Download All

BC|INSIGHT User Guide

Version

Version number: 5.17.0

Release date: 14th of October 2020

Content tree

- BC|INSIGHT - 1. Introduction
  - BC|INSIGHT - 1.1 Data architecture
    - BC|INSIGHT - 1.1.1 Form structure
    - BC|INSIGHT - 1.1.2 Dataset structures
    - BC|INSIGHT - 1.1.3 Dataset metadata
- BC|INSIGHT - 2. Navigation
  - BC|INSIGHT - 2.1 Logging into BC|INSIGHT
  - BC|INSIGHT - 2.2 Data Navigator
    - BC|INSIGHT - 2.2.1 Dataset view
    - BC|INSIGHT - 2.2.2 Searching in Navigator
    - BC|INSIGHT - 2.2.3 Folder content table
    - BC|INSIGHT - 2.2.4 Navigation tree
  - BC|INSIGHT - 2.3 Generic search
  - BC|INSIGHT - 2.4 Ontology browser
- BC|INSIGHT - 3. Data management
  - BC|INSIGHT - 3.1 Data forms introduction
    - BC|INSIGHT - 3.1.1 Form families
  - BC|INSIGHT - 3.2. Managing data structures
    - BC|INSIGHT - 3.2.1 Create and edit new forms
    - BC|INSIGHT - 3.2.2 Web form questionnaires
  - BC|INSIGHT - 3.3 Data ontologies and terminologies
    - BC|INSIGHT - 3.3.1 Inbuilt ontologies
    - BC|INSIGHT - 3.3.2 Use of ontologies in forms
    - BC|INSIGHT - 3.3.3 Tools that use ontologies
    - BC|INSIGHT - 3.4.4 Data vocabularies
  - BC|INSIGHT - 3.4 Creating a dataset
    - BC|INSIGHT - 3.4.1 Granting permissions to a dataset
  - BC|INSIGHT - 3.5 Creating subsets
    - BC|INSIGHT - 3.5.1 Simple subsets using filtering options in the data grid
    - BC|INSIGHT - 3.5.2 Using Subset tool
    - BC|INSIGHT - 3.5.3 Joining dataset information in a subset
    - BC|INSIGHT - 3.5.4 Advanced subsets
    - BC|INSIGHT - 3.5.5 Gene range annotations
  - BC|INSIGHT - 3.6 Uploading data
    - BC|INSIGHT - 3.6.1 Update an existing subject
    - BC|INSIGHT - 3.6.2 Add a new subject to the dataset
    - BC|INSIGHT - 3.6.3 Upload a single file
    - BC|INSIGHT - 3.6.4 Upload a file using the upload wizard
    - BC|INSIGHT - 3.6.5 Upload files on server
    - BC|INSIGHT - 3.6.6 Save files to datasets as objects
    - BC|INSIGHT - 3.6.7 Sample-Subject ID conversions
    - BC|INSIGHT - 3.6.8 Revert accidental changes
  - BC|INSIGHT - 3.7 Genomic data management
    - BC|INSIGHT - 3.7.1 VCF data management
    - BC|INSIGHT - 3.7.2 Composite VCF in SQL structure
    - BC|INSIGHT - 3.7.3 Tiled composite VCF

- BC|INSIGHT - 4. Analysis and tools
  - BC|INSIGHT - 4.1 Visualising distribution of data values
  - BC|INSIGHT - 4.2 Conversions and reports
    - BC|INSIGHT - 4.2.1 Pivot datasets
    - BC|INSIGHT - 4.2.2 Aggregate statistics
    - BC|INSIGHT - 4.2.3 Reports
  - BC|INSIGHT - 4.3 Running embedded analyses
    - BC|INSIGHT - 4.3.1 Queue system
  - BC|INSIGHT - 4.4 Analysis results
    - BC|INSIGHT - 4.4.1 Results content
    - BC|INSIGHT - 4.4.2 Uploading results to database
  - BC|INSIGHT - 4.5 R script interface
    - BC|INSIGHT - 4.5.1 R Data input
      - BC|INSIGHT - 4.5.1.1 R Genotypes
      - BC|INSIGHT - 4.5.1.2 R Imputed data
      - BC|INSIGHT - 4.5.1.3 R Omics and multiQTL data
      - BC|INSIGHT - 4.5.1.4 R Phenotypes
    - BC|INSIGHT - 4.5.2 R script Data output
    - BC|INSIGHT - 4.5.3 Storing and sharing R scripts
      - BC|INSIGHT - 4.5.3.1 External R libraries
    - BC|INSIGHT - 4.5.4 R script Examples
  - BC|INSIGHT - 4.6 Genome browsers
    - BC|INSIGHT - 4.6.1 LocusZoom
    - BC|INSIGHT - 4.6.2 Manhattan and QQ plots
    - BC|INSIGHT - 4.6.3 UCSC genome browser
    - BC|INSIGHT - 4.6.4 Embedded IGV
    - BC|INSIGHT - 4.6.5 Data service for IGV and LocusZoom
  - BC|INSIGHT - 4.7 Embedded analysis tools
- BC|INSIGHT - 5. Administration
  - BC|INSIGHT - 5.1 User management
  - BC|INSIGHT - 5.2 User role management
  - BC|INSIGHT - 5.3 Managing dataset ownership
  - BC|INSIGHT - 5.4 User group management
  - BC|INSIGHT - 5.5 Browsing event log information
- BC|INSIGHT - 6. Use-cases and HOWTOs
  - BC|INSIGHT - 6.1 Store BAM and FASTQ files
  - BC|INSIGHT - 6.2 IMPUTE2
    - BC|INSIGHT - 6.2.1 Running IMPUTE2
    - BC|INSIGHT - 6.2.2 Uploading IMPUTE2 results
    - BC|INSIGHT - 6.2.3 Analysis of imputed genotypes
    - BC|INSIGHT - 6.2.4 Optional imputation features
    - BC|INSIGHT - 6.2.5 X chromosome imputation
    - BC|INSIGHT - 6.2.6 Troubleshooting IMPUTE2
  - BC|INSIGHT - 6.3 PLINK analysis
    - BC|INSIGHT - 6.3.1 Example analysis with PLINK
    - BC|INSIGHT - 6.3.2 PLINK results
    - BC|INSIGHT - 6.3.3 Quantitative traits with PLINK
  - BC|INSIGHT - 6.4 Upload PLINK genotype files
    - BC|INSIGHT - 6.4.1 Import genotypes from PLINK files
    - BC|INSIGHT - 6.4.2 Importing pedigree and affection status as PLINK files
    - BC|INSIGHT - 6.4.3 Import marker map from PLINK files

BC|INSIGHT - 1. Introduction

- BC|INSIGHT - 1.1 Data architecture
  - BC|INSIGHT - 1.1.1 Form structure
  - BC|INSIGHT - 1.1.2 Dataset structures
  - BC|INSIGHT - 1.1.3 Dataset metadata

Introduction to BC|INSIGHT

BC|INSIGHT is a data and research management platform for genetic studies. The platform offers tools for complex queries in clinical and genetic data for multiple purposes. BC|INSIGHT scales up from small candidate gene studies to massive international collaboration environments. The platform is used to organize data projects, and to administer user and user-group access to sources.

BC|INSIGHT helps data and research managers to maintain and increase data integrity, quality, and efficiency of work. Research data is stored in a database on a server, but because the user interface is web-based, the system can be used on any computer with a browser. The various implementations of the BC|INSIGHT system is shown in Figure 1.

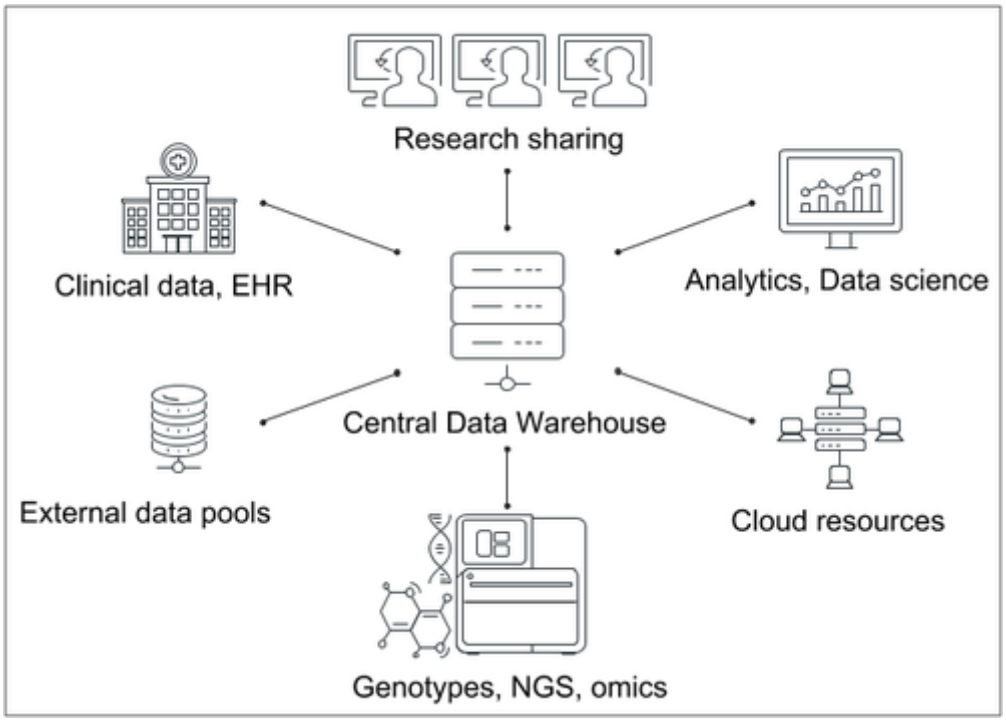


Figure 1. Implementations of BC|INSIGHT.

BC|INSIGHT - 1.1 Data architecture

- BC|INSIGHT - 1.1.1 Form structure
- BC|INSIGHT - 1.1.2 Dataset structures
- BC|INSIGHT - 1.1.3 Dataset metadata

This introductory part of the documentation is meant for brief familiarisation with the fundamental data and storage structures in BC|INSIGHT. Each of these topics is covered in more detail, and with references to use-cases in further chapters of this document.

Structure and storage of data

The architecture of data storage in BC|INSIGHT is a mixture of structured SQL data, structured file storage (tiling, compressed) and unstructured data (user files). All data in BC|INSIGHT, regardless of the underlying storage type, is catalogued in SQL database, and users interact with the data via available management features. The stored data is organised into **datasets**, with distinct ownership, permissions, and data type -specific tools. A dataset consists of three interactive elements: structure defined by **form**, data content, and metadata. BC|INSIGHT datasets can be further classified in roughly four different types based on how their data content is organised: conventional SQL datasets, composite datasets, tiled datasets, and compressed datasets. We will shortly explain in the subsections the basic features of each of these dataset types. Image 1 illustrates the relation between a form and data content.

FORM

VARIABLE	DESCRIPTION	TYPE	KEY
SUBJECT	Subject ID	TEXT	1
SEX	Gender	1=male 2=female	
AGE	Age in years	NUM	
HEIGTH	Height (cm)	NUM	
WEIGHT	Weight (kg)	NUM	

DATASET

SUBJECT	SEX	AGE	HEIGHT	WEIGTH
AA001	2	33	163	52
AA002	2	52	171	65
AA003	1	18	189	76
AA005	1	49	179	91

Image1. How the structure defined by a form relates to the data content in dataset.

A **subset** is a partial view of the original dataset, and is based on dynamic SQL definition. No data is therefore physically copied and all modifications in the original data also update content in the subset. A subset of a dataset can be created to view only the rows that match certain criteria (see Image 2), or several datasets can be joined using either keys of other data content. A subset can also be used to hide data columns.

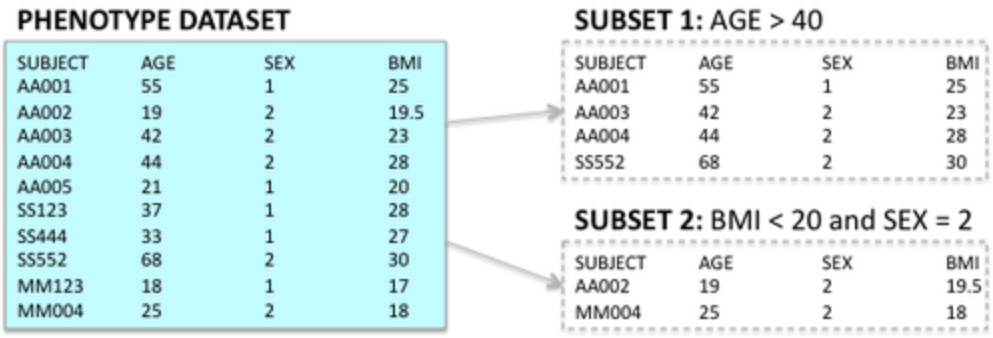


Image 2. Subsets can be used for showing only rows that match user -defined filtering criteria.

BC|INSIGHT - 1.1.1 Form structure

- Form
- Key fields
- Field annotations - ontology

Form

A form describes dataset's structure. The structure consists of typed fields, and constraints to the values stored in those fields. A form must typically define 1-3 **primary key** fields. Key fields are used to define unique data entities in the SQL dataset, and allow users to pick specific entries for analysis. Typically key fields are things like subject identifiers that uniquely identify study subjects, genetic marker names for specific allelic variations, timestamps for longitudinal measurements, sample identifiers, and so on. Key fields are also used to connect data between different datasets. For example, a study subject may have clinical measurements, genotypes, and family structure stored in three different datasets, which can then be pooled together using the study subject identifier.

Field types in a form define what kind of data each field is capable of storing, and what are the accepted values. A field with type Integer only accepts numbers without fractional component, Date type fields only accept dates in recognised format like '2018-04-27', and so on. Table 1 describes the field data types in BC|INSIGHT, and if they can be used as part of the primary key.

Table 1. Summary of variable types.

Variable type	Type name in the form	Notes	Can be key
Text	Text	String variable	**yes**
Integer	Integer	Number without fractional component	**yes**

Float	Float	Number with fractions	no
Date	Date	Date format at data input can be defined	**yes**
Multiple choice	Choice	At least two options needs to be defined, data is stored as a integer in the database	**yes**
Checkbox	Checkbox	Unchecked value is stored as 0 in the DB2 database	no
Paragraph text	Paragraph	32,000 characters can be stored	no
Timestamp	Timestamp	Format: yyyy-mm-dd hh:mm:ss	**yes**
Number	Number	Deprecated: Used in legacy BC forms, either float or integer	no

Key fields

Each dataset has one, two, or three variables to identify the table **primary key**. The key identifies unique rows in a dataset (see Image 1) and also serves as a linking identifier across different data sections. In phenotype tables, the primary key typically contains the subject identifier, but the composition of the primary key is for the user to decide. If the subject ***ID*** field is the only key field, the number of rows in the dataset corresponds to the number of subjects.

In phenotype table, when the subject ID is always of type text in BC|INSIGHT, the second and third key can be either text, date, timestamp, integer, or multiple choice. The date of the laboratory measurement or the hospital visit is often used as a second key. The third key can be used to define, for example, several measurements of a subject in a day. With this key arrangement each subject would have one row in a dataset for each measuring point on each visit.

SUBJECT	VISIT	CHOL	GLUC	TRIG
BC_A2799	2008-06-15	5.50	14.5	55
BC_A2799	2009-08-22	5.30	17	51
BC_A2799	2010-04-30	4.9	16	54
BC_A2883	2010-03-11	5.6	22	80

SUBJECT	BIRTHDATE	SEX	HOPITALCODE
BC_A2799	1963-11-28	1	104
BC_A2883	1955-08-19	2	104
BC_A2891	1959-01-04	1	104

SUBJECT	VISIT	MEDICATIONCODE
BC_A2799	2008-06-15	305
BC_A2799	2009-08-22	305
BC_A2799	2010-04-30	352

Image 1. Primary keys define unique data entries, and link data between datasets.

Although research users can relatively freely choose the structure and keys for their own datasets, BC|INSIGHT comes with a collection of inbuilt forms, which are used to define specific data types, like genotypes, genetic marker information, and data content for omics experiments. When these inbuilt structures are used, the composition of keys is predefined and cannot be changed.

Field annotations - ontology

The form structure contains annotations for the fields, or ontology. BC|INSIGHT uses an internal ontology to describe the content and purpose of fields, information which is then further used by the system tools to make field choices. All BC|INSIGHT ontology terms are defined with 'BC' - namespace, like 'BC:subject', which means a subject identifier. Image 2 shows an example of a form structure that includes field annotations.

Variable details

Id	Key	Description	Type	Choices	Annotations	Required
MARKER	2	Marker ID	Text	-	BC:marker,BC_VARCLASS:marker	yes
SUBJECT	1	Subject ID	Text	-	BC:subject,BC_VARCLASS:patient	yes
AINDEX1		Allele index 1	Integer	-	BC:allele_index	no
AINDEX2		Allele index 2	Integer	-	BC:allele_index	no
IS_PHASED		Is genotype phased?	Choice	0 = No, 1 = Yes	BC:dt_alternat	no
PLOIDY		Genotype ploidy	Integer	-	BC:dt_integer	no
ALT_DOSE_INT		Combined dose of ALT alleles (integer)	Integer	-	BC:dt_integer	no

Image 2. VCF dataset structure defines annotations - or internal ontology - for all fields in the dataset.

BC|INSIGHT - 1.1.2 Dataset structures

- Conventional SQL dataset
  - Files in dataset
- Composite genetic datasets
  - Composite SQL dataset
  - Composite tiled dataset
- Compressed dataset

Conventional SQL dataset

This is the most commonly used dataset type in BC|INSIGHT. It contains structured data, and references to files. An SQL dataset can be thought of as a large spreadsheet, in which dataset fields represent the column headers and data items represent different data rows. BC|INSIGHT has multiple inbuilt forms for various data types, and these forms describe the data with appropriate field annotations. The data itself is stored in an SQL table, and user can interact with the data using various tools in BC|INSIGHT, including some tools that allow SQL to be used directly for querying the data. If the system is configured to provide an ODBC access, the conventional SQL datasets can be queried using an appropriate ODBC client.

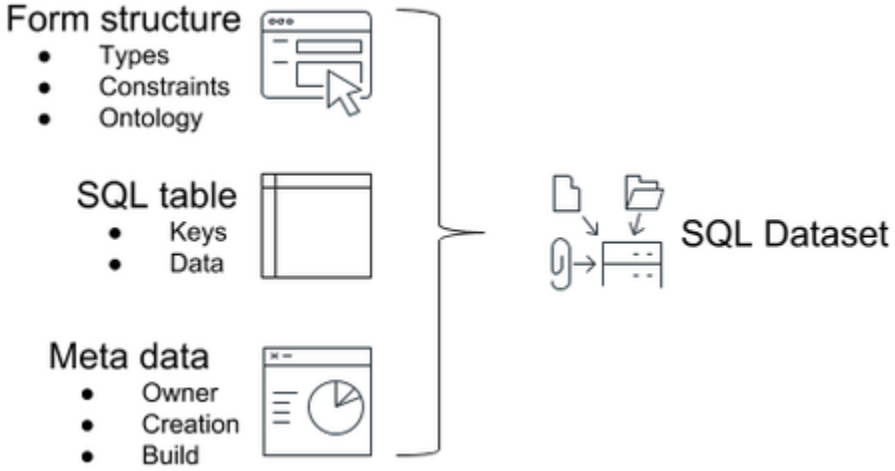


Image 1. The components of a conventional SQL dataset in BC|INSIGHT. Form defines the structure, data is contained in SQL table, and metadata accompanies the dataset.

Files in dataset

SQL datasets are capable of storing file references. Users are able to add files to file storage via Add data -tool, or upload file references from a file that lists the locations of the files in the BC|INSIGHT system. Note that it is possible for the files to be stored in an external file storage or a cloud archive as well. Users are able to interact with the stored files in various ways. When an individual data row is opened with file reference in it, users can either attempt to open it directly, or download it. In case of image and PDF files, the web browsers typically are able to provide a suitable tool for viewing the file. With some other file formats more specialistic tools may be required. The R script interface and external data science tools like Jupyter, are able to access the files as well.

