,  #omim_gene th,
#omim_gene_batch th{
  background-color: #b3ffb3;
}

.pheno{
  padding: 0px 4px;
}

.drug, #div_drug_gene_kid th, #div_drug_gene_oth th{
  background-color: #d9b3ff;
}

.drug{
  padding: 0px 4px;
}
/*#snp-label, #SNP_batch-label, #file1-label{
  font-size: 16px;
}
*/

.gwscredinputbox .shiny-options-group label{
  font-size: 16px;
  /*font-weight: bold;*/
  padding-bottom: 4px;
}

.gwscredinputbox{
  padding-top: 4px;
}

.snpinputbox, .snplocuszoom{
  position: relative;
  width: min(600px, calc(95% - 15px));
  left: 20px;
  top: -15px;
}
.snpinputbox .shiny-options-group .checkbox{
  padding: 2px 4px;
}

.snplocuszoom .shiny-options-group .checkbox{
  padding: 2px 4px;
```

```
}

.snpinputbox .shiny-options-group .checkbox:nth-of-type(1){
  background-color: #ccffff;
}
.snpinputbox .shiny-options-group .checkbox:nth-of-type(2){
  background-color: #ffdb99;
}
.snpinputbox .shiny-options-group .checkbox:nth-of-type(3){
  background-color: #ffdb99;
}
.snpinputbox .shiny-options-group .checkbox:nth-of-type(4){
  background-color: #ffdb99;
}
.snpinputbox .shiny-options-group .checkbox:nth-of-type(5){
  background-color: #ffdb99;
}
.snpinputbox .shiny-options-group .checkbox:nth-of-type(6){
  background-color: #ffdb99;
}
.snpinputbox .shiny-options-group .checkbox:nth-of-type(7){
  background-color: #ffdb99;
}


#gpscontainermap, #gpscontainersignal, #gpscontainerfilter {
 background-color: #f5f5f5;
 min-height: 400px;
 max-height: 1000px;
 border: 1px solid #dddddd;
 border-radius: 5px;
 padding: 15px;
 margin: 10px;
}

.gps-container {
  display: flex;
}

/*.gps-panel {
  flex: 1;
  display: flex;
  flex-direction: column;
  border: 1px solid #dddddd;
  border-radius: 5px;
  padding: 15px;
  margin: 10px;
}*/



/* Moving the div to the right */
#gpsppafilter {
 margin-left: 15px;
}
```

```
#gpsmapppafilter {
 margin-left: 15px;
}


/*#nofiltersignal, */
#nofiltermap, #nofiltergenelist { /*i have no idea why this
needs to be here, but it has to. */
 position: absolute;
 bottom: 0;
}

#furthersignalfilter label {
  font-weight: normal;
}
/*#nofiltersignal {
 position: absolute;
 bottom: 0;
 margin-bottom: 5px;
}*/
#nofiltermap {
 position: absolute;
 bottom: 0;
 margin-bottom: 5px;
 overflow-x: visible;;
}
#nofiltergenelist {
 position: absolute;
 bottom: 0;
 margin-bottom: 5px;
}

#GPS div.dataTables_wrapper {
 position:relative;
 overflow-x: scroll;
}

#DataTables_Table_0_length {
 padding-right: 2px;
 padding-left: 2px;
 margin-right: 7px;
 /*! margin-left: 3px; */
}

#ppafiltergenemap {
  border-left: 1px dashed grey;

}
.shiny-input-container:not(.shiny-input-container-inline) {
 width: initial;
 max-width: 100%
}
```

```
.Geneofficial, .Locusidwithcomment, .Signalidwithcomment,
.validationwithcomment, .DMwithcomment,
.declinewithcomment,.eQTLglowithcomment,.eQTLtubwithcomment,
.PPA, .REF, .ALT, .CADDscore, .pos, .snpsearch, .maxPPA{
  position: relative;
  text-decoration: underline grey dotted;
}
.Genecomment, .Locusidcomment, .Signalidcomment,
.validationcomment, .DMcomment, .declinecomment, .PPAcomment,
.REFcomment, .ALTcomment, .CADDscorecomment, .poscomment,
.snpsearchcomment, .maxPPAcomment{
  display:none;
  position:absolute;
  z-index:1000000;
  border:1px;
  background-color:white;
  border-style:solid;
  border-width:1px;
  border-color:grey;
  padding:4px 4px;
  border-radius: 6px;
  font-weight: normal;
  color:black;
  left: 105%;
  transform: translate(0, -60%);
  text-align: center;
}
.eQTLglocomment,.eQTLtubcomment{
  display:none;
  position:absolute;
  z-index:1000000;
  border:1px;
  background-color:#ffdb99;
  border-style:solid;
  border-width:1px;
  border-color:grey;
  padding:4px 4px;
  border-radius: 6px;
  font-weight: normal;
  color:black;
  left: 105%;
  transform: translate(0, -60%);
  text-align: center;
}
.Geneofficial:hover span.Genecomment{
  display:block;
  min-width: 100%;
  max-height: 95%;
}

.Locusidwithcomment:hover span.Locusidcomment{
  display:block;
  min-width: 100%;
  max-height: 95%;
}
.Signalidwithcomment:hover span.Signalidcomment{
```

```
  display:block;
  width: 150px;
  min-width: 100%;
  max-height: 95%;
}
.validationwithcomment:hover span.validationcomment{
  display:block;
  width: 200%;
  min-width: 100%;
  max-width: 300%;
  max-height: 85%;
}

.DMwithcomment:hover span.DMcomment{
  display:block;
  width: 200%;
  min-width: 100%;
  max-width: 300%;
  max-height: 85%;
}

.declinewithcomment:hover span.declinecomment{
  display:block;
  width: 200%;
  min-width: 100%;
  max-width: 300%;
  max-height: 85%;
}