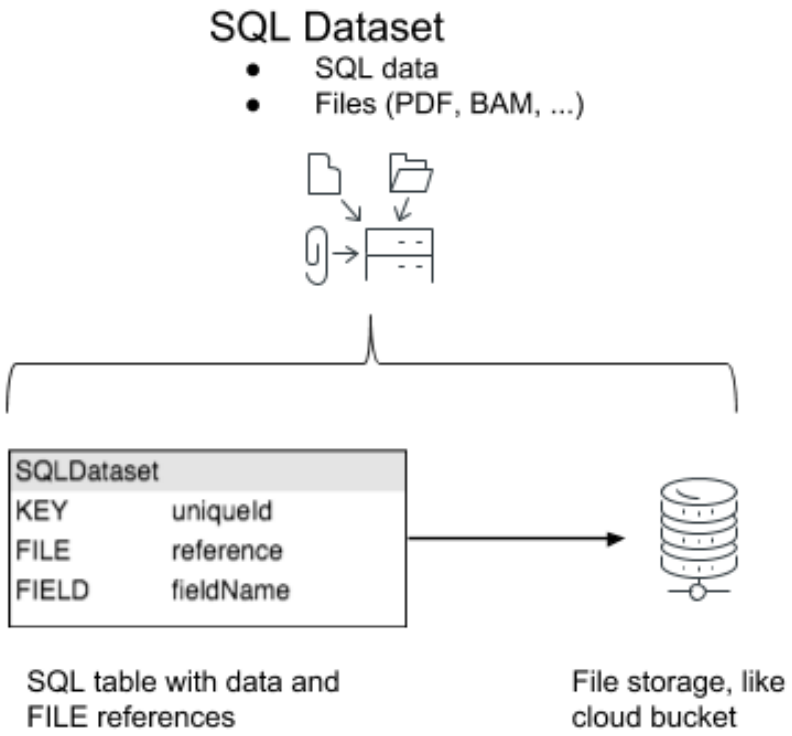


Image 2. Conventional SQL dataset can store references to files, which in turn may be located in the server local filesystem, in external file storage, or in a cloud archive.

Composite genetic datasets

Composite datasets are currently specific for VCF genotype, imputed, and VCF NGS data. The composite dataset as a format is different from the conventional SQL dataset in one very important way: the structure of a composite is not defined by one form. In fact the composite dataset derives it's structure from the incoming data, at first data upload, and generates the necessary forms for itself.

For example, if you create a composite dataset for VCF files, the content of the VCF will determine, how subject and marker information is stored in the database. In each composite dataset the data is split into multiple sub tables, kept together for easier navigation. The data is typically divided into subject data, marker data, and data about alleles. This division will also make it easier for the users to combine the existing genotype or NGS data with external data sources like annotations or phenotypes.

Composite SQL dataset

Composite SQL dataset typically stores VCF data. As germline and somatic VCF data can look very different due to multiple samples per subject, and multiallelic genetic variants, BC|INSIGHT provides two different upload methods for importing of VCFs - one for somatic data and one for germline. See the VCF upload details in later chapters of this document.

In this type of dataset multiple forms are generated for each of the split data tables, and the data itself of contained in SQL tables. Some tables are joined views between the sub tables, to create combined data tables for analysis.

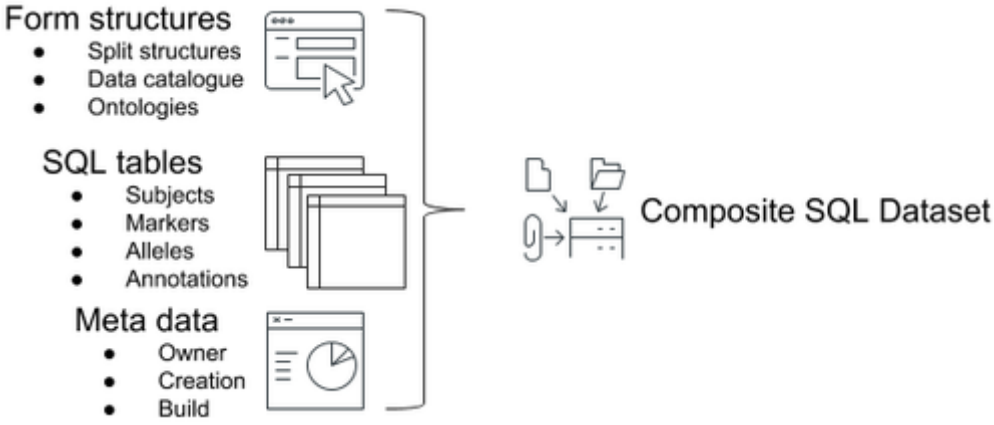


Image 3. Composite SQL dataset has multiple forms, and all data is stored in the SQL database.

Composite tiled dataset

Composite tiled dataset is a structure meant for large volumes of genotype data like imputed genotypes, and large NGS datasets. Accepted input formats are CHIAMO and VCF. The difference to Composite SQL dataset is that the subject genotypes are stored in Tiled data structure, which is a file index for extremely efficient compression and read access for analytical purposes. In addition to the file index, subject and marker vectors are stored as SQL tables for the dataset, for user convenience. This allows joining phenotype and marker annotation data to the Tiled dataset, and filtering information for export and analysis using the external data.

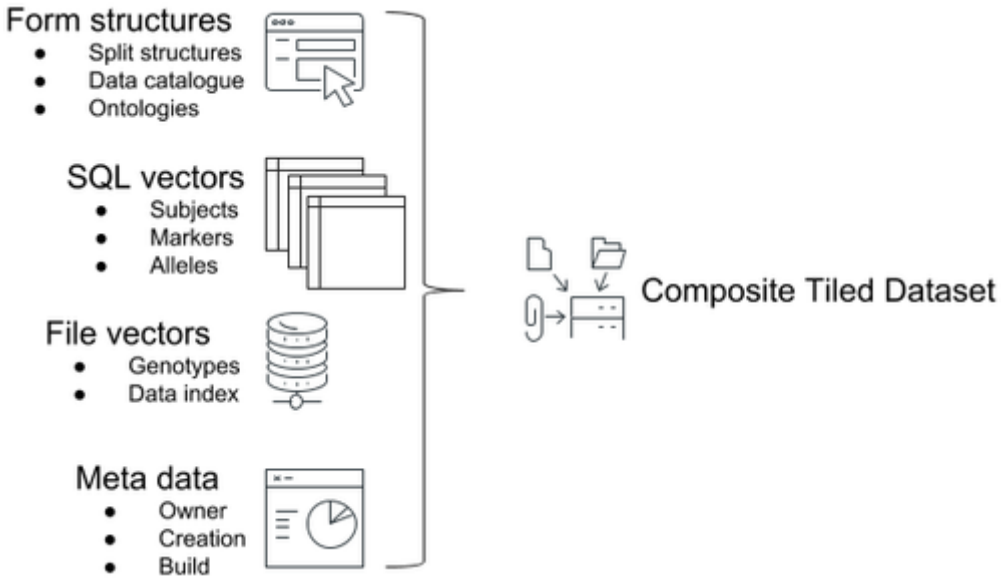


Image 4. Composite tiled dataset stores the bulky genotype data in file index structure, which is highly compressed, and enables very fast read access to data. In addition the subject and marker data are stored as normal SQL tables for more convenient access.

Compressed dataset

Compressed dataset is also known by the name 'BCD image', and is specifically used for storing genotypes (ACGT or 1/2 coded) in a binary file structure outside the SQL table. Note that this is different from Tiled data structure, where genotypes are also stored in files. Compressed dataset is a format, which is used for fast read access in moderate size genotype datasets, but it is not suitable for large data volumes (i.e. over 1000 individuals with 1 million SNPs each). Although it is still possible to create Compressed genotype datasets, the user-interfacing Compressed dataset is being deprecated, and BC Platforms strongly recommends against using this type of dataset directly.

Although we say that Compressed dataset is a deprecated format for user interaction, it has multiple uses under the hood of BC|INSIGHT system. When you save your genotype data in an ACGT dataset, it is stored as such in the SQL database. This makes reading the data inefficient, when the data needs filtering for analysis. BC|INSIGHT keeps an up-to-date BCD image for such datasets, for fast read access and analysis. You may see in the Info -page of a conventional genotype dataset the words 'Image is up-to-date'. It simply means that the BCD image of the genotype data has been refreshed, and the dataset is ready for analysis. If the refresh has not been made, it is typically the very first action the system performs when user needs to read the data from the SQL table. This action shows up in the queue as 'Refresh BCD image'.

BC|INSIGHT - 1.1.3 Dataset metadata

Child pages:

Table of contents:

- Dataset meta-information
  - Species and genome build
  - Dataset meta information
  - Data update logs
  - Update history of a data row

Dataset meta-information

Species and genome build

Species and Genome build in the dataset creation and in the dataset info page are for the optional identification of genome specific meta information of the dataset. Species and Genome build information allows system to control automatically certain aspects of data analysis, for example:

1. Analyzing only species -specific chromosomes
2. Not allowing analysis of implicitly named markers (chr:XXXXX), where chromosomes are not defined in the species
3. Analysis interface is able to show only dataset choices that either match the current species, or where species tag is missing

Supported list of species:

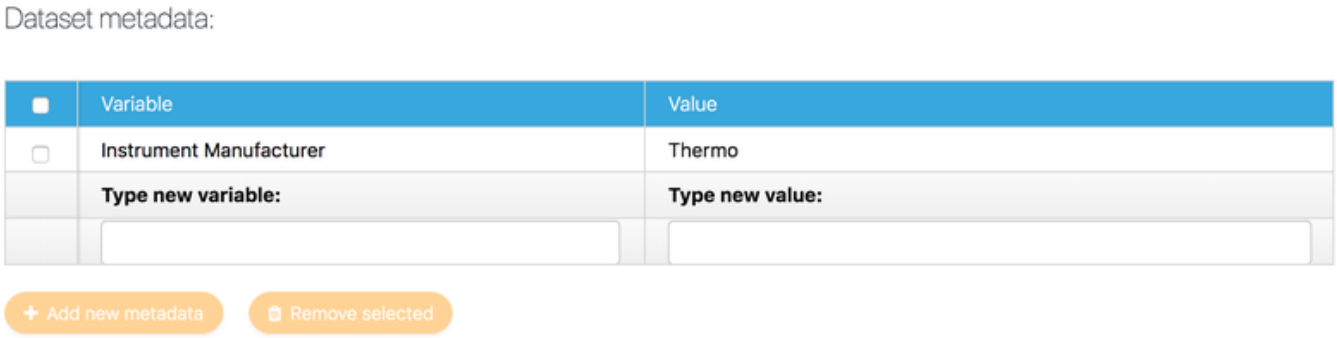
- If you species is missing from the list contact support@bcplatforms.com

NCBI taxon ID	English name	Latin name	No of autosomes	All chromosomes in BC|GENOME (1)
9606	Human	Homo sapines	22	1-22A X:X Y:Y XY:XY MT:MT
10090	Mouse	Mus musculus	19	1-19:A X:X Y:Y MT:MT
10116	Rat	Rattus norvegicus	20	1-20:A X:X Y:Y MT:MT
9913	Bovine	Bos taurus	29	1-29:A X:X Y:Y MT:MT
9796	Pig	Sus scrofa	18	1-18:A X:X Y:Y MT:MT
9940	Sheep	Ovis aries	26	1-26:A X:X Y:Y MT:MT
9796	Horse	Equus caballus	31	1-31:A X:X Y:Y MT:MT
9615	Dog	Canis famialaris	38	1-38:A X:X Y:Y MT:MT
9031	Chicken	Gallus gallus	38	1-38:A W:W Z:Z MT:MT
4577	Maize	Zea mays	10	1-10:A MT:MT
4530	Rice	Oryza sativa	12	1-12:A MT:MT
9139	Turkey	Meleagris gallopavo	30	1-30, W:W Z:Z MT:MT

Dataset meta information

User is able to specify any meta information to be stored with dataset in the dataset info page.

- Variable if a value short name that can contain any characters up to 256 characters expect special characters such as ´ \ | / " [{
- Value can be any string of text specified up to 16384 characters



Data update logs

Update history of a dataset can be found in the Info page. The history is organised into a table structure with timestamp and uploading user account.

Update history of a data row

The data modification history can be viewed for phenotype tables by using the the View entry function. The change history is accessible via an 'info' button next to each variable in the phenotype table. History view is not available with other data types than phenotypes.

BC|INSIGHT - 2. Navigation

Child pages:

User Roles
BC|INSIGHT user

- BC|INSIGHT - 2.1 Logging into BC|INSIGHT
- BC|INSIGHT - 2.2 Data Navigator
  - BC|INSIGHT - 2.2.1 Dataset view
  - BC|INSIGHT - 2.2.2 Searching in Navigator
  - BC|INSIGHT - 2.2.3 Folder content table
  - BC|INSIGHT - 2.2.4 Navigation tree
- BC|INSIGHT - 2.3 Generic search
- BC|INSIGHT - 2.4 Ontology browser

Getting started

To login to BC|INSIGHT, you need both a personal user ID and a password, which are provided by a local BC|INSIGHT administrator. When you receive your ID and password you can log in to BC|INSIGHT.

The BC|INSIGHT user interface is a web-based database platform and it is optimised to be used with the most recent Mozilla Firefox, Safari and Chrome web browsers.

BC|INSIGHT - 2.1 Logging into BC|INSIGHT

User roles
BC|INSIGHT user

Once you have received your ID and password you can log in to BC|INSIGHT:

1. Go to **https://server/bcapp/** (replace server with the name or IP address of your own server).

The account information is provided by your local database owner (usually the project PI).

2. Type your user ID.

3. Type your password.

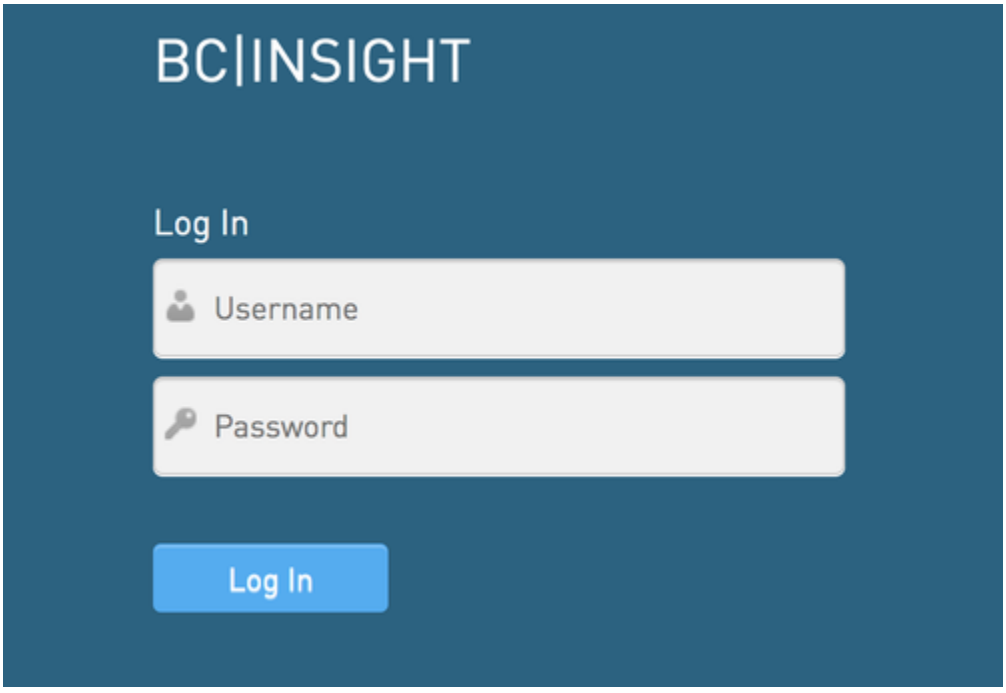


Figure 6. BC|INSIGHT login page.

4. Select **Log In**.

Note

You can enter an incorrect user name and password a maximum of 10 times before access to BC|INSIGHT is locked. If this happens, contact support@bcplatforms.com to unlock the account.

BC|INSIGHT - 2.2 Data Navigator

User roles
BC|INSIGHT user

Child pages:

- BC|INSIGHT - 2.2.1 Dataset view
- BC|INSIGHT - 2.2.2 Searching in Navigator
- BC|INSIGHT - 2.2.3 Folder content table
- BC|INSIGHT - 2.2.4 Navigation tree

Table of contents:

- Application organisation
  - Dashboard
  - Applications menu
- Navigation modes
- Dataset types

Application organisation

When user opens the BC|INSIGHT application, the landing page is organised into roughly three parts. The top contains the application menus for navigating between different apps, and to access user manuals and other resources. The left-hand side serves as the main navigation area, with tree structure for datasets or other data objects, search and filtering tools. The largest area provides various views into the BC|INSIGHT Data Warehouse, depending on what kind navigation mode is being used, and what user has selected.

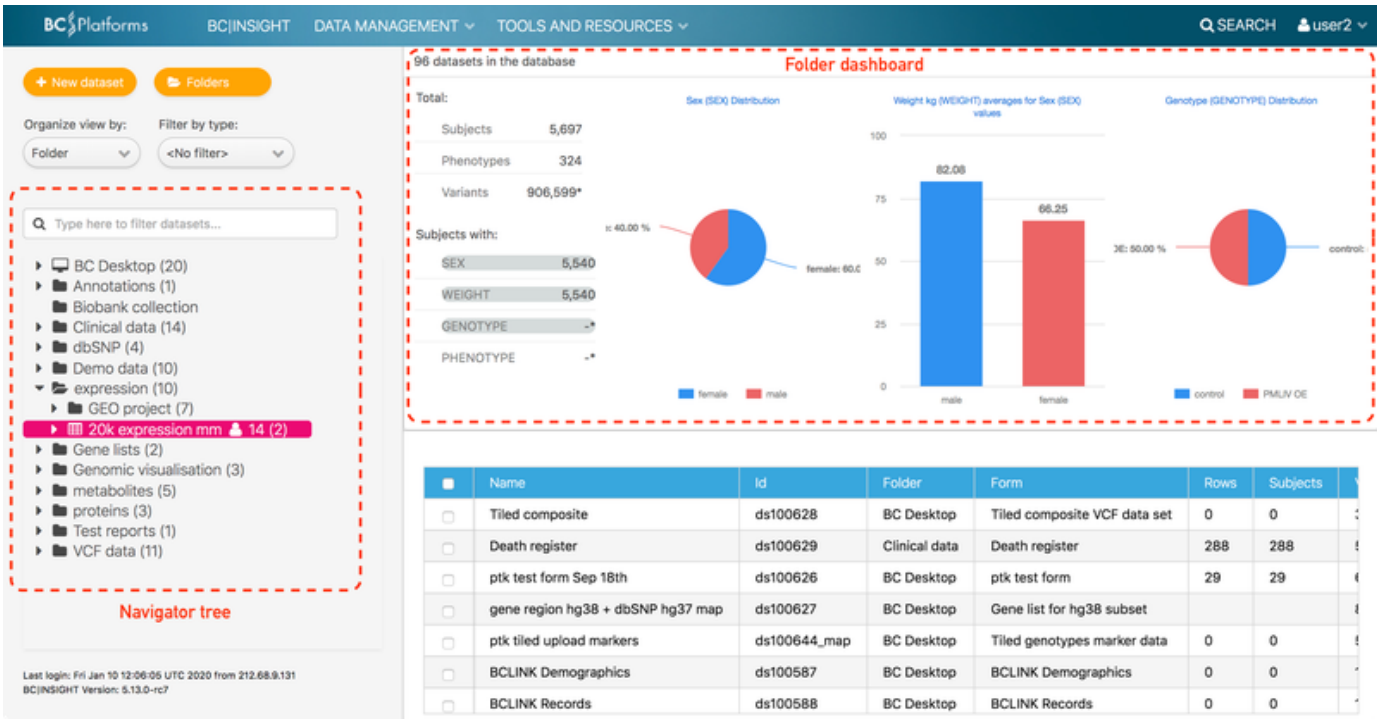


Image 1. The landing page typically shows the left hand side navigation tree with available datasets, and a Dashboard of the root-level of the database, including some basic statistics about subjects, phenotypes and variants available. The Dashboard shows the statistics for the currently selected folder, and at the landing phase this is the root or Desktop folder. Below the dashboard there is the list of all datasets visible to the user, with various details and metadata included.

Dashboard

In some system configurations a data Dashboard is made visible to the users. Depending on the configuration of that Dashboard, the data view will be somewhat different, providing summary statistics on the content of the folder. An example of data Dashboard configuration can be seen in the image above.

Applications menu

The top menu bar has links to other applications in the BC|INSIGHT system. The top link BC|INSIGHT will always take the user to the landing page of the system. The DATA MANAGEMENT menu contains sub menus for

- Subject Browser
- Data Navigator (this page)
- Subset (3.5.2 Using Subset tool)
- Web form editor (3.2.2 Web form questionnaires)
- Structure editor (3.2.1 Create and edit new forms)
- Queue (4.3.1 Queue system)
- Result archive (4.4 Analysis results)
- Cancel job (4.3.1 Queue system)

Navigation modes

The **Organize view by** selector changes the perspective, or the mode, of the navigation tree, and dependent on the mode this may affect the view how the data is being displayed. The '**Folder**' mode is the traditional organization of datasets into user created folders. If a folder contains datasets that are permissible to the user, they will be able to see the hierarchy structure of that folder. Note that users will never be able to see datasets not permitted to them. The '**Type**' mode organises the datasets into groups based on the dataset type (see below). Finally, the '**Ontology** ' mode will not display datasets in the tree, but value ontologies stored in the datasets. This allows users to navigate data using values in ontologies and terminologies stored within the system. See 2.4 Ontology browser for more details.

Dataset types

The **Filter by type** -selector allows user to filter only specific dataset types in the Navigator tree. The table below describes what these different dataset types are, and what kind of data is typically stored in them.

Dataset type	Information
Variations	Variant information (VCF, SNPs, microsatellites), haplotypes, imputed data
NGS	NGS raw files like FASTQ and BAM
Annotations	Marker maps, annotations, frequencies, analysis results
Pedigrees	Family information in the linkage format
CNV	Copy number variation data, derived from SNP or probe data.
multiQTL	Various omics and phenotype data stored in formats suitable for QTL -type analysis
Phenotypes	Phenotype data, including questionnaires, clinical data, measurements, etc
SampleIDs	Sample IDs to subject IDs conversion information for variation data uploads
Scripts	User-made R and SAS scripts
Files	Files indexed in a dataset

BC|INSIGHT - 2.2.1 Dataset view

User roles
BC|INSIGHT user
Internal user
Analyst

Table of contents:

- Dataset view
  - DATA
    - Tools and export
    - Add data, view and edit data
    - Row count button
    - Column button
    - Sorting
    - Refresh menu
    - Subset
  - INFO
  - VISUALIZATION

- STRUCTURE
- PERMISSIONS
- ANALYSIS

Dataset view

When user selects a dataset from the navigator tree in Folder or Type navigation modes, the right hand side view opens a view into the data, and to specific tools and actions available for the selected dataset. Different views are organised under named tabs, which are explained in more details here.

DATA

The Data page of a dataset provides tools for viewing and browsing dataset content in the data grid. In this view user is able to perform multiple data management and reporting tasks. Please see 4.2 Conversions and reports for more information about specific tools available. Some tools come available at the top of the Data Grid when user selects specific rows using the checkboxes, or when user filters the data using the Filter text boxes above each column.

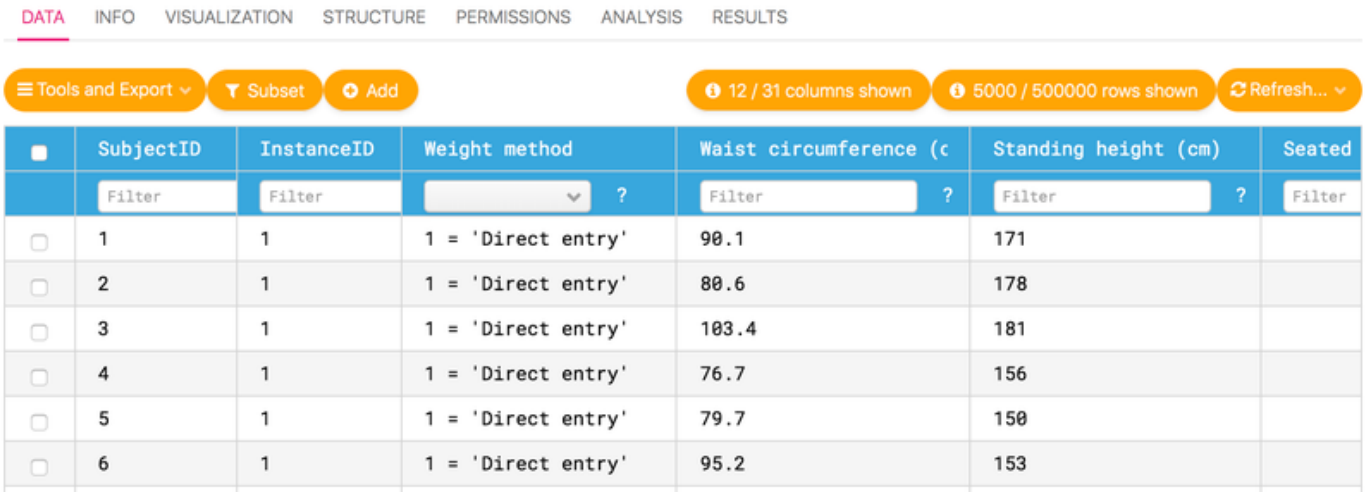
User can filter the data in the Data Grid by typing filter values in the Filter text box, and hitting ENTER button. Multiple filters can be combined. To clear all filters in the Grid, the 'Refresh' button drop-down can be used. Syntax for filtering values depends on type of data in the field.

- In any data type a single data value can be used.
  - Search is case-sensitive
  - Search matches text values anywhere within the text
  - Wildcard characters are not supported
- Numeric values can be filtered using
  - range filters with dash '-', ex. '100-200'
  - smaller and equal values with '<' and <=', ex. '<=100'
  - larger and equal values with '>' and '>=', ex. '>=100'
- Choice variables from choice menu
- Date / timestamp variable can be search by the whole value or year
  - ISO format required 'yyyy-MM-dd', ex. '2017-06-30', or '2017', or '2017-06' respectively for year and year-month combination
  - Range and comparison operators cannot be used.

User can sort the data in ascending or descending order by clicking on the field title. This will then display an arrow symbol indicating the direction of sort. Sort fields cannot be combined.

User can display summary statistics calculated from the dataset field by hovering the mouse cursor over the question mark by the field title. This is available for phenotype datasets only. The summary is different for different data types: Text and Choice fields will show number of total count of rows with a a value, and distribution of 15 most common distinct values. Numerical, Date, and Timestamp fields will show observed minimum, maximum, median, and mean, plus first and third quartiles (Q1, Q3).

When a new new dataset or subset has been created, the statistics for the phenotype table fields are calculated on the background, and will appear, when ready. For very large datasets this may take a few minutes. By default, statistics are not calculated at all to phenotype datasets larger than 10 million rows.



Note

By default BC|INSIGHT Data grid calculates the statistics to phenotype datasets of 10 000 000 or fewer rows in size. This threshold may have been changed to different value by the system administrators.

Tools and export

'Tools and export' provides quick export options in CSV and Excel formats. These export tools star the file download to user desktop directly, without going through the queue. Prepare for long download times with large datasets (100k rows and more), depending on your network speed.

- If the dataset is a view (filtered and/or joined with another dataset), execution of the data export will add to the time required by the system to prepare an export file.

Availability of other tools in this menu depends on the type of the dataset. Commonly available ones are Summary and Text report, for phenotype and variation datasets.

Note
Export features require 'Internal user' -role. Summary and Text report and other statistical tools require 'Analyst' -role.

Add data, view and edit data

If the user has write permission to the dataset, the 'Add' button will be visible. This opens a dialog for adding one row of data, using a web form, which is based on the structure of the dataset.

Row count button

The row count button ('All xxx rows shown') can be used to adjust the maximum number of rows (5000 by default) shown when dealing with large datasets. If the dataset is a Variation dataset, it may require index refresh in order to display true total number of rows. You can refresh the index for Variation data on the INFO page.

Note
The Data Grid accepts maximum number of displayed rows to be 100 000. Values above this are not accepted.

Column button

The column button ('X / Y columns shown') is used to control visibility of fields in the DATA view. The dialog that opens displays a list of available fields, from which user can select the ones they wish to have visible. By default the first 20 columns are displayed. If any fields are hidden, the 'Subset' button will be visible, provided the user has permission to create subsets in BC|INSIGHT. Fields that belong to the datasets primary key cannot be hidden from view. During data export the hidden fields can be left out from the extracted file.

Sorting

By default the data in the grid is sorted by the key columns. The order of key columns is used to determine the final sort order. If, for example, dataset has only one primary key column, like SUBJECT, the default sort is by the SUBJECT column. The order of sorting can be checked in the STRUCTURE tab of the dataset, where the key columns, and their order, is shown. User is able to change sorting from default to a column of their choice, by clicking on the column name at the top of the grid. Clicking more than once will change the order of sort from ascending to descending, and back. Note that only one column can be used for sorting this way. The default sort will not be reset by using the 'Refresh' -menu (see below). In order to restore default sorting. user must select another dataset and reselect the dataset, where reset to default sorting is needed.

Refresh menu

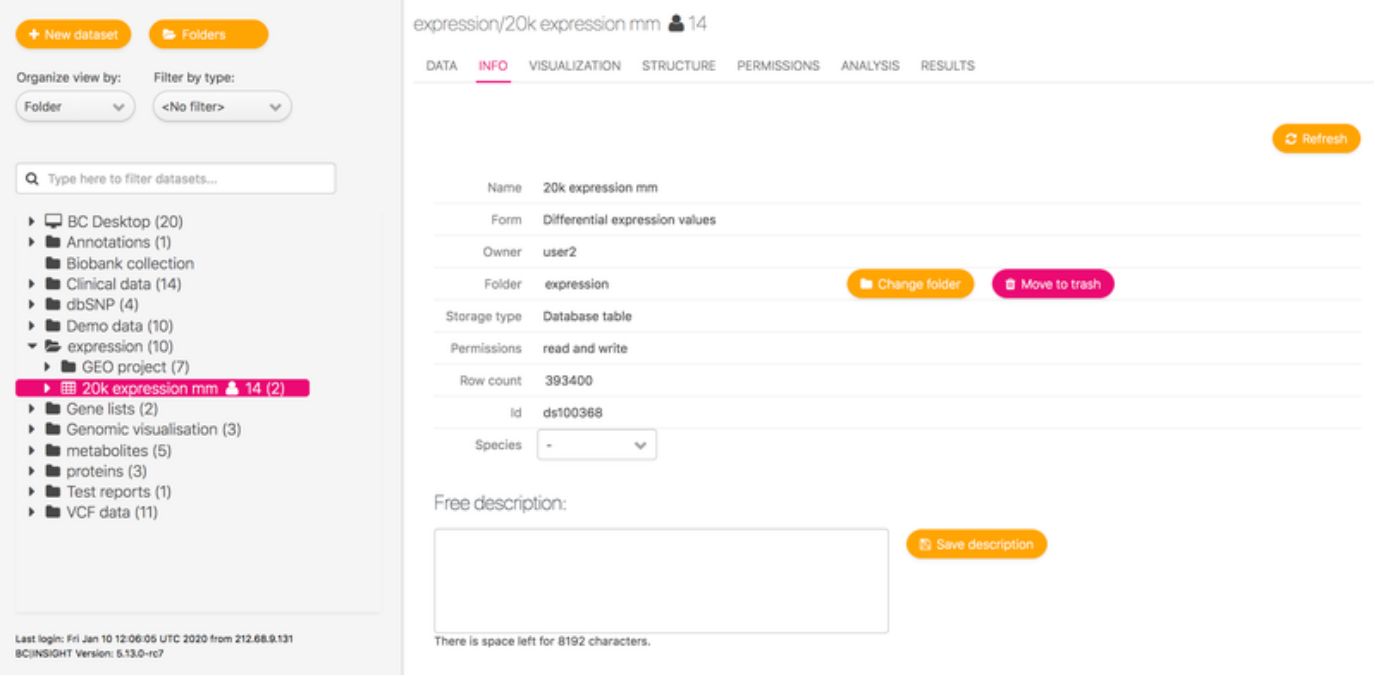
Refresh button is also a drop-down menu, where user can clear all filters from the grid. Refresh will update the dataset content, and is commonly used when changes to data are to be expected, for example after data upload, or adding new rows. Clearing filters will automatically also refresh the grid view. Only the filter values are cleared. Sorting of columns is not reset (see above).

Subset

Subset -button will appear when the dataset has been filtered in any way. The filtering could consists of restricting field values, or visible fields. The system will suggest a name for the subset, but user can override this by typing in their own subset name.

INFO

Dataset Info shows metadata and other information about the dataset. If user is the owner of the dataset, they are able to edit some metadata in this view. For variant datasets this view will also contain the 'Refresh index' tool, for displaying accurate item counts, and to ready the dataset for analytical tasks.



Fields that dataset owner is able to edit by clicking on the field with mouse, are

- Dataset name
- Species (drop down of choices)
- Description

Dataset owner can change the folder of the the dataset with the 'Change folder' button on the Folder row. Trash -tool is right next to it. This will move the dataset to the Trash folder.

Other important information displayed about the dataset is:

- Form used for creating a dataset
- Owner of a dataset is the user who has all permissions to a dataset
- Folder name in which the dataset is shown
- Storage type tells how content of a dataset is stored in BC|INSIGHT for this dataset, see table below.
- Permissions the user has to this dataset
- Row count
- Id is the BC|INSIGHT generated identifier for the dataset

	Physical location or format of data	Value updates	Subset supported
Database table	In SQL database	By row / file(s)	Yes
Compressed dataset	Files in the server	By file(s)	No
Composite dataset	In SQL database, in multiple automatically generated parts	no	Yes
Tiled dataset	BC file system and SQL database, in multiple automatically generated parts	no	Yes, for SQL tables

VISUALIZATION

Visualization page gives the user tools to create different types of charts from the the dataset content. Some dataset types support application -specific visualization tools like genomic browsing, heatmaps, etc. When these options are available for dataset, the button 'Charts' will be available above the graph definition tools. These dataset types are:

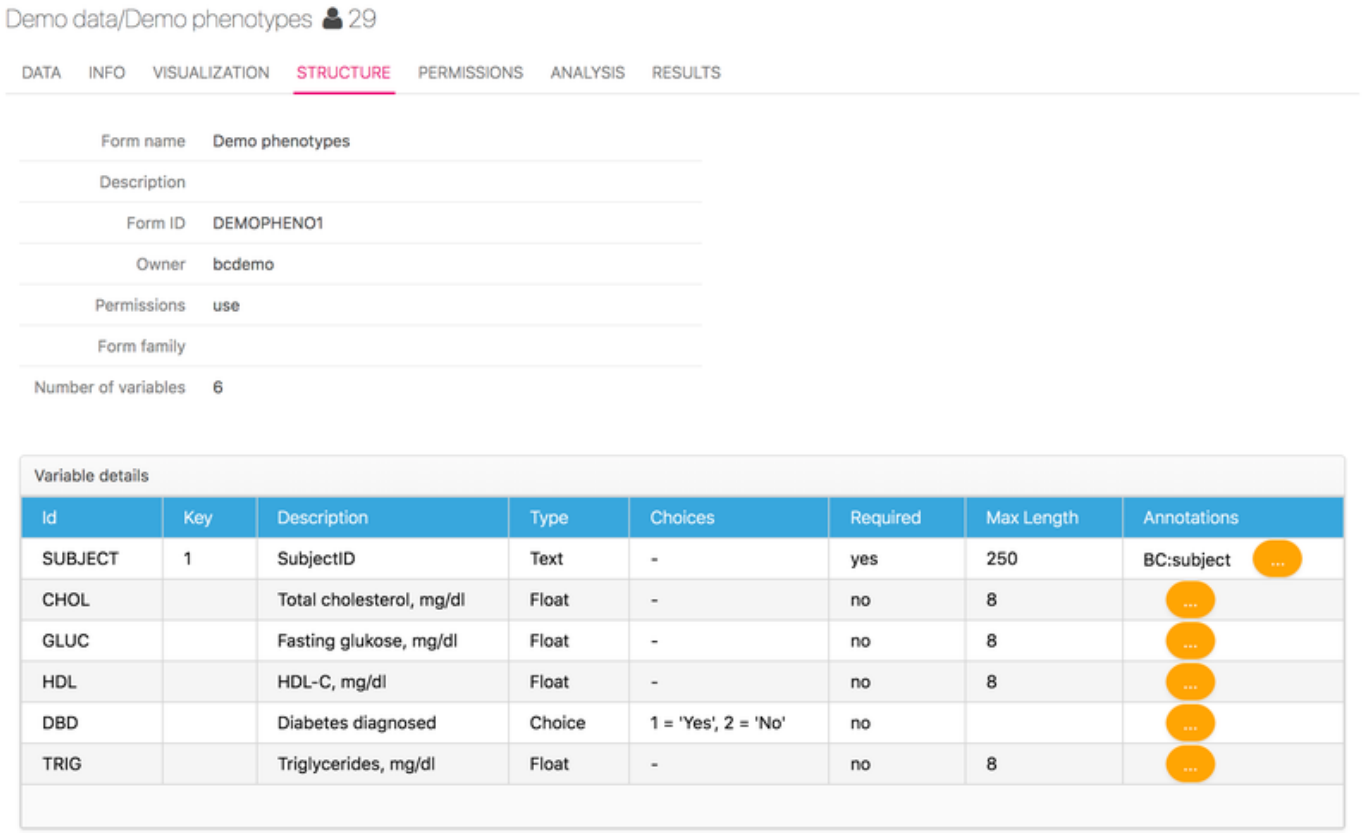
- GWAS results and raw NGS file types (see 4.6 Genome browsers)

- Expression data (see 4.1 Visualising distribution of data values)

All datasets provide visualization of data distribution using the in-build chart definition tools described in detail in 4.1 Visualising distribution of data values.

STRUCTURE

Structure page gives metadata about the dataset's structure, specifically about the form that was used to create the dataset. Form name, and Description are first defined at the time the form has been created, and can be changed via the Structure editor. Form ID is system -generated unique identifier for this form. Many system templates come with more descriptive identifiers. Form owner is the original creator of the form, and is the only person with full permissions over the form manipulation. Permission tells the current user's permissions for the use of the form, typically only 'use'. Permission type 'edit' would allow the user to modify the form in the Structure editor. Form family concept has been explained in section 3.1.1 Form families.



The form grid displays all variables (fields) in the dataset, with metadata about their types, restrictions on values, and annotations attached to each field. Typically annotations are set at form creation time in Structure editor, but form owner is able to change them in this view by clicking on the annotations buttons. See 3.3.2 Use of ontologies in forms for more details.

PERMISSIONS

Permissions page is usable only for the dataset owner. Other users will only see a message that the tool is not usable for them. On permission page the dataset owner is able to grant read and write permissions to the different users and user groups, according to the access and user group configurations in the system. It is possible to make restricted grants to user groups, and then provide wider permissions to individual within the group. The permissions stack together.

ANALYSIS

Content visible in the Analysis page depends on the dataset type, genomic datasets providing typically most options for statistical tools. All datasets have *in situ* R interface for running scripts with copy-paste manner, or selecting pre-saved scripts from a script dataset. Different additional choices can be made in the R script interface for additional data and parameters, depending on the dataset type. See 4.5 R script interface for more details.

> **Note**
> Analysis features require 'Analyst' -role.

BC|INSIGHT - 2.2.2 Searching in Navigator

User roles
BC|INSIGHT user

Table of contents:

- Navigator tree search
- Dataset grid

Navigator tree search

The search field at the top of the Dataset Navigator tree is for quickly finding your way around the datasets. The searches are case insensitive, and scan through multiple parts of the datasets' metadata. Subsets and ontology terms are included in the search scope. To perform a simple search:

1. In Folder and Type -modes, type the search string to a field. It can be used to identify
    a. Dataset owner identifier
    b. Dataset / subset name and identifier
    c. Column names and descriptions
    d. Ontology terms stored in the datasets
    e. Form name and description
    f. Folder name
2. Search matches are shown organised by the navigation Mode selected - Folder or Type.
    a. Clicking the dataset / subset name opens the dataset in the editor canvas on the right
    b. Clicking the right pointing arrow with the folder name (Demo data) opens the matching dataset
    c. Number in brackets shows the number of subsets created using the dataset that user has a permission
    d. Clicking the right pointing arrow with the folder name shows the subsets created using the dataset information
3. In Ontology -mode only the ontology terms in datasets are matched, and the ontology tree displays the matching branches
    a. Datasets with hits are shown in the dataset table, together with the value distribution information

Dataset grid

Content of the dataset navigator can be shown in a dataset grid: It shows the list of all datasets and subsets in the selected folder as well as in the sub folders.