.PPA:hover p.PPAcomment{
  display:block;
  width: 400%;
  min-width: 100%;
  max-height: 95%;
}

.eQTLglowithcomment:hover span.eQTLglocomment{
  display:block;
  min-width: 100%;
  max-height: 95%;
}

.eQTLtubwithcomment:hover span.eQTLtubcomment{
  display:block;
  min-width: 100%;
  max-height: 95%;
}

.REF:hover span.REFcomment{
  display:block;
  min-width: 100%;
  max-height: 95%;
}
.ALT:hover span.ALTcomment{
  display:block;
  min-width: 100%;
```

```
  max-height: 95%;
}

.CADDscore:hover span.CADDscorecomment{
  display:block;
  width: 170px;
  min-width: 100%;
  max-height: 95%;
  transform: translate(-200%, -60%);
}

.pos:hover span.poscomment{
  display:block;
  width:170px;
  min-width: 100%;
  max-height: 95%;
  transform: translate(-150%, -60%);
}

.snpsearch:hover span.snpsearchcomment{
  display: block;
  width: 200px;
}

.maxPPA:hover span.maxPPAcomment{
  display: block;
  min-width: 100%;
  max-height: 95%;
}

.Genecomment::after, .Locusidcomment::after,
.Signalidcomment::after, .validationcomment::after,
.declinecomment::after, .DMcomment::after, .PPAcomment::after,
.eQTLglocomment::after, .eQTLtubcomment::after,
.REFcomment::after, .ALTcomment::after, .maxPPAcomment::after{
/* Arrows of the tooltip boxes */
  content: " ";
  position: absolute;
  top: 50%;
  right: 100%; /* To the left of the tooltip */
  margin-top: -5px;
  border-width: 6px;
  border-style: solid;
  border-color: transparent grey transparent transparent;
}
.CADDscorecomment::after{ /* Arrows of the tooltip boxes */
  content: " ";
  position: absolute;
  top: 50%;
  left: 100%; /* To the right of the tooltip */
  margin-top: -5px;
  border-width: 6px;
  border-style: solid;
  border-color: transparent transparent transparent grey;
}
```

```
.poscomment::after{ /* Arrows of the tooltip boxes */
  content: " ";
  position: absolute;
  top: 50%;
  left: 100%; /* To the right of the tooltip */
  margin-top: -5px;
  border-width: 6px;
  border-style: solid;
  border-color: transparent transparent transparent grey;
}

th.hover:hover{
  background-color: #e6e6e6;
}

/*wenn ich einzelne Table-header Cellen farbig machen m?chte,
muss ich denen nur vorher class zuweisen
#GPSrows thead th.Geneofficial  {
  background-color: red;
}
*/

#GPS_rows thead th{
  padding-bottom: 20px;
}

#CADD_gene th{
  height: 75px;
}

.shiny-text-output{
  padding-bottom: 4px;
}

#SNP-input-1 label{
  font-weight: bold;
}

td[data-type="factor"]{
    width: 80px !important;
}
#dm_stat_gene tr:nth-child(1) th:nth-child(3) {
  border-right: 1px solid lightgray;
}
#dm_stat_gene tr:nth-child(1) th:nth-child(4) {
  border-right: 1px solid lightgray;
}
#dm_stat_gene tr:nth-child(1) th:nth-child(5) {
  border-right: 1px solid lightgray;
}

#dm_stat_gene tr:nth-child(2) th:nth-child(5) {
  border-right: 1px solid lightgray;
}

#dm_stat_gene tr:nth-child(2) th:nth-child(10) {
```

```
  border-right: 1px solid lightgray;
}



#dm_stat_gene td:nth-child(3),
#dm_stat_gene td:nth-child(8),
#dm_stat_gene td:nth-child(13) {
  border-right: 1px solid lightgray;
}
```

f) Note on the publication Stanzick et al. 2023

SELBSTSTÄNDIGKEITSERKLÄRUNG

Ich, Kira Julia Stark, geb. Stanzick geboren am 20.10.1997 in Wolfenbüttel, erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.
Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Insbesondere habe ich nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater*in oder andere Personen) in Anspruch genommen.
Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Regensburg, 10.02.2025

Ort, Datum eigenhändige Unterschrift

Danksagung

Zunächst gilt mein Dank meiner Doktormutter, Frau Professor Dr. Iris Heid. Ihre unermüdliche Unterstützung, ihre fachliche Expertise und ihre wertvollen Ratschläge haben maßgeblich zum Gelingen dieser Arbeit beigetragen.

Ein herzliches Dankeschön geht auch an das gesamte Team des Lehrstuhls für Genetische Epidemiologie, bei dem ich meine Forschung durchführen durfte. Besonders möchte ich mich bei Dr. Thomas Winkler für die tatkräftige Unterstützung und die wertvollen wissenschaftlichen Diskussionen bedanken. Seine Hilfe und Zusammenarbeit waren essentiell für den Fortschritt meiner Arbeit.

Ein besonderer Dank gilt meiner Familie, insbesondere meinen Eltern, die mich während meines gesamten Studiums bedingungslos unterstützt haben. Ihre Liebe, ihr Glaube an mich und ihre ständige Ermutigung haben mir die Kraft gegeben, dieses Projekt erfolgreich abzuschließen.

Zu guter Letzt möchte ich meinem Ehemann Klaus danken. Deine Unterstützung, dein Verständnis und deine Geduld haben mir in den herausforderndsten Momenten Halt gegeben und mich stets motiviert, weiterzumachen.

Diese Arbeit ist nicht nur das Ergebnis meiner Anstrengungen, sondern auch das Produkt der Unterstützung und Ermutigung vieler wundervoller Menschen. Euch allen gilt mein tief empfundener Dank.