BC|INSIGHT - 2.2.3 Folder content table

User roles
Data manager

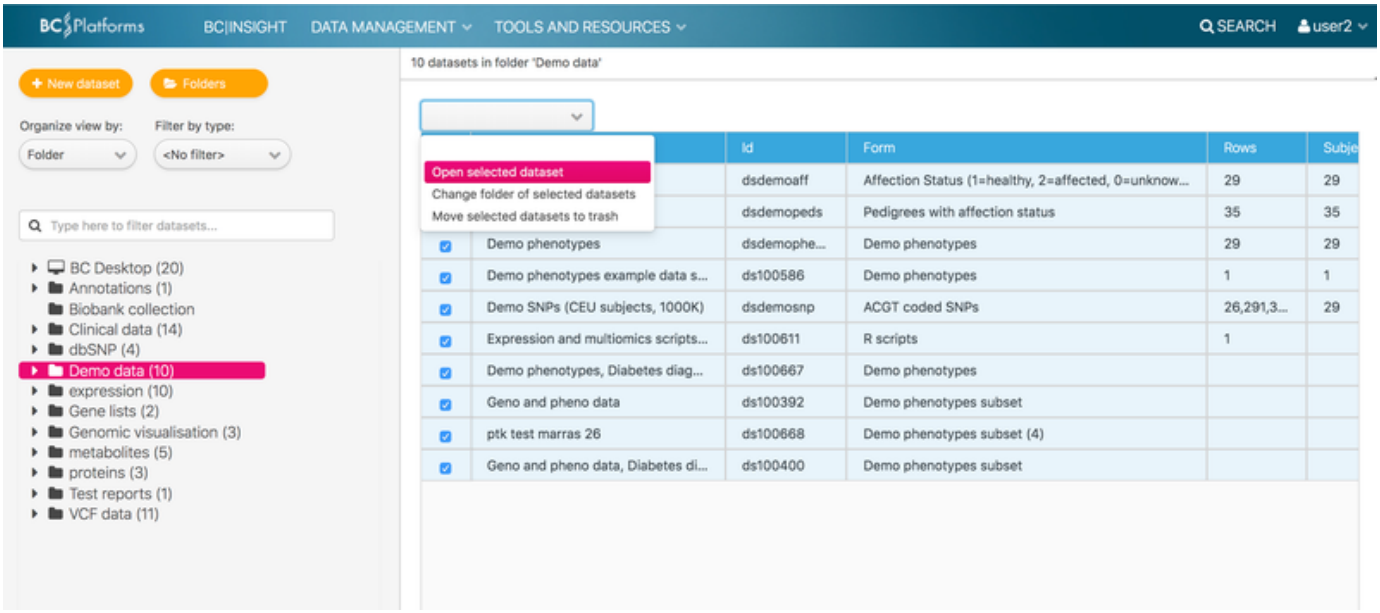
Table of contents:

- Tools available in the folder content table

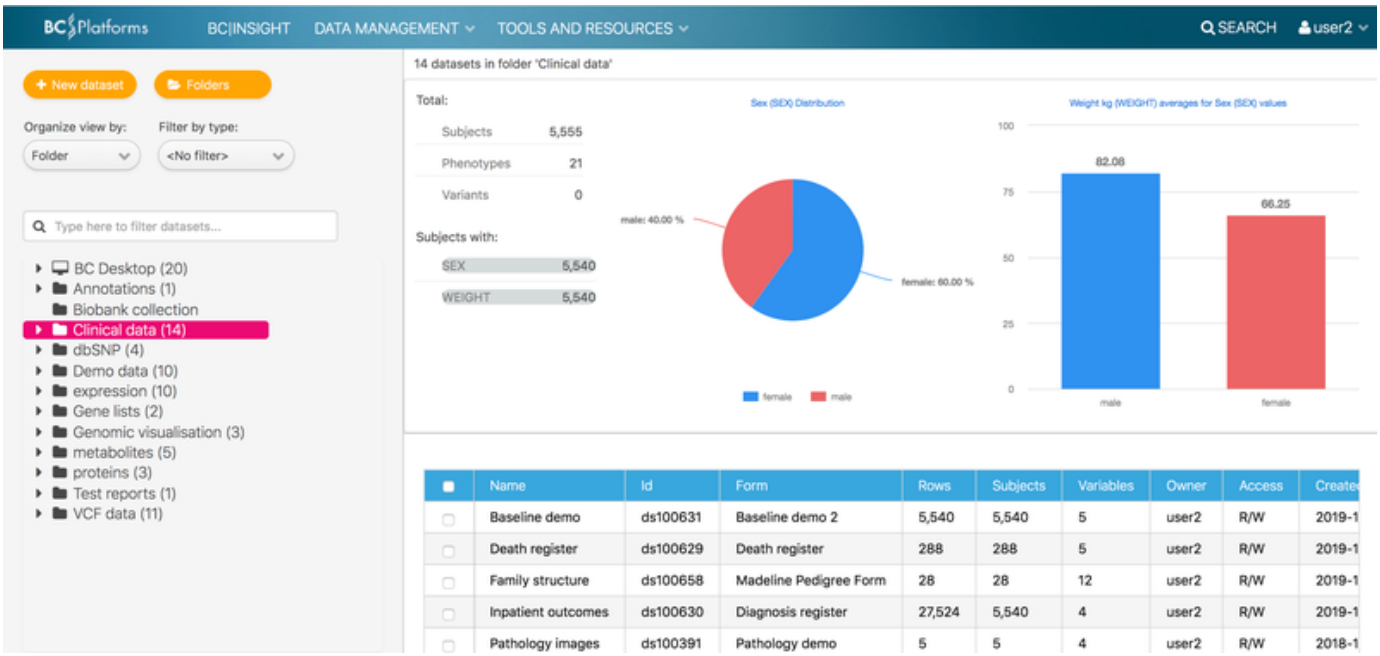
Tools available in the folder content table

When user selects a folder from the Data Navigator tree, the content of that folder is shown in a table containing the datasets. It shows the list of all datasets and subsets in the folder as well as in the sub folders. See figure below for overview on datasets in a table view. In this view user is able to

- Open selected dataset when only one dataset / subset specified
- Change folder of selected datasets for moving datasets to a new folder
- Move selected datasets to trash



In some system configurations a data Dashboard is made visible to the users. Depending on the configuration of that Dashboard, the above view will be somewhat different, providing summary statistics on the content of the folder. An example of data Dashboard configuration can be seen in the image below.



The chart view is controlled for each dataset in the dataset INFO page in Metadata section.

BC|INSIGHT - 2.2.4 Navigation tree

User Roles
Data manager

Table of contents:

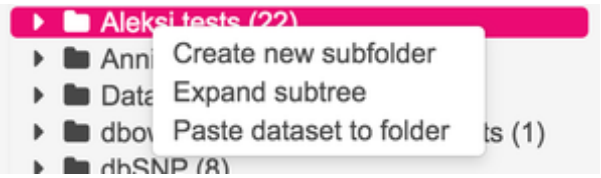
- Navigation tree modes
  - Folder -mode
  - Trash -folder
  - Type -mode
  - Ontology -mode

Folder -mode

The Folder-mode is the default navigation mode in the data tree. This mode displays the conventional folder structure and the datasets and subsets within each folder. By selecting either a folder or a dataset/subset, the right-hand side working area display changes accordingly. Searching dataset in this mode displays search hits as datasets, for user to select. (see BC|INSIGHT - 2.2.3 Folder content table and BC|INSIGHT - 2.2.2 Searching in Navigator).

In the folder mode the following context menu commands are available for folders and datasets:

- Create new subfolder - Create new folder as a subfolder for the selected one
- Expand subtree - Open the subfolder tree, if the selected folder has subfolders
- Paste Dataset to folder - If user has copied a dataset in another folder, this function pastes the dataset into this folder



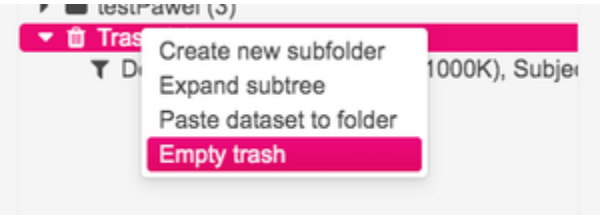
Selected datasets have the following context menu options:

- Copy dataset for moving - This copies the dataset to be moved into a selected folder using the 'Paste dataset to folder' -command
- Mote to trash - Moves the dataset to Trash folder

Trash -folder

The special folder Trash has the same context menu as normal folder, but in addition has the 'Empty trash' command, which will open the list of datasets and subsets contained in the Trash. User is asked to confirm the purge of datasets in the Trash. This will permanently remove all datasets and subsets in the Trash. It is possible to move datasets and subsets away from Trash by selecting the 'Restore dataset' from the context menu, when dataset is int he Trash.

Please note that subsets can only be restored, if their respective parent dataset still exists outside Trash.



Type -mode

Int he dataset type -mode the view of the datasets is organised by dataset type. This often follows the form-type, and therefore the respective context menu in this view allows the user to create new datasets of the same or similar type, than the selected type. The context menu items are:

- Create new dataset with this form - Opens the dataset creation dialog for creation of new datasets of the same type, using the same form
- Expand subtree - Opens the subtree items, if the selected items contains them.

Ontology -mode

Ontology mode is only available if the system has been configured to allow navigation of ontology terms in datasets. In the ontology mode the datasets are not selectable in the tree, but the tree contains data terms, which can be searched for hits in the data. See more about ontology browsing in BC|INSIGHT - 2.4 Ontology browser.

BC|INSIGHT - 2.3 Generic search

User roles
BC|INSIGHT user

Table of contents:

- Datasets, results, and metadata
- Search categories
- Using wildcards and phrases

> **Note**
> The search index is not kept real-time to avoid clashes with user activity and system performance. The index is updated when BC|INSIGHT server is idle. Therefore your search results may not always show very recently added items. The last index refresh time is shown at the bottom of the Search -tool page.

Datasets, results, and metadata

Generic SEARCH tool is opened from the obvious location at the right hand upper corner of the BC|INSIGHT application. You can type your search terms in the text box of the Search -tool page in any order. The search mechanism will give scores to the hits it finds based on relevance and how well the search terms match the data item.

Your hits are displayed in a grid that provides further filtering and sorting for the results. By default sorting is by hit Score and you can change this by clicking on the grid headers. Each hit displays the name of the item the search found, the category of the item, the exact text match justifying the hit, and finally the relevance score.

It is possible to navigate to the source of the hit by clicking the provided link in the hit result name. For example for dataset hits you can navigate to the dataset in question by following the provided link in the table.

Search categories

Each search hit comes from a separate index, which are categorised in following way:

Category	Description
datasets	Datasets metadata and primary key values, and all values type of text. Index includes dataset name, description and other meta. Primary keys like SUBJECT field values are indexed.
form information	Dataset form information. Form names and possible metadata.
result folders	Final reports and result folders.
help information	Online documentation and other help files

It is possible to restrict the search hits to only selected categories, you need to run the search again to narrow down the search scope.

> **Note**
> Results may produce usually low-scoring hits from hidden information stored about each job. This hidden information may include server details and other similar data included in the job metadata but not visible to the user in result page. Check the Match to see where the hit is generated.

Using wildcards and phrases

The Search -tool allows the use of asterisk (*) as wildcard character. You can use it to create search terms that match only the beginning of the words, like 'cohort*' would match 'cohort' and 'cohorts'. It is not possible to use asterisk to mask the beginning of the word.

If you use multiple words or parts of words, the search tool will try to match all words in the documents (logical operator AND). This effectively narrows down your search results when you add new search terms. Remember that order of words in this case does not matter. If you want to match an exact phrase, like a name of a dataset, you should enclose the search in double quotes. See the following example scenario:

```
When dataset name is "Anni's many SNPs":

SNPs many     # matches

"SNPs many"   # will not match

"many SNPs"   # matches
```

BC|INSIGHT - 2.4 Ontology browser

User roles
BC|INSIGHT user
Analyst
Internal user

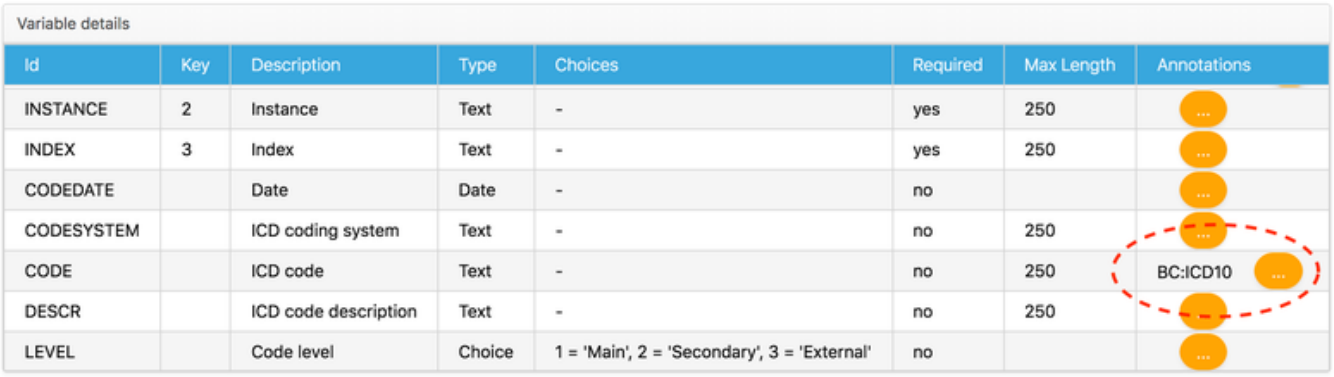
Table of contents:

- Pre-requisites
- Ontology -mode in Data Navigator
    - Searching data using terms
    - Filtering terms
        - Counts
    - View data, subset, and export

Pre-requisites

BC|INSIGHT can provide an ontology navigation mode for your data. In order to utilise this feature, please ask the BC support to enable the ontology browsing in the system. Once this feature is on, you need to establish a terminology (ontology) that you wish to use. Some commonly used and available standard vocabularies like ICD10 are pre-installed in BC|INSIGHT. Ask BC support for more information about currently available ontologies. If you wish to use another standard (licensed vocabulary, in-house ontology) this is possible, but may require conversion of the terminology into a format suitable for use within BC|INSIGHT. Typically the system is able to support OWL and OBO formats.

In order for the ontology terms to become navigable, there must be at least one dataset in the system permissible to the user, where these terms are being used. The dataset field, where the terms or codes are stored, must be correctly annotated, using the BC ontology reference to the respective namespace. For example, in case you have a table 'Outcomes' where diagnostic codes are stored as ICD10 in a field named 'DIAGNOSIS' (any valid field name can be used), you would annotate that field using 'BC:ICD10' annotation term.



Variable details

Id	Key	Description	Type	Choices	Required	Max Length	Annotations
INSTANCE	2	Instance	Text	-	yes	250	...
INDEX	3	Index	Text	-	yes	250	...
CODEDATE		Date	Date	-	no		...
CODESYSTEM		ICD coding system	Text	-	no	250	...
CODE		ICD code	Text	-	no	250	BC:ICD10 ...
DESCR		ICD code description	Text	-	no	250	...
LEVEL		Code level	Choice	1 = 'Main', 2 = 'Secondary', 3 = 'External'	no		...

Once a dataset annotated this way is available, and it contains data, the ontology tree activates the ontology in question, and you can start searching and navigating it. There can be multiple datasets and fields annotated for the same ontology. Remember that if there is no data containing the terms defined by an ontology, the list of terms will not be visible in the ontology browser.

Ontology -mode in Data Navigator

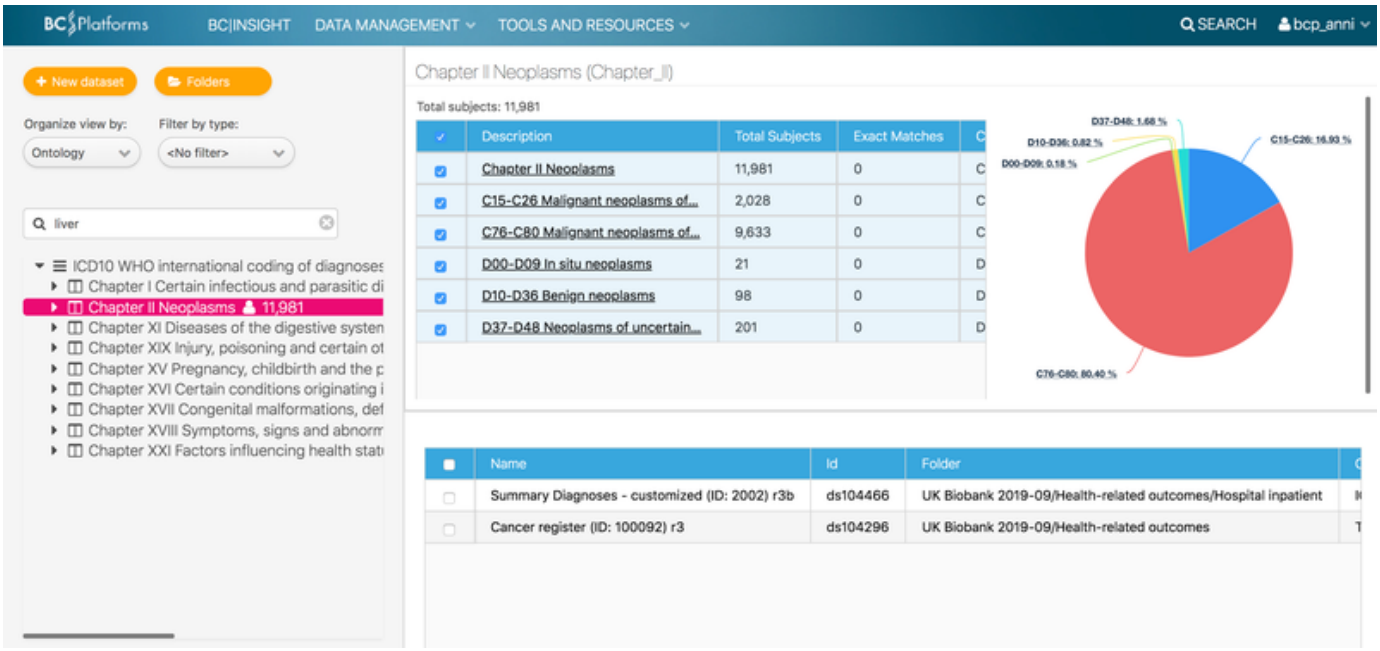
In the Data Navigator tree the Ontology organisation mode switches the tree to display enabled ontologies, instead of a dataset tree.

You can switch to Ontology navigation mode by selecting 'Ontology' option from the **Organise view by** -selector. This will change the view into ontology browser view. You will see your available ontologies and terms. If the ontology is hierarchical, you will see a tree structure. You can navigate the tree as it is, or you can search for specific terms.

Searching data using terms

Term search scans through the terms themselves but importantly also their Descriptions. Typically a term is the code in the ontology, like in ICD10 we could have “J45” as top-level category code for Asthma, where “Asthma” would be the description of that code. These terms are typically not very user-friendly, and users often prefer to search with text describing the terms of interest.

Ontology browser matches terms in case insensitive way, and performs partial term search, ensuring that user will capture as wide selection fo terms as possible. The ontology tree will shrink as less choices become available for the term. When user selects any of the terms in the tree, the Ontology browser will display the content within that term on the right hand side Search hit panel.



Filtering terms

If the search matches a branch or top-level term, the tree displays all sub-terms for the matching term. Equally, it is possible to open the full-parent hierarchy of a matching term, and select any parent branch. The right hand side Search hit panel displays the sub-terms as selectable list, with key attributes like number of hits in the datasets for the exact term, and number of hits in total under the high-level term. A Pie-chart with show visually the distribution of different sub-terms.

In order to only capture data for specific sub-terms, user can unselect terms using the checkboxes. This will narrow the search match scope in the datasets displayed in the Grid below.

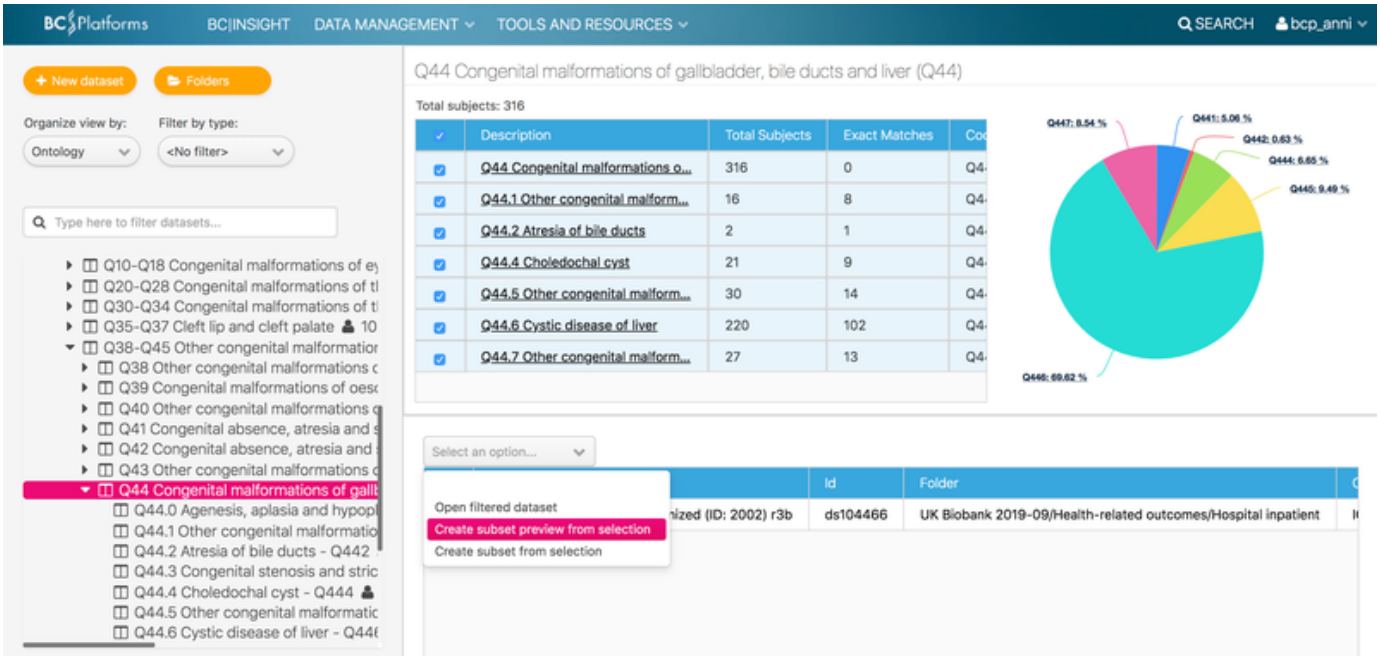
Counts

The Ontology tree always displays the number of subjects with hits from the selected term, or it's child terms. This count is for all datasets captured with the search. The summary table on the right displays the number of unique subjects captured either under the term, 'Total subjects', or from exact hits with that specific term 'Exact matches'. The dataset summary table displays per dataset the number of rows, and number of unique subjects captured with the selected search term and the child terms. The table also contains information about where in the dataset the term was found, giving the name of the column(s) and hit count.

View data, subset, and export

> **Note**
> Subset and Export functionalities are available only to Analyst, and Internal user -roles, respectively.

Under the Search hit panel the user can see a Dataset Grid displaying the datasets that contain hits for the selected terms. This table provides more metadata about the datasets like owner, size, and number of unique subjects (if dataset contains subjects). From this tool user is able to capture the data matching the search and ontology terms.



By double-clicking on any dataset row, user can open a preview of the content in that dataset. Instead of this, user could also select one or more datasets for preview by using the checkboxes in the Dataset Grid, and selecting functions from the enabled drop-down menu.

- Open filtered dataset, (works for only one dataset a a time)
- Create subset preview from selection, creates a joined rpeview of the selected datasets
- Create subset from selection, creates a joined subset of the selected datasets

In the preview mode of single dataset it is possible to create a new subset out of the search results. In the preview mode of multiple datasets it is possible to export the data out in CSV or Excel format, the file is directly downloaded to user's desktop. It is also possible to export a list of unique subject identifiers, in case the dataset contains subject data.

If user needs to export data out in other formats, it is necessary to first create a subset from the search result, and move to the 'Folder' or 'Type' navigation mode for that dataset. This will then provide the more advanced export tools in the subset's DATA -page.

BC|INSIGHT - 3. Data management

User roles

Data manager

Child pages:

- BC|INSIGHT - 3.1 Data forms introduction
  - BC|INSIGHT - 3.1.1 Form families
- BC|INSIGHT - 3.2. Managing data structures
  - BC|INSIGHT - 3.2.1 Create and edit new forms
  - BC|INSIGHT - 3.2.2 Web form questionnaires
- BC|INSIGHT - 3.3 Data ontologies and terminologies
  - BC|INSIGHT - 3.3.1 Inbuilt ontologies
  - BC|INSIGHT - 3.3.2 Use of ontologies in forms
  - BC|INSIGHT - 3.3.3 Tools that use ontologies
  - BC|INSIGHT - 3.4.4 Data vocabularies
- BC|INSIGHT - 3.4 Creating a dataset
  - BC|INSIGHT - 3.4.1 Granting permissions to a dataset
- BC|INSIGHT - 3.5 Creating subsets
  - BC|INSIGHT - 3.5.1 Simple subsets using filtering options in the data grid
  - BC|INSIGHT - 3.5.2 Using Subset tool
  - BC|INSIGHT - 3.5.3 Joining dataset information in a subset
  - BC|INSIGHT - 3.5.4 Advanced subsets
  - BC|INSIGHT - 3.5.5 Gene range annotations
- BC|INSIGHT - 3.6 Uploading data
  - BC|INSIGHT - 3.6.1 Update an existing subject
  - BC|INSIGHT - 3.6.2 Add a new subject to the dataset

- BC|INSIGHT - 3.6.3 Upload a single file
- BC|INSIGHT - 3.6.4 Upload a file using the upload wizard
- BC|INSIGHT - 3.6.5 Upload files on server
- BC|INSIGHT - 3.6.6 Save files to datasets as objects
- BC|INSIGHT - 3.6.7 Sample-Subject ID conversions
- BC|INSIGHT - 3.6.8 Revert accidental changes

- BC|INSIGHT - 3.7 Genomic data management
  - BC|INSIGHT - 3.7.1 VCF data management
  - BC|INSIGHT - 3.7.2 Composite VCF in SQL structure
  - BC|INSIGHT - 3.7.3 Tiled composite VCF

BC|INSIGHT functions are described in this chapter. The main tasks are:

- Creating forms and datasets to be used in BC|INSIGHT
- Uploading data to be used in BC|INSIGHT
- Viewing results and reports in BC|INSIGHT

If you have any additional questions or problems, contact BC support at support@bcplatforms.com.

BC|INSIGHT - 3.1 Data forms introduction

User roles
Data manager

Child pages:

- BC|INSIGHT - 3.1.1 Form families

Table of contents:

- Data structures in BC|INSIGHT
- Form variables
  - Variable data types
  - Restrictions

Data structures in BC|INSIGHT

Most data in BC|INSIGHT is either stored or referenced from SQL table structures, and are called 'datasets'. The structure of a dataset can be pre-built in BC|INSIGHT, or created from scratch by the users. The structure of each dataset is described by a 'form', and one form can be shared by multiple datasets. A form as it's simplest defines the columns, data types, and column descriptions of a dataset, and which columns make up the key for the dataset. These items are directly translated by BC|INSIGHT into SQL language definition of a dataset, and finally a stable structure in the SQL database - a table.

In addition to the simple structure consisting of field names and data types, there is another layer of metadata available. The BC|INSIGHT system can recognise the meaning of different fields or combinations of them, by relying on structural ontologies. Ontologies in BC|INSIGHT are used to tag fields and values to attach meaning of the respective ontology term to that field or value. BC|INSIGHT comes with an inbuilt structure ontology that allows the system to internally make logical decisions about possible uses for data, and to help users in their data selections with various tools. System users are able to apply their own ontologies for structure. Typical value ontologies consist of hierarchical codification of data values, for example using standard ICD10 (WHO) codes for diagnoses, ATC (WHO) codes for medications, or non-standard internal terminologies for codifying patient traits.

Together the forms and the ontologies define the full structure of data and metadata in BC|INSIGHT, and various tools in the system help users to navigate and manage the data utilising these structures.

Form variables

A form describes the structure of data in a dataset. Variables in a form are the form fields, or columns, and they have different attributes depending on their type. The following chapters describe the supported data types, constraints, and other attributes associated with generating data structures. Following chapter discusses the use of data vocabularies, or ontologies, to describe data structures in BC|INSIGHT.

Variable data types

Text: By default max 250 long text field. Feel free to choose shorter field length. Input accepts all characters, except in the case of BC:marker fields, where the genomic marker names have restricted formats.

(**Number**: Used for backwards compatibility to older forms. *Do not use unless you know what you're doing.)*

Integer: Number without fractions, like 1, 2, 100, 1001

Float: A floating point number or number with fractions, like 1.234, 1000.322, or 234.0

Date: Accepted date formats are 2001-12-31 and 2001/31/12. To use other formats, see Upload wizard

Timestamp: Date with attached time element, only accepted format is 2001-12-31 13:13:00

Choice: Enumeration of multiple choices, the data type in SQL will be integer, but user sees human readable descriptions in data

File: You can attach a file to the form field, which is stored in BC|INSIGHT file system

Paragraph: Long text field for large chunks of text, max length 32k

Restrictions

The underlying SQL database imposes the following restrictions to the dimensions of any SQL dataset:

- Max 1012 fields per dataset: which in practice means the same limitation to number of fields in a form, as one BC|INSIGHT form defines one dataset in SQL.
- Max ~1 mega byte of data per row: Different data types occupy different amounts of data in bytes. One character in alphabet takes one byte, so a Text field with length 100 characters will occupy 100 bytes. Date -fields need 4 bytes, Integer fields also 4 bytes, and these all add up towards the maximum limit of one row.

> **Old systems**
> Older installations (before 2018) may still have the limitations imposed in legacy database versions, namely one row could contain max 32k of data.

BC|INSIGHT - 3.1.1 Form families

User roles
Data manager

Child pages:

Table of contents:

- Grouping forms into families
- Functions of form families
  - Genotype and variant data
  - NGS data
  - Omics data
  - Family and pedigree data
  - Phenotypic and sample data
  - Data annotations

Grouping forms into families

BC|INSIGHT groups forms with similar functions and content into families. A form family defines, where the form content is usable, and where not, and how the generic content could be used in various tools and analytical tasks. This model helps the BC|INSIGHT to provide best guesses as to choice of tools at various situations, but also to optimise handling of data, to speed things up. When different datasets are joined together to generate new views, the borders of the form families blur, and become mostly irrelevant for many purposes. Equally it is often necessary to combine information from different form families into one form structure. Form family is being replaced by a more comprehensive internal ontology system in BC|INSIGHT, but many older tools operate on the family concept.

Data type	Use cases	Form families
Variations	Variant information (VCF, SNPs, microsatellites), haplotypes, imputed data	SNP, GENOTYPE, SNP12, SNPPROB
NGS	VCFs, indexing of either gVCF, FASTQ or BAM files	ALIGNMENT, FASTQ, VCFT, (composite VCF forms are automatically generated)

Annotations	Marker maps, annotations, frequencies, analysis results	ANNOTATION, GREGION, FREQ, MAPSNP, MAPKOS, MAPHAL, SNPINFO, VAREFF
Pedigrees	Family information in the linkage format	PEDIGREE, PEDLIST
CNV	Copy number variation data	SNPCNVI, SEGCNV
multiQTL	normalization -omics data used as phenotypic values	MULTIQTL, MQVARLIST, EXPR
Phenotypes	Phenotype values, affection status	PHENOTYPES, AFFSTAT, PHENOLIST, (individual form IDs used in older forms)
SampleIDs	Sample IDs to subject IDs conversion, Sample annotations for experiments	SAMPLEID
Scripts	User-made R and SAS scripts	USERSCRIPT
Files	Files indexed in a dataset	BAM, FASTQ, VCFT, (files are typically simply embedded to forms as variables)

Functions of form families

Genotype and variant data

BC|INSIGHT is able to store and manage multiple different genomic varian types, including microsatellites, SNPs, haplotypes, CNVs, and so on. Some of these data are derived from NGS processes, some from array genotyping, or similar array processes. The common nominator for all these data types is that they describe a variation - in reference to a known consensus genome - in the genome of the individual. This variation typically has a location in a chromosome, and relates somehow to other genomic structures in the vicinity. The reference genome used to decide if a structure is a variant or not, is an important piece of information, and is typically part of the metadata associated with the variant data.

Variations are often subject to annotation tools, where the annotators apply different labels to the variants, based on their location, the nature of the variation, and so on. The annotation data can be attached directly to the variation data using joining and annotation tools provided by BC|INSIGHT.

Typical storage format for variants currently is VCF format. BC|INSIGHT provides a way of automatically generating required structures (forms) based on the content in the incoming VCF. For other genomic data sources the pre-built forms are typically sufficient, at least with minor changes. It also possible to simply store VCF files just as files, in VCFT family of forms.

NGS data

In BC|INSIGHT the category 'NGS' has a strict meaning: not fully analysed NGS process data. These data are typically raw FASTQ files, or files associated with various work steps in the NGS analysis process, like BAM (alignment) files. They are not yet variants. BC|INSIGHT provides some options to work with the NGS files, through script tools, external software packages, and is able to overlay for example BAM data with variant data, it is not fully fledged NGS pipeline platform. For working with NGS pipelines, please refer to other BC products like GeneVision, and BC|INSIGHT.

For NGS data storage there are two structures for storing the raw data files: BAM and FASTQ. FASTQ family has separate forms for paired and single -end FASTQ files form respectively different types of NGS processes.

Omics data

Omics data in BC|INSIGHT is almost the same as "anything that's not genomics variants, or phenotypes". Currently the structures are based on MULTIQTL family, or EXPR family of forms. This means that the basic storage structure is simply a listing of abundance, intensity, or score values for expressed molecular targets, like genes, metabolites, proteins. Also methylation data can be stored in these structures for CG methylation counts. ANNOTATION family is used to bring in external database information about metabolites, gene pathways, etc. Use these to store information from sources like Reactome, Kegg, and others.

Family and pedigree data

The original format for family or pedigree data storing in BC|INSIGHT has been the 'linkage' format. This allows BC|INSIGHT to conveniently export and import family data from commonly used tools available for family data workflows. In BC|INSIGHT we have added some extra support for items like twin-status, which can be utilised by third-party tools embedded in the system. Family structures also allow formation of trios for various purposes, depending on the tools available to the user.

Phenotypic and sample data

All forms created as phenotype forms in the Structure Editor become members of PHENOTYPES family. Older phenotype forms will have their form ID as the name of the form family, which will make looking for them the form list a little more difficult, so you may have to rely on searching

by owner, name, or fields. PHENOTYPES are a very free format, and can contain any data. It is important to understand that PHENOTYPES form can also contain something else than patient data, like sample data, or lists. These forms can have up to 3 key variables defined, and are available for phenotype R scripts and some inbuilt statistical report tools.

AFFSTAT family describes for analytical purposes the affection status of each patient in the dataset. It codes the affected=2, healthy=1, and unknown=0. This enumeration is directly consumed by many statistical algorithms as such, and is also relatively easy to generate from a phenotype dataset, so it is used for convenience.

SAMPLEID family is specifically meant to link subjects to samples, and to attach some meaningful information about the samples to the research data. This could be experimental conditions, or similar information. The family forms are typically used to create conversions from sample identifiers to patient identifiers, and to filter patient data based on the sample attributes.

DEMOPHENO1 is s special family with just one form in it. It is used for demonstration and training purposes, and it always comes with the BC|INSIGHT installation.

Data annotations

These families are mostly dealing with annotation of genotypic data and genomic structures, but also omics data. Ranged annotations are typically stored GREGION family of forms, while things like marker information lists go into structures like SNPINFO. For maps BC|INSIGHT traditionally has offered the absolute chromosomal position map MAPSNP, but also supports centimorgan distances in structures like MAPKOS and MAPHAL (for Kosambi and Haldane, respectively). These latter map types are used by some external tools dealing with linkage based statistics, and typically require presence of genetic family data.

BC|INSIGHT - 3.2. Managing data structures

User roles
Data manager

Child pages:

- BC|INSIGHT - 3.2.1 Create and edit new forms
- BC|INSIGHT - 3.2.2 Web form questionnaires

Table of contents:

- Data structures in BC|INSIGHT

Data structures in BC|INSIGHT

Most data in BC|INSIGHT is either stored or referenced from SQL table structures, and are called 'datasets'. The structure of a dataset can be pre-built in BC|INSIGHT, or created from scratch by the users. The structure of each dataset is described by a 'form', and one form can be shared by multiple datasets. A form as it's simplest defines the columns, data types, and column descriptions of a dataset, and which columns make up the key for the dataset. These items are directly translated by BC|INSIGHT into SQL language definition of a dataset, and finally a stable structure in the SQL database - a table.

In addition to the simple structure consisting of field names and data types, there is another layer of metadata available. The BC|INSIGHT system can recognise the meaning of different fields or combinations of them, by relying on structural ontologies. Ontologies in BC|INSIGHT are used to tag fields and values to attach meaning of the respective ontology term to that field or value. BC|INSIGHT comes with an inbuilt structure ontology that allows the system to internally make logical decisions about possible uses for data, and to help users in their data selections with various tools. System users are able to apply their own ontologies for structure. Typical value ontologies consist of hierarchical codification of data values, for example using ICD10 codes for diagnoses, or ATC codes for medications. Both of these examples are coding systems provided by WHO, and the codes are typically used as values in clinical datasets.

Together the forms and the ontologies define the full structure of data and metadata in BC|INSIGHT, and various tools in the system help users to navigate and manage the data utilising these structures.

BC|INSIGHT - 3.2.1 Create and edit new forms

User roles
Data manager

Table of contents:

- Create new form
  - New form from scratch
  - Generate
  - Copy
  - Import
  - Making forms usable - publishing
- Editing forms
  - Published vs unpublished

Create new form

Structure Editor can be found from the top-level menu in ***DATA MANAGEMENT DATA STRUCTURES STRUCTURE EDITOR***. You will see a listing of existing forms, which you can filter based on name, form family, owner, form status, and so on (Image 1).

Image 1. Listing of forms in the BC|INSIGHT, filtered by name.

To create new form you have multiple options. You can either use the 'New form' -tool to start from scratch, you can copy an existing form and edit the copy, you can import a form structure from file, or you can provide an example data file, which is then used by the Generate -tool to create a guess about the structure of your data.

Note that with 'New form' and 'Generate' -tools you are only able to create forms that belong to the Phenotypes form family. Explanation about form families can be found in the chapter Form families. If you need to create a form that belongs to some other form family, look for forms that belong to that family (filter for Family and for status 'System'), and copy that form for yourself. Then you are able to add new information and create your own version of the form.

New form from scratch

By selecting 'New form' you open the editor application, where you are able to add new columns, remove them, and apply ontology terms to the form variables. Note that by starting from scratch you are not able to create other than Phenotypes -family forms. You should give the form a unique name, and a useful description.

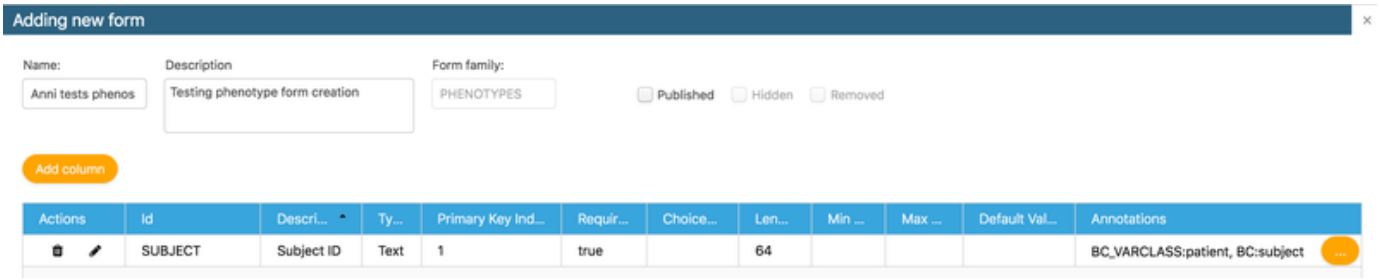


Image 2. Starting from scratch. You can use Add column to create new variables.

Once you have created a new field with the Add column button, a new row appears in the form, with dummy name and description. You should now edit the column by clicking the editing icon on the left hand side. Editor dialog gives you the options to set the variable ID, which will become the SQL name for the dataset in the database. Therefore the format of the column ID is very restricted in length and structure. If you attempt to use a non-conforming ID for the field, the system will attempt to automatically correct it. The Description of the variable is more free, but must not exceed 250 characters.

You then need to select the variable type from the drop-down menu. Please see the chapter Form structure for your options. If you create a Text field, you should consider changing the maximum length for the field. If you choose a Choice set, you can edit the choices after you have saved the column.

You can at this point also add a new key. Forms support up to three key fields, to create unique entries in the dataset. In the example of Image 3, a Date -type field named Visit is made into second key in the form, in order to collect longitudinal data from follow-up visits.

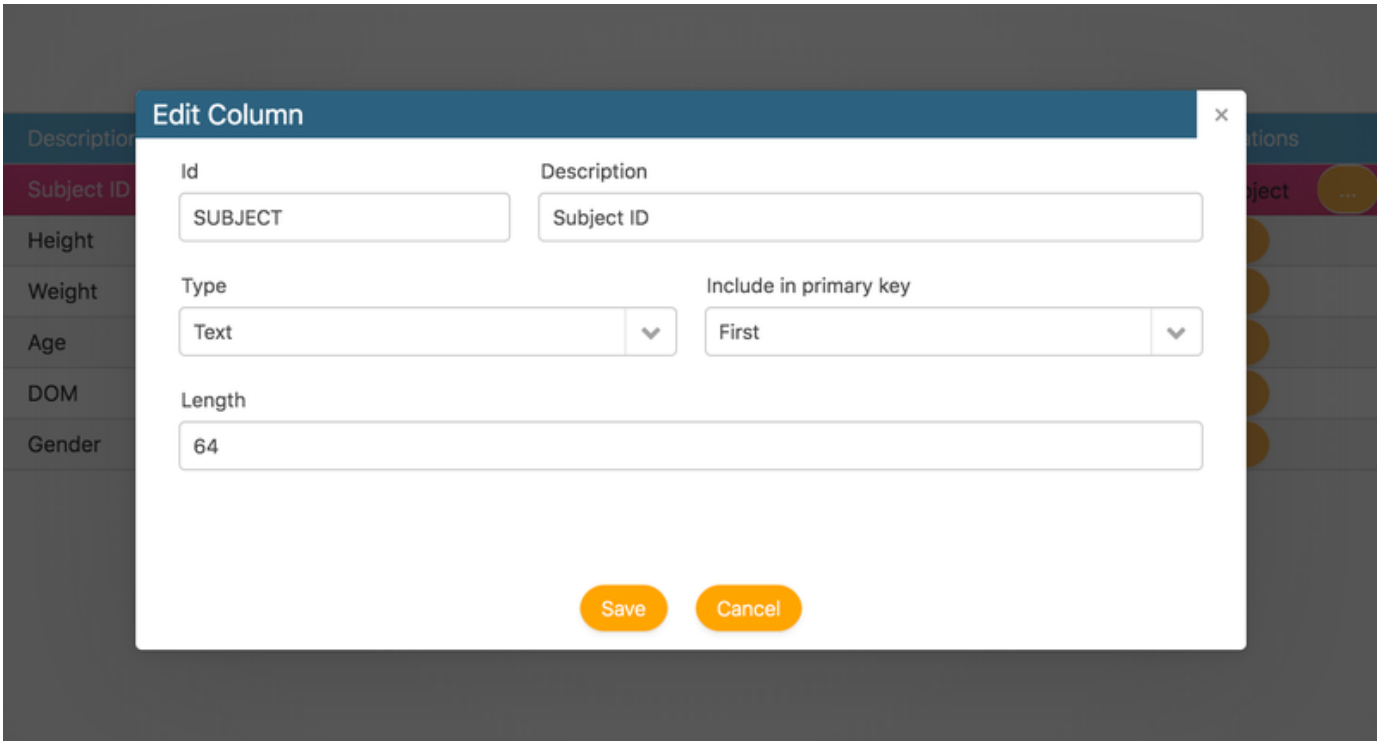


Image 3. Editing a form variable. This variable will become the first key in the form - a Subject ID.

After you have created the form variables you can save your work from the Create -button at the bottom of the editor.

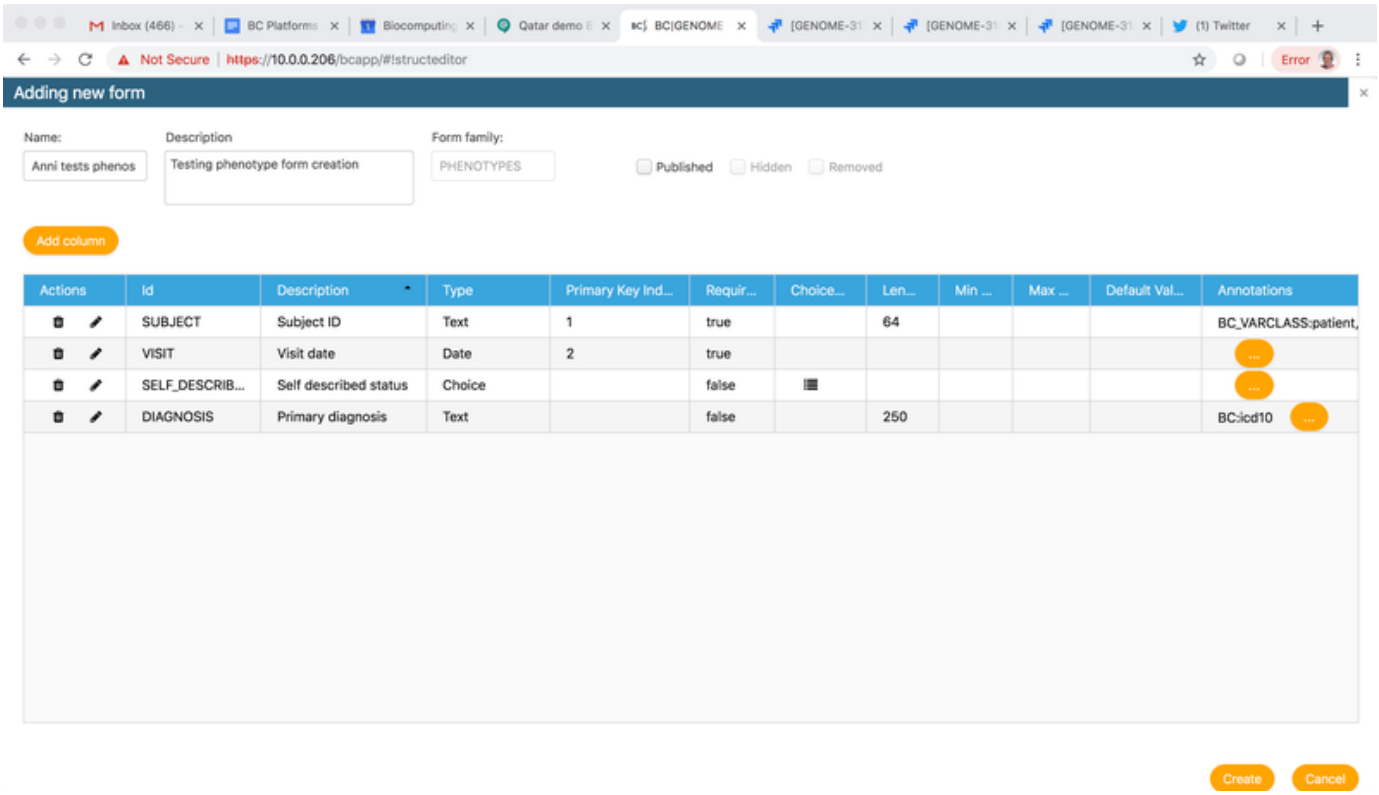


Image 4. An overview to a form ready to be created.

Generate

If you have a representative file of your data in CSV or TSV format, you can give it to the Generate -tool. The tool will attempt to give a best guess of the structure of the form required to host the data in the file. It follows a certain rigid logic, and it always requires your decision in the end to either accept, or make changes to the generated form. Always have a look at the structure before taking it into use. Check the following:

- You have at least one key field, and if you know your data needs more keys, check these are set appropriately
- Make sure any variable set as Integer will not be required to contain numbers with fractions (floats)
- Date fields and timestamps appropriately assigned
- Choice sets are generated from the existing values in the data file, you may want to double check these, and add more values if needed
- Length of text fields

Copy

In the list of forms, by each form you have the tool for making a copy of the form. This is used to create new versions of existing structures, but also to expand on an existing in-built form, which you are not able to create from scratch, like genotypic annotations, or similar forms. Once you have created a copy, the copy becomes under your ownership and you are free to edit it. However, it is good to keep in mind that if you copy a form in order to create an extension to a system form, it is usually a good idea to leave the ontology terms from the original form untouched. This will make sure that the new form is still recognised correctly by the BC|INSIGHT tools as being a member of a certain family of forms.

Import

You can use the Import -tool to provide a form file that describes the structure of your form. An example below shows the minimum things required to describe a form. Note where the multiple choice question's options are defined. You refer to the multiple choice options by their choiceset number, in the example '0'. Use tab-character to separate values in the file, and always make sure each row has the same number of fields defined, even if they are empty.

```
<choicesets>
0        1=male        2=female
</choicesets>
<variables>
VARIABLE        DESCRIPTION        TYPE        KEY        MINVAL
MAXVAL        CHOICESET        REQUIRED        TEXTLENGTH
VALUEFUNCTION
SUBJECT        Subject        Text
1                                        64
AGE        Age in years
Number
HGT        Height, cm
Number
WGT        Weight, kg
Number
DOM        Date of measurements
Date
SEX        Gender        Choice
0
</variables>
```

The fields in the form file are presented within the <variables> element. Always remember to include the closing element </variables>.

VARIABLE: the ID of the column

DESCRIPTION: the user-friendly name for the column

TYPE: can be any of the supported types, see Form structure

KEY: if this column is also a key, you can choose key order from 1, 2, and 3

MINVAL: for numerical and date values, minimum allowed value

MAXVAL: for numerical and date values, maximum allowed value

CHOICESET: the index of choice options used here, choice sets are defined within the <choicesets> element

REQUIRED: Key fields are always required, but set this to True for other fields if needed

TEXTLENGTH: for text values, max length

VALUEFUNCTION: not used in this context, ignore

Making forms usable - publishing

You have already probably noted the checkbox "Published" at the top of the editor window. If you check this box and save the form, it will become visible and usable in the dataset creation dialog in the Data Navigator. If there are datasets already created using the form, you are no longer able to hide the form by unticking the box. All existing datasets built using that form must first be removed from the system. You can see the status of this in the form listing, where the number of tables created is shown.

Published vs unpublished

You can freely edit unpublished forms. There are no restrictions if you are the owner of the form. If you do not own the form but have visibility to it, you are allowed to take a copy for yourself for editing.

Published forms are editable only if they do not yet have attached datasets, i.e. no datasets have yet been created using the form. If this is the case, it would be strongly recommended to make the form unpublished, by unchecking the 'Published' checkbox and saving, before proceeding to edit the form. If somebody uses the form to create a new dataset whilst you are editing it, the dataset may end up in unusable state.

If there are datasets created already using the form. You can start editing the form, but a new copy will be created for the editing. The name of the for you are editing will stay as original, but **all datasets already created will have their form renamed to include the work 'old' in it**. This is to make sure the structure of the already existing datasets is not compromised, and also that users swill be aware that a newer version of the original form exists.

BC|INSIGHT - 3.2.2 Web form questionnaires

User roles
Data manager

Table of contents:

- Web forms introduction
- Web form parameters
  - SUBJECT ID (unique identifier for subject)
  - VARIABLE ID
  - TYPE
  - KEYS
  - REQUIRED
  - CHOICESET
  - MAX TEXT LENGTH
  - MINIMUM AND MAXIMUM
  - VALUE FUNCTION
  - DEFAULT VALUE
- Create webform from scratch
- Import webform from a file
- Publish a form
- Form settings
  - Autosave
  - Date format

Patient questionnaires can be created and structured using the Web form editor provided with BC|INSIGHT. Web form editor is a tool for constructing web pages with fill-in fields, selection menus and checkboxes that are aware of the required data type, and can enforce restrictions

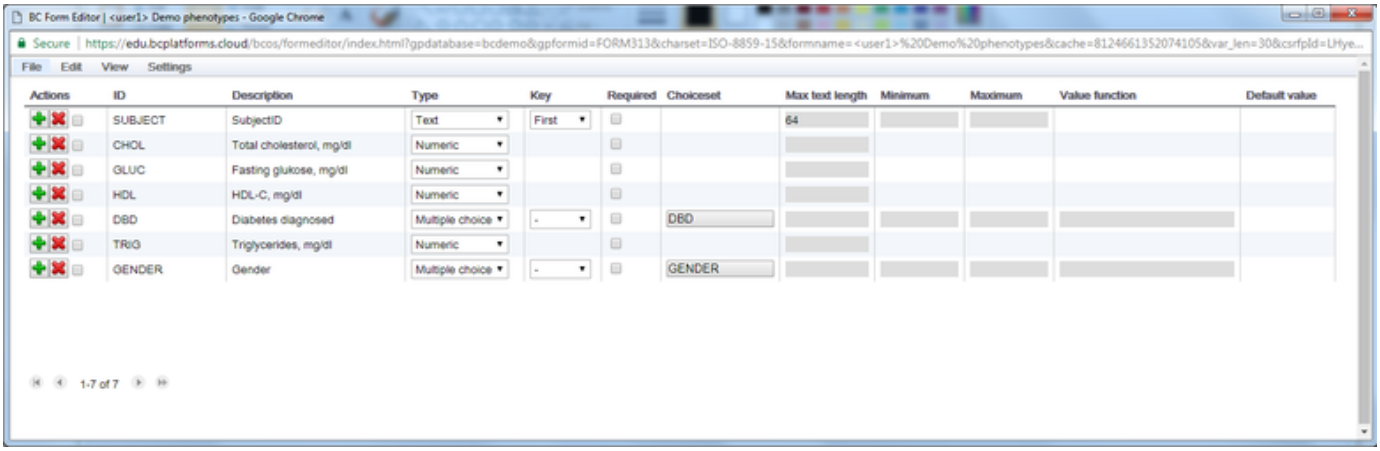
to the data being entered. A web form will typically contain some logical structure for grouping questions together, calculating values on the fly based on previously filled-in values, and providing practical guidance for the users filling in the forms.

A web form can become a highly customised and controlled way of collecting phenotypic and clinical information for projects, stored directly to the BC|INSIGHT database for further examination and analysis.

User can access Web form editor in the top menu ***DATA MANAGEMENT DATA STRUCTURES WEB FORM EDITOR*** .

Web form parameters

Variables are used to define the fields in a form. Variables in a form can be added, removed or edited. In BC|INSIGHT, each variable has a set of variable attributes assigned to it. The form displays in the ***Spreadsheet editor*** view and shows several variables. You can edit, remove or add variables.



> **Note**
> For every form, it is mandatory to have a primary key variable that is used to identify each record. By default, when a new form is created, the SUBJECT variable (visible as the first variable) has the primary key assigned to it and is used as a unique identifier for the record.

SUBJECT ID (unique identifier for subject)

By default, when a new web form is created, the default variable ***SUBJECT ID*** is created to make a *unique identifier* for the record.

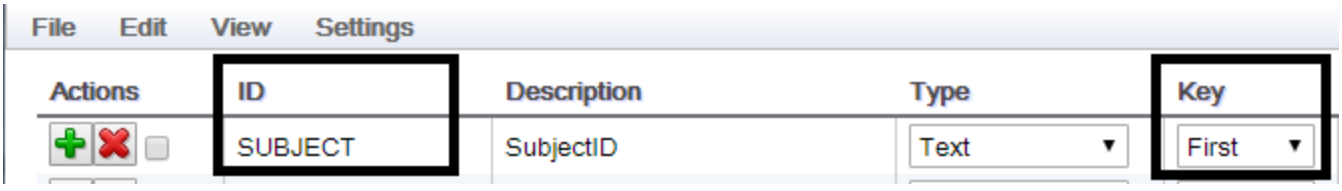
This variable ID must have a unique name in the form (only alphanumeric characters, no whitespaces, and always starting with a character) and the first ***Key*** assigned to it.

You can then edit the rest of the attributes for this variable as required.

1. Double-click the ***ID*** variable field.

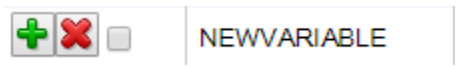
2. Type a unique name.

3. Select ***File > Save*** to save changes to the form.



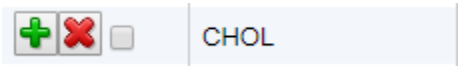
VARIABLE ID

New variables are added with the generic name NEWVARIABLE.



1. Double-click the ***ID*** variable field.

2. Type a new name for the variable.



Note
When you enter the unique identifier ***ID*** , note the following:

- No empty spaces are allowed.
- Allows only characters **A-Z**, 0-9 and _

DESCRIPTION

This field is used to add a brief description of the variable.

1. Double-click the ***Description*** variable field.

2. Type a description for the variable.

TYPE

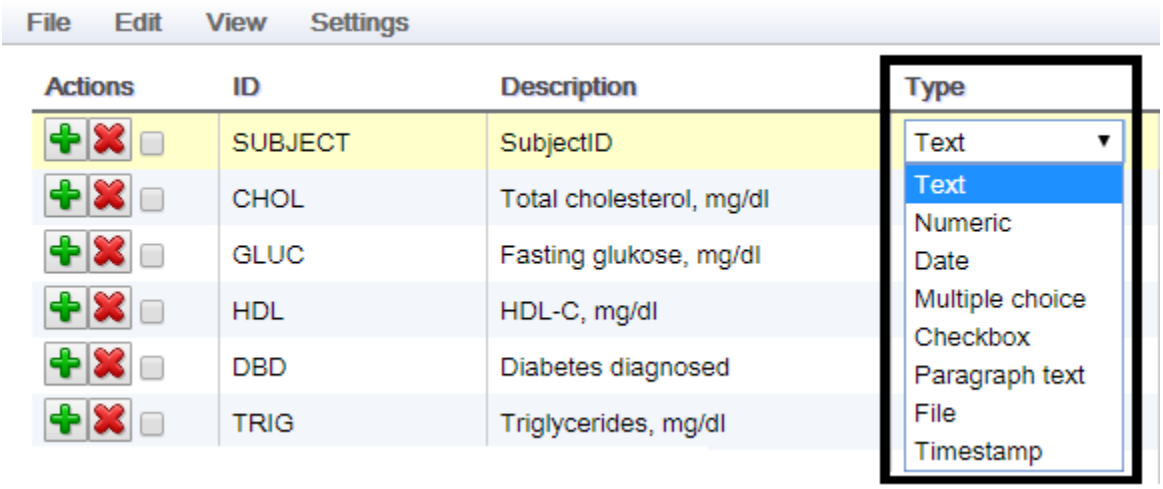
Type defines the kind of the variable.

The following variable types are available:

- Text
- Numeric
- Date
- Multiple Choice
- Checkbox
- Paragraph text
- File
- Timestamp

1. Select the arrow in the ***Type*** variable field.

2. Choose the variable type from the drop-down menu.



KEYS

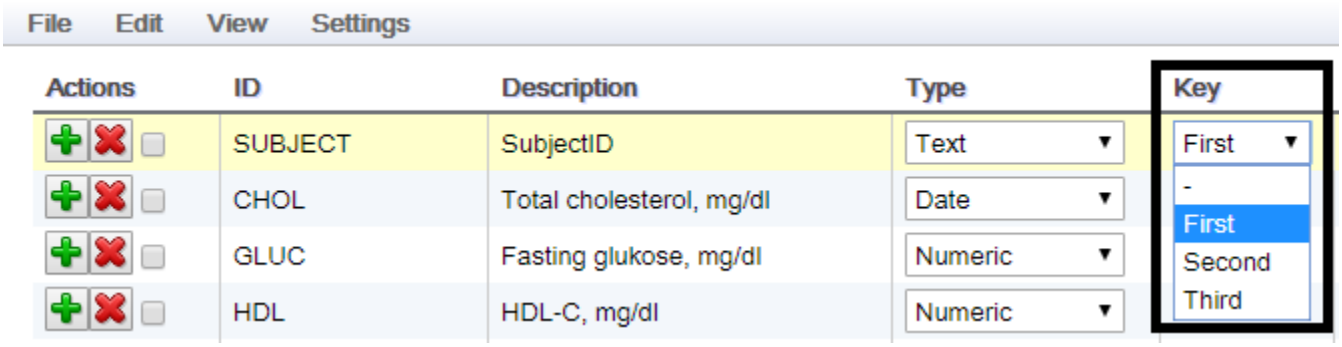
Keys define unique records in a dataset and also serve as a linking identifier across different data sections. By default, the primary key is the SUBJECT ***ID*** . The second and third key can be variable ***Type*** :

- text
- date
- timestamp
- multiple choice
- checkbox variable

1. Select the arrow in the ***Key*** variable field.

2. Choose the key from the drop-down menu.

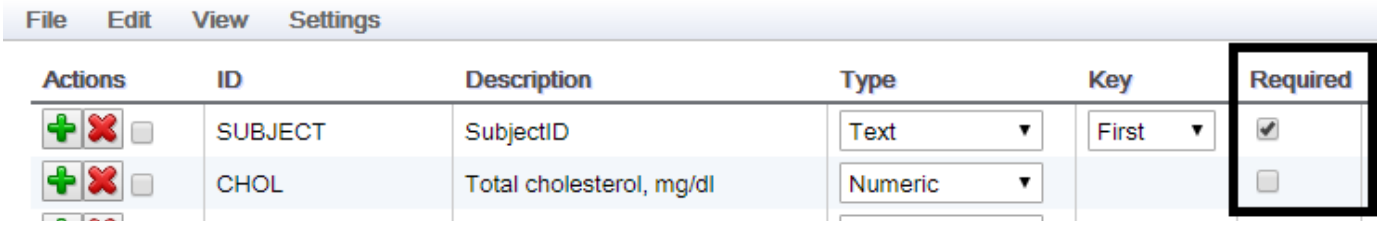
The following variable keys are available: First (primary key), Second (secondary key) and Third (tertiary key).



REQUIRED

This setting defines if the user must fill-in data for the corresponding variable.

1. Select the checkbox if data entry for the variable is mandatory.

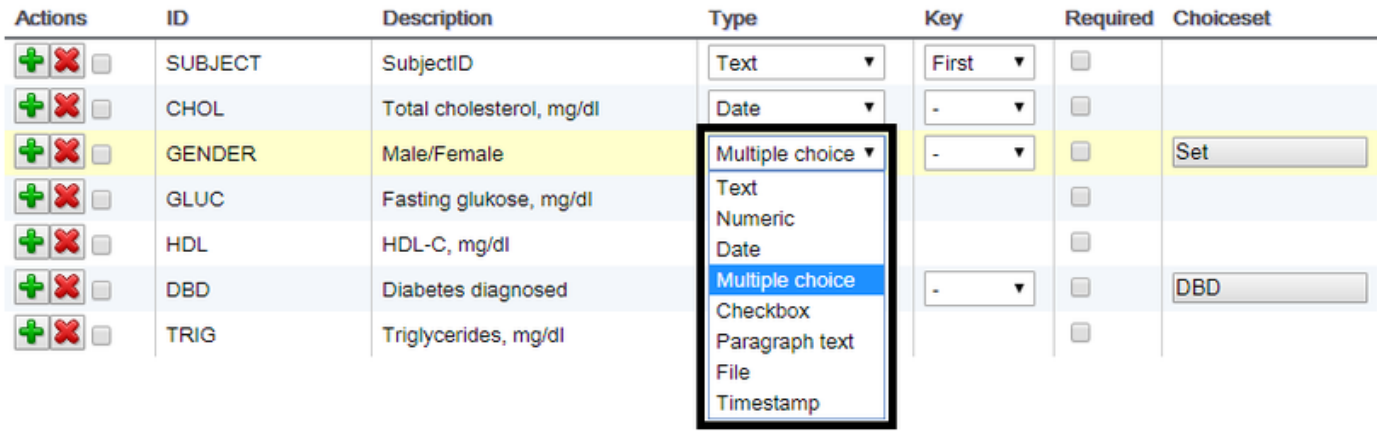


CHOICESET

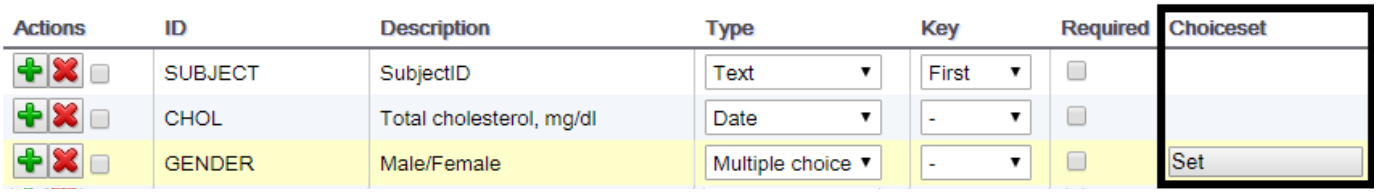
If a variable was defined as type Multiple choice, you must provide the list of items (= set) the user can select from in the column ***Choiceset*** .

The following variable keys are available: First (primary key), Second (secondary key) and Third (tertiary key).

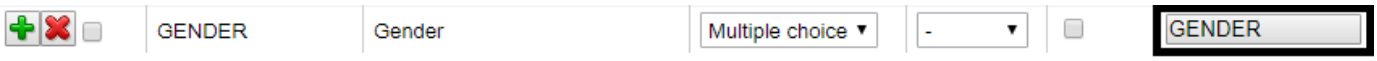
1. Select ***Multiple choice*** from the drop-down list in the ***Type*** column.



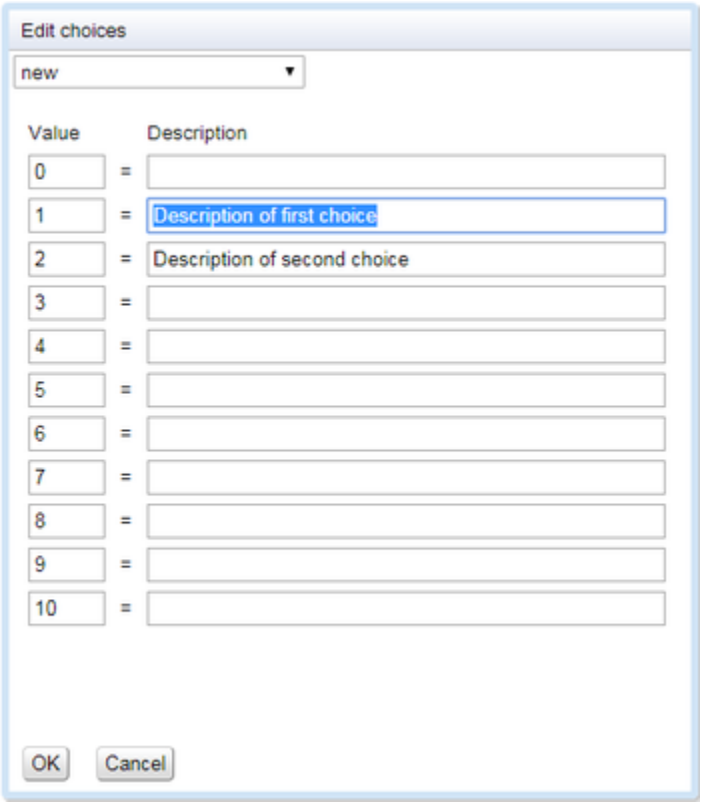
A selectable field becomes available in the ***Choiceset*** column.



2. Double-click the ***Choiceset*** field.



3. The ***Edit choices*** window opens.

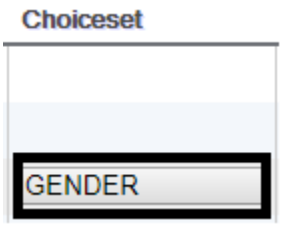


4. Type the choices you want for the new choice set.

Note
If biological sex information is needed in your statistical analyses specify ID of *SEX* in the form.

5. Select ***OK*** .

When you select ***OK*** , the system automatically renames the Choiceset based on the variable ID.

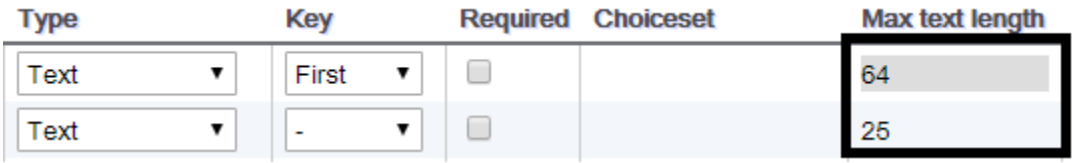


MAX TEXT LENGTH

This setting defines the maximum length of a text variable.

1. Double-click the ***Max text length*** field of the desired text variable.

2. Type the desired maximum text length.



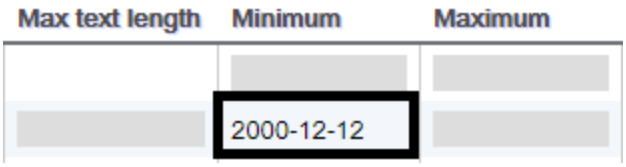
MINIMUM AND MAXIMUM

This setting defines the minimum value for a ***Type*** variable. The option is only available for variable types that have a minimum and maximum value, for example, a date or numbers.

1. Select the variable ***Type*** , for example ***Date*** or ***Numeric*** .

2. Double-click the ***Minimum*** field of the variable.

3. Type the desired value.



VALUE FUNCTION

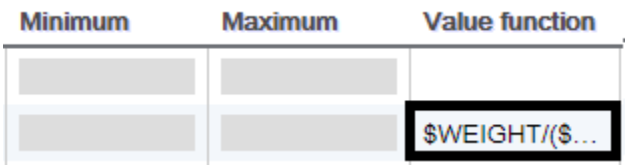
Value function allows the user to enter formulas, for example, to calculate the body mass index (BMI). Note that this setting is only available when you are viewing or editing single data entry forms.

> **Note**
> The ***Value function*** setting is only available when you are viewing or editing single data entry forms. If the field background is greyed out, you cannot edit the maximum value.

1. Select the variable ***Type*** , for example ***Text*** , ***Date*** or ***Numeric*** .

2. Double-click the ***Value function*** field of the variable.

3. Type the desired formula, for example, ***$WEIGHT/($HEIGHT*$HEIGHT*** , to calculate the BMI.



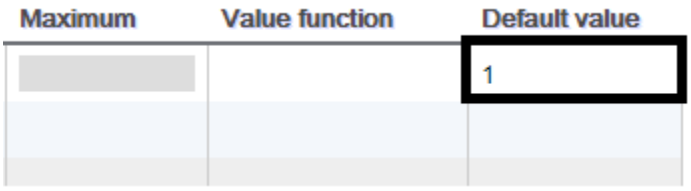
DEFAULT VALUE

This setting defines a default value for a variable. The default value is automatically displayed for that variable when the user fills in the form, and can be changed. Note that this setting is only available when you are viewing or editing single data entry forms.

> **Note**
> The ***Default value*** setting is only available when you are viewing or editing single data entry forms.

1. Double-click the ***Default value*** field of the variable.

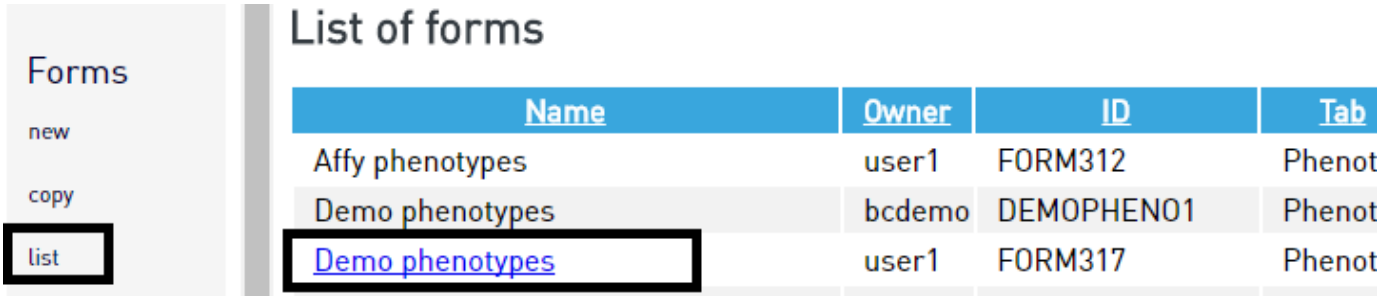
2. Type the desired default value.



Create webform from scratch

Navigate to the **DATA MANAGEMENT DATA STRUCTURES WEB FORM EDITOR** and select ***Forms > new*** .

1. Type a name for your form
2. Select the ***Form family*** from the drop-down list
3. Select ***Create*** . Your new form displays in the *List of forms*.

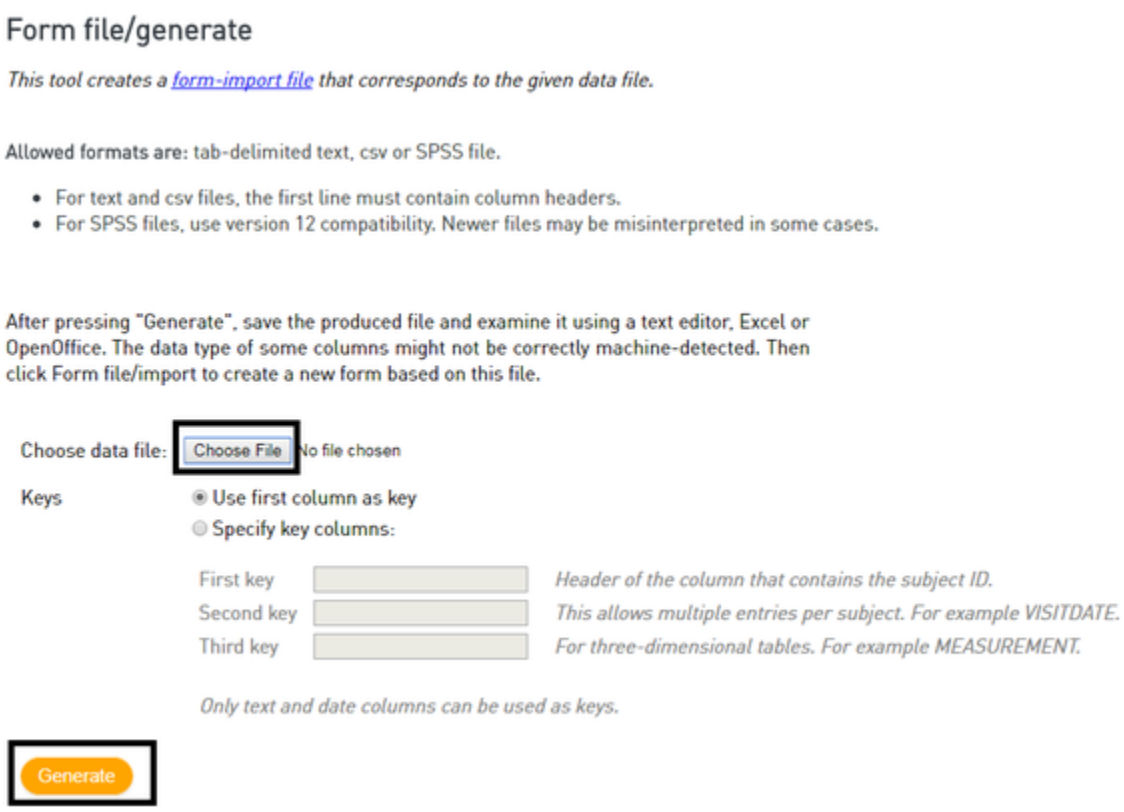


Import webform from a file

You can create a phenotype form automatically, whose structure matches the data in an existing import file. The import file contains variables and data values, as in the example *BMI_study_phenotypes.txt*. You can use this example file to test this feature out. We recommend you open the file in excel or text editor to see the structure and composition of the example data to get better understanding of how form generation works.

To generate a form by using a data file, navigate to the **DATA MANAGEMENT DATA STRUCTURES WEB FORM EDITOR** and ***Form file > generate*** .

Select ***Choose File*** and browse to the *BMI_study_phenotypes.txt* file.



Select ***Generate*** .

> **Note**
> The file that is generated is named automatically with ***_form*** in the title, and is most likely saved to the \ *Downloads* folder or to your Desktop.

Save the file (for example, *BMI_study_phenotypes* ***_form*** *.txt)* to your desired location. Open the generated file in Excel or text editor, and compare the variable names and types with the data in *BMI_study_phenotypes.txt*.

Create a new form using the form import file you just generated:

Select ***Form file > import*** .

1. Type a new name for your form, for example, *BMI study phenotypes*

2. Select ***Choose File*** and browse to the form import file you generated, for example, *BMI_study_phenotypes* **_form** *.txt*
3. Select ***Create*** .

Your new form displays in ***Forms > list*** . Select your new form to open it. You can now make changes to the form and publish it.

Publish a form

Steps to Publish the form to a dataset:

1. Click the Publish button in the in ***Forms > List*** of ***Web form editor***

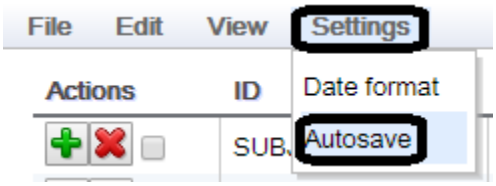
2. Click the title “Click here to open it” when you see the confirmation of the created dataset

Form settings

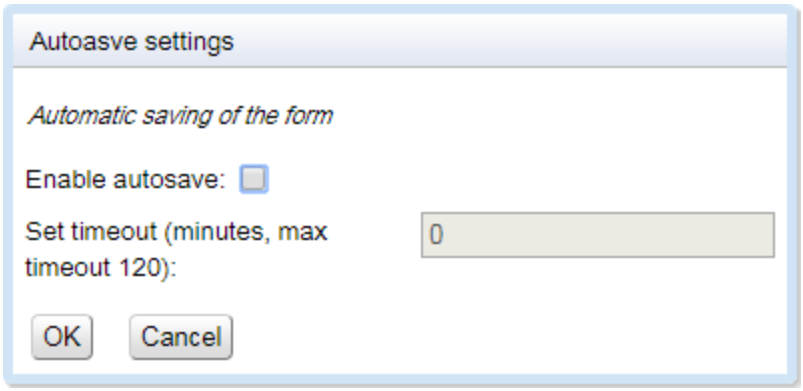
Autosave

To change the automatic save settings:

1. Go to ***Settings > Autosave*** to edit the autosave settings.

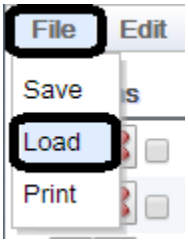


2. Select ***Enable autosave*** and set the timeout (in minutes).



3. Select ***OK.***

If you need to restore the previous save, select ***File > Load***.



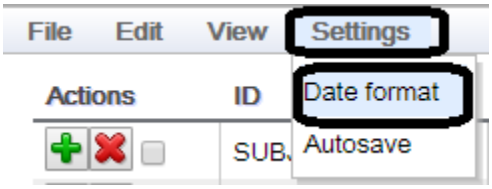
4. Choose the file to load based on the timestamp information and select ***Load***.

5. Select ***OK*** to confirm.

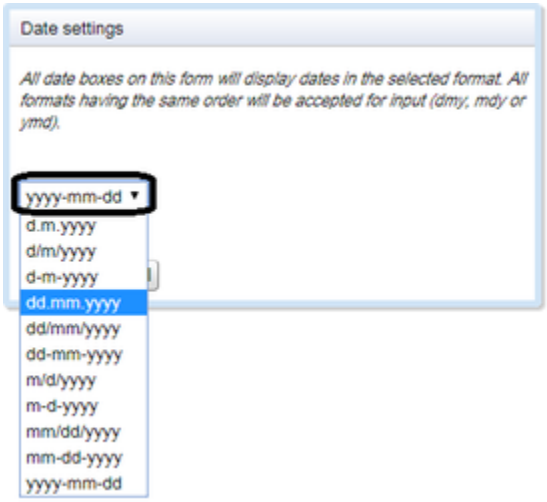
Date format

To change the date format settings:

1. Go to ***Settings > Date*** format.



2. Select the date format you want to use from the drop-down menu.



3. Select ***OK***.

BC|INSIGHT - 3.3 Data ontologies and terminologies

User roles
BC|INSIGHT user

Child pages:

- BC|INSIGHT - 3.3.1 Inbuilt ontologies
- BC|INSIGHT - 3.3.2 Use of ontologies in forms
- BC|INSIGHT - 3.3.3 Tools that use ontologies
- BC|INSIGHT - 3.4.4 Data vocabularies

Describing the data

It is often the case that simply creating data structures that are able to accommodate text, numbers, dates and so on, is not enough to describe the data, and to understand the relationships between different variables. A research project may be able to simply agree upon a way how things are described, and stick with those rules, especially if the design of the data model is strictly enforced in the research. However, when the project grows, or new projects start, old rules may need to change, and some will be forgotten.

A commonly used solution is to apply a data vocabulary - or dictionary - on top of the actual data structure, and describe the content that way. A typical example could be the patient identifier. In BC the convention is to use SUBJECT as the name of the variable that denotes patients ID. However, it is only a convention, and thus can be replaced with something else. Also when the project receives data from outside sources, there may be different kind of variable naming in use. The alternative field names can quickly become confusing, and when one needs to view the data for the same patient across differently built data tables, these changes may become challenging.

In BC|INSIGHT the SUBJECT field comes with an annotation "BC:subject". This is an inbuilt attribute to a field that has a meaning: "This is patient ID". If users wish, or need, to use different naming for the patient ID variables, application of the annotation "BC:subject" will tell the BC|INSIGHT as a system, the the variables mean patient ID. This helps BC|INSIGHT to find tables with patient data, and allows it give automated suggestions for content and datasets in various tools working on patient data.

The same applies to almost everything in the data model in use. Genomic marker names in BC|INSIGHT are annotated "BC:marker", and chromosomes "BC:chromosome", and so on. It is possible for users to have their own designed vocabulary in BC|INSIGHT, to describe their data model and the data. BC|INSIGHT has an increasing number of tools that are able to utilise this information.

Ontology - annotation

You will see throughout this document the words ontology and column annotation being used in mixed manner. It may seem confusing. Ontology means the data vocabulary describing the data, and the individual items in an ontology are called terms. However, the documentation often refers to a term that is applied to a column - like "BC:subject" - as the ontology for that column. Annotation of a column means the process of applying vocabulary to the columns. Hence the seemingly mixed use of these terms, that essentially mean the same thing in the context of columns.

Do not mix column annotation with genomic or omics data annotation, which is again a different thing.

BC|INSIGHT - 3.3.1 Inbuilt ontologies

User roles
BC|INSIGHT user

Table of contents:

- Multiple fields combined for one function

BC operating system provides ontologies for inbuilt data structures available for data management. At the moment BC systems use two different ontologies, named "BC_VARCLASS", and "BC". Of these two "BC_VARCLASS" is partially deprecated and is being replaced by the "BC" ontology. Typical examples of forms annotated still with BC_VARCLASS ontologies are for example "ACGT coded SNPs" (form ID PLAINSNPS).

Variable details

Id	Key	Description	Type	Choices	Annotations	Required	Max Length	BC D
SUBJECT	1	SubjectID	Text	-	BC_VARCLASS:patient	yes	16	
MARKER	2	MarkerID	Text	-	BC_VARCLASS:marker	yes	16	
ALLELE1		Allele1	Text	-	BC_VARCLASS:snp_allele1	no	1	5
ALLELE2		Allele2	Text	-	BC_VARCLASS:snp_allele2	no	1	5
MENDEL		Mendel error	Choice	0 = no, 1 = yes, 2 = unsure	-	no		

Image 1: ACGT coded SNPs -form annotated with BC_VARCLASS ontology for patient, marker, and alleles.

All new data structures are annotated using the more recent and more structured BC ontology, like VCF Composite datasets.

Variable details

Id	Key	Description	Type	Choices	Annotations	Required
MARKER	2	Marker ID	Text	-	BC:marker,BC_VARCLASS:marker	yes
SUBJECT	1	Subject ID	Text	-	BC:subject,BC_VARCLASS:patient	yes
AINDEX1		Allele index 1	Integer	-	BC:allele_index	no
AINDEX2		Allele index 2	Integer	-	BC:allele_index	no
IS_PHASED		Is genotype phased?	Choice	0 = No, 1 = Yes	BC:dt_alternat	no
PLOIDY		Genotype ploidy	Integer	-	BC:dt_integer	no
ALT_DOSE_INT		Combined dose of ALT alleles (integer)	Integer	-	BC:dt_integer	no

Image 2: VCF Composite genotypes form annotated with BC ontology in addition to BC_VARCLASS.

"BC_VARCLASS" vs "BC" ontologies are different. "BC_VARCLASS" provides pure classification for important data fields that are used internally by the BC operating system. "BC" -ontology on the other hand provides classification (like patient and marker), but also typing for data, like in the above example BC:dt_alternat, which tells the tools in the system that the field must contain enumerated choices (i.e. 1:male, 2:female, 3:NA). The latter feature is used to distinguish between fields that are similar on data model (SQL) level but provide different functionalities (alternative choices coded as integers, vs. actual integer numbers like 'ploidy').

Multiple fields combined for one function

The ontology design allows application of ontologies in such way that they define a function that requires multiple fields. The combination of multiple fields then becomes so called virtual column, accessible by tools as one logical item. A good example of a virtual column is the genomic range annotation, which is defined by three fields: chromosome, start location, and end location.

Variable details

Id	Key	Description	Type	Choices	Annotations	Required	Max Le
REGION	1	Genomic region covered by the reference allele	Text	-	BC_VARCLASS:region	yes	250
CHROM		Chromosome	Text	-	BC:chromosome	no	64
POS		Position	Integer	-	BC:bp_position	no	
REFSIZE		Reference allele length	Integer	-	-	no	
REF_ALLELE		Reference allele	Text	-	BC:ref_allele	no	250
ALT_ALLELE		Alternative allele	Text	-	BC:allele	no	250

Image 3: Virtual field 'range', which is used to tell the BC system that this annotation form is capable of providing genomic range data to tools. Note that the form has both BC_VARCLASS, and the more modern BC:bc_position annotations for relevant fields.

BC|INSIGHT - 3.3.2 Use of ontologies in forms

User roles

BC|INSIGHT user

Data manager

Table of contents:

- Ontologies and annotating form variables
  - Editing ontology terms

Ontologies and annotating form variables

BC systems operate on SQL and other data structures, which store different types of data. The structures are described in BC systems as table 'forms' or table 'templates'. A form describes the fields in a table structure, the data type in each field, and some additional constraints about how the field should behave, like maximum length in text fields, options in alternative choices, etc. The form also tells the SQL or other data structures, which field are to be used as key fields for indexing the data.

Many forms are similar in structure, and user will often see things like 'SUBJECT' or 'MARKER' being used across multiple forms to denote patient and genetic SNP marker, respectively. This naming convention is used to make the structures easier to understand with quick glance. For example, seeing a dataset with the field 'SUBJECT' in it tells the user that the table contains patient data, and that 'SUBJECT' is likely to be a key field of type text.

However, it is perfectly possible for users to define their own forms, where the field that means patient key, might not be called 'SUBJECT'. The BC operating system should regardless be able to recognise the field as equivalent to SUBJECT field, and adjust the behaviour of data management and analytical tools accordingly. The system also defines inbuilt forms containing fields with less clear meanings, and the user may want to override these in her own forms, without losing the functionality of specialised tools that take use of those fields.

BC internal ontologies are used to give any field or group of fields a functional or conceptual meaning. Ontology is used to create a vocabulary inside the BC system that describes the structure of the data - beyond the mere field names and descriptions - in such way that the underlying operating system, and analytical and logical tools can use the information in meaningful way. See Image 1 for an example of usage of BC internal ontology terms.

Variable details

Id	Key	Description	Type	Required	Max Length	Annotations
SUBJECT	1	SubjectID	Text	yes	64	BC_VARCLASS:patient, BC:subject
SAMPLE	2	Sample	Text	yes	64	BC_VARCLASS:sample, BC:sample
PROBE	3	Probe	Text	yes	64	
MZ		Retention time	Float	no	8	
INTENSITY		Intensity	Float	no	8	

Image 1. Annotation of inbuilt form, seeing the structure in the dataset's STRUCTURE -page. The tools in 'Annotations' -field can be used to edit the ontology terms.

For example, when joining patient data tables together, it is useful if the system already understands the concept of 'patient', and can suggest joins automatically using the fields that hold patient index in both datasets. When creating annotations for genomic features like SNPs, it is useful if the system automatically will recognise a field as a genetic SNP marker, understands how a gene range is formed from chromosome and start and stop positions, and so on. These are functional examples of the use of internal ontology to help built meaningful tools.

Editing ontology terms

> **Note**
> Editing data structure requires 'Data manager' user role.

It is sometimes necessary to edit the ontology terms of existing datasets, to add meaning to fields, to make them visible in searchers or usable in ontology-aware BC|INSIGHT tools. Sometimes during data model design phase it is helpful to be able to tweak the annotations of already existing structures to see their effects in use. You can apply ontologies either directly to the forms in Structure Editor, or you can ue the dataset's STRUCTURE -page to do so (Image 2).

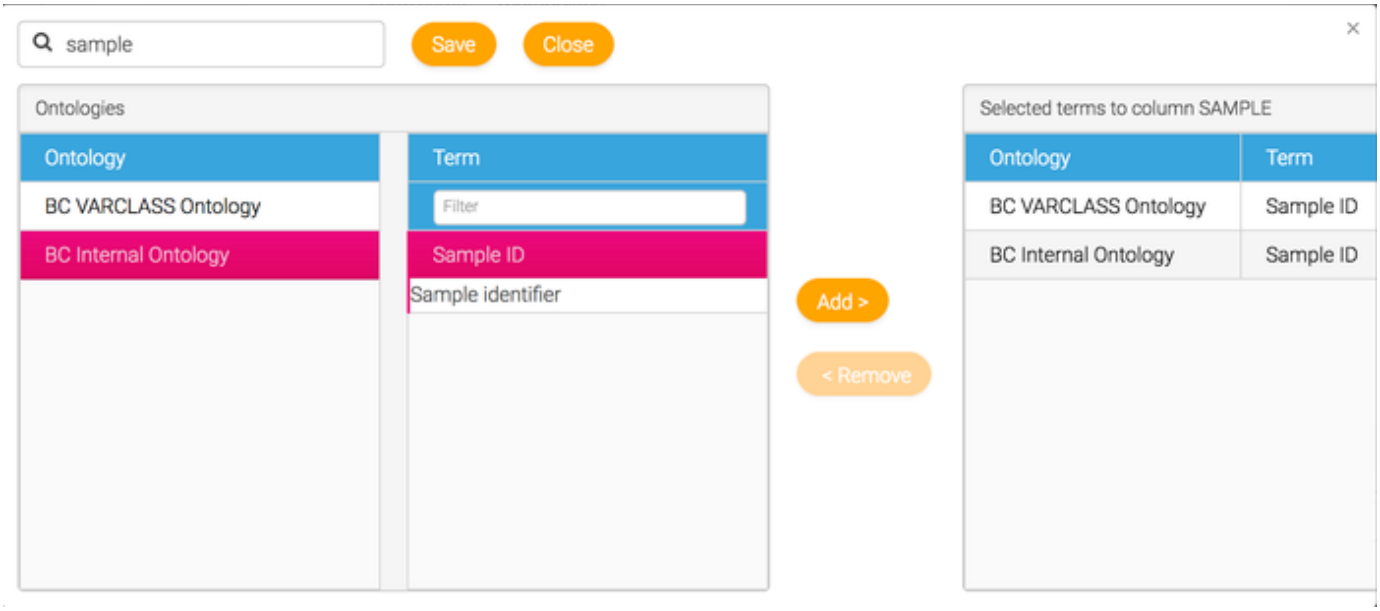


Image 2. The ontology or annotation -tool displays available ontology structures, and the user is able to search and apply the chosen terms to the table form.

BC|INSIGHT - 3.3.3 Tools that use ontologies

The list of tools that automatically utilize the internal BC ontologies is constantly growing. Some notable ones available in the current production version of BC|INSIGHT are:

- Dataset search: Search tool in the dataset navigator indexes the datasets based on metadata and field ontologies
- Analytical tools: Analysis interface automatically collects and joins data as input for analysis algorithms based on patient, marker, etc ontologies
- Subset tool: Matching datasets for joins using BC:subject
- Annotation tool: Uses BC:genomic_range and BC:genomic_pos to annotate genetic marker lists with corresponding gene annotations

BC|INSIGHT - 3.4.4 Data vocabularies

Research data vocabulary

Beyond functional features in the system, it is possible for the research project to define their own internal data vocabulary that describes the data in the system. The BC application programming interfaces (APIs) provide means for searching and recognising table fields based on their annotated ontology, some search features can utilize the information automatically, and many future tools will allow the user to use the data vocabulary for data management and visualisation.

Data vocabulary describes the research meaning of the data. There are many existing ontologies that could be used for this purpose, in the areas of clinical research, human genetics and genomics, health data, and so on. These ontologies (see examples for Gene Ontology Consortium, Hum an Phenotype Ontology, Ontology of Clinical Research) are useful on their own, and harmonise the content in the BC database to a more widely used standard. However, from data management point of view, ready-ontologies can appear stiff and restricting. If this is the case, providing project-specific vocabulary is called for.

Data vocabulary development should start from the system usability point of view. Data administrators should ask themselves:

1. What are the typical data questions being asked by the platform users
2. Which data items are involved in the answers
3. What kind of accelerating or aiding structures the database could host to help discovery

Based on the assessment of use-cases for data discovery, the vocabulary can be applied to commonly used fields, units, measurements, processes, observations, methods, etc. These items should form the basis for searchers, automation and workflows, and reporting in the system.

BC|INSIGHT - 3.4 Creating a dataset

User roles

Data manager

Child pages:

- BC|INSIGHT - 3.4.1 Granting permissions to a dataset

You can create a new dataset based on a pre-existing form if there is no suitable dataset available.

To create a new dataset:

1. Select ***New dataset*** from the data navigation page
2. In the ***Create new dataset*** dialog, type a unique name for your new dataset in the ***Dataset Name*** field.
3. Use ***Select Folder*** if you want to show the dataset somewhere else than in the folder where you are now.
4. In ***Species***, select the species you want from the drop-down menu. This is used as meta data in genomic analyses.
5. In ***Select form,*** search for the form you want to use for your dataset.
    a. You can view the variable details of a form on the right side of the dialog
6. Click ***Create dataset*** to create your new dataset.

Your new dataset is visible in the navigation pane in the main page.

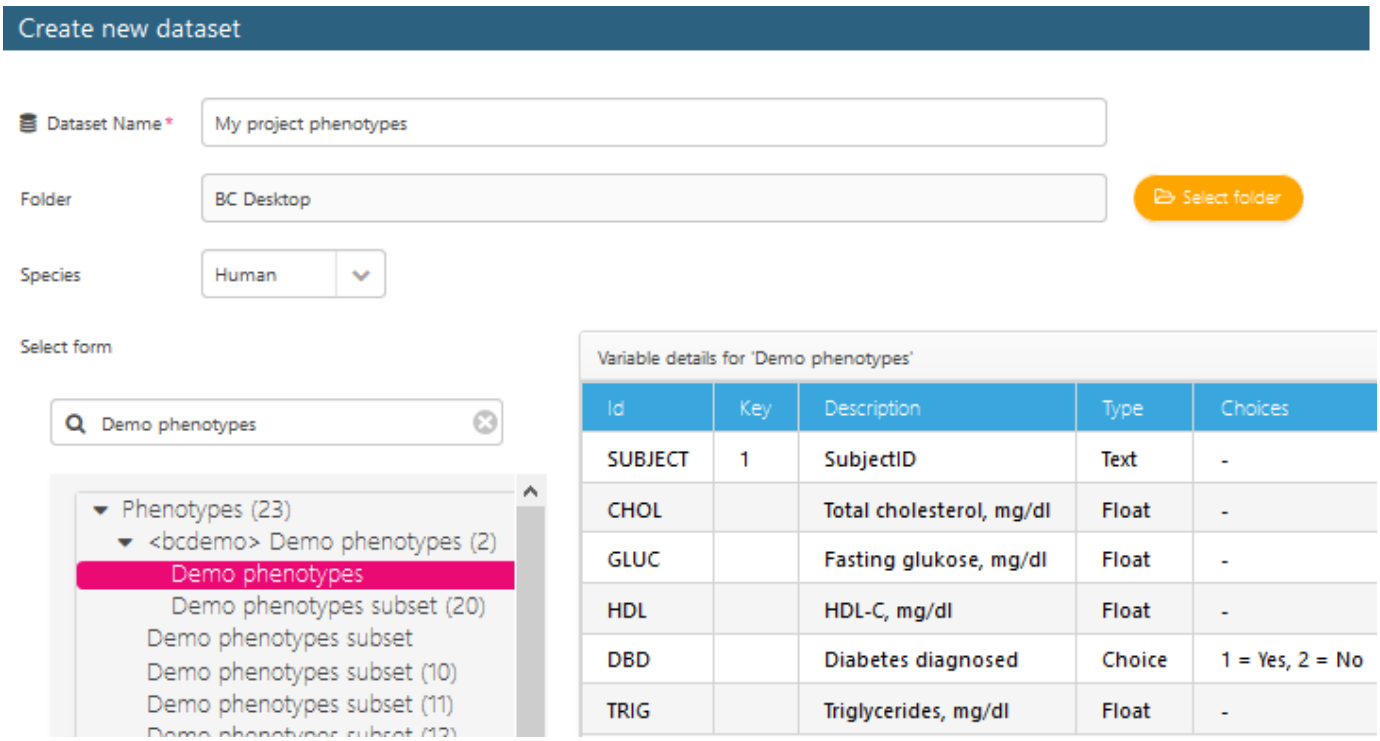


Image 1. The New dataset -dialog gives you options to choose the form and the location for the dataset. Remember to give the dataset a unique name.

BC|INSIGHT - 3.4.1 Granting permissions to a dataset

User roles
Data manager

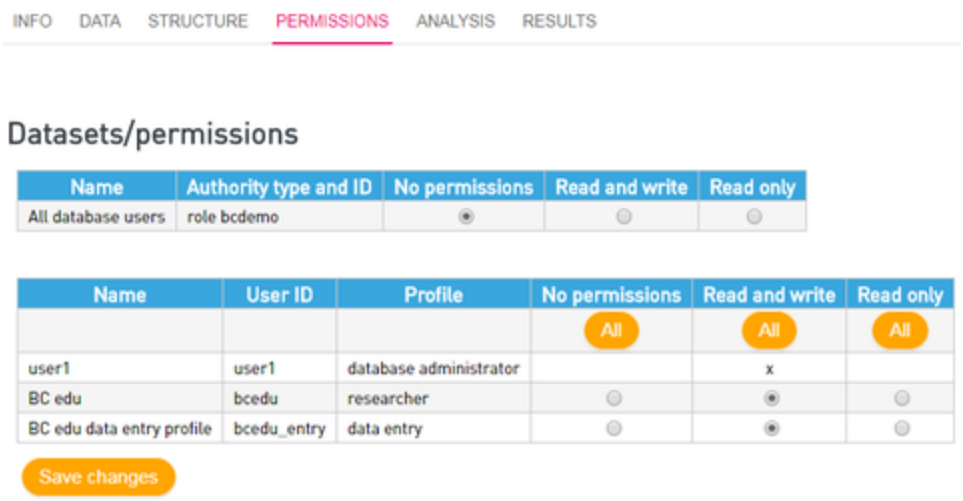
Permissions basics

By default, only the user who has created a dataset can see and use it, but the user can give read and write permission to other BC|INSIGHT users using the Permissions -tool. Reports in the result archive can also be shared with other users.

1. Select the ***PERMISSIONS*** tab to open the PERMISSIONS page.

2. Set the permissions for the users.

3. Select ***Save changes***.



BC|INSIGHT - 3.5 Creating subsets

User roles
Analyst

Child pages:

- BC|INSIGHT - 3.5.1 Simple subsets using filtering options in the data grid
- BC|INSIGHT - 3.5.2 Using Subset tool
- BC|INSIGHT - 3.5.3 Joining dataset information in a subset
- BC|INSIGHT - 3.5.4 Advanced subsets
- BC|INSIGHT - 3.5.5 Gene range annotations

Subsets are used if you want to keep a set of criteria for filtered information. You can:

- Create simple subsets using filtering options in the data grid
- Create subsets using DATA MANAGEMENT > SUBSET
- Join information between tables

BC|INSIGHT - 3.5.1 Simple subsets using filtering options in the data grid

User roles
Analyst

Table of contents:

- Filter data in the DATA page
- Row count button

- Column button
- Refresh menu

The data grid Subset feature

The ***DATA*** tab provides tools to filter data stored in a dataset. It uses the *AND* option between the columns. The ***Subset*** button appears when you start filtering information in a column's filter field, or if you hide columns.

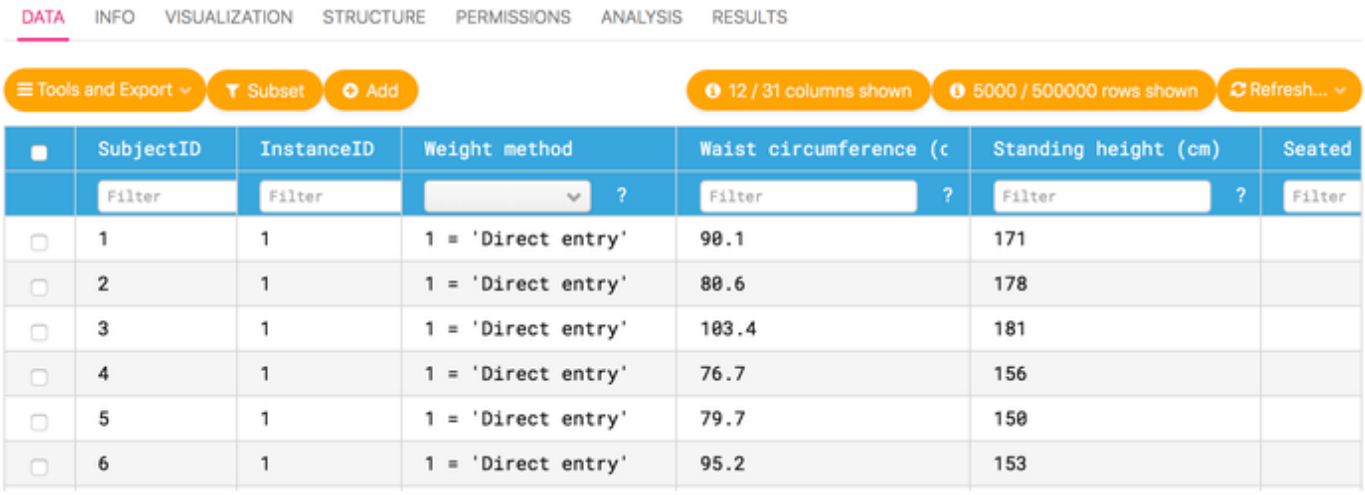


Image 1. Filtering data directly in the data grid activates the Subset function. Hiding columns will also activate the tool.

Select ***Subset*** to create a subset based on the filtering criteria. This opens the ***Creating a new subset*** dialog, where you can type a descriptive name (max 250 characters) for your subset. After creation, the new subset appears in the data navigator as a child to the original table.

Filter data in the DATA page

User can filter the data in the Data Grid by typing filter values in the Filter text box, and hitting ENTER button. Multiple filters can be combined. To clear all filters in the Grid, the 'Refresh' button drop-down can be used. Syntax for filtering values depends on type of data in the field.

- In any data type a single data value can be used.
  - Search is case-sensitive
  - Search matches text values anywhere within the text
  - Explicit wildcard characters are not supported (i.e. '*', '?')
- Numeric values can be filtered using
  - range filters with dash '-', ex. '100-200'
  - smaller and equal values with '<' and <=', ex. '<=100'
  - larger and equal values with '>' and '>=', ex. '>=100'
- Choice variables from choice menu
- Date / timestamp variable can be search by the whole value or year
  - ISO format required 'yyyy-MM-dd', ex. '2017-06-30', or '2017', or '2017-06' respectively for year and year-month combination
  - Range and comparison operators cannot be used.

Row count button

The row count button ('All xxx rows shown') can be used to adjust the maximum number of rows (5000 by default) shown when dealing with large datasets. If the dataset is a Variation dataset, it may require index refresh in order to display true total number of rows. You can refresh the index for Variation data on the INFO page.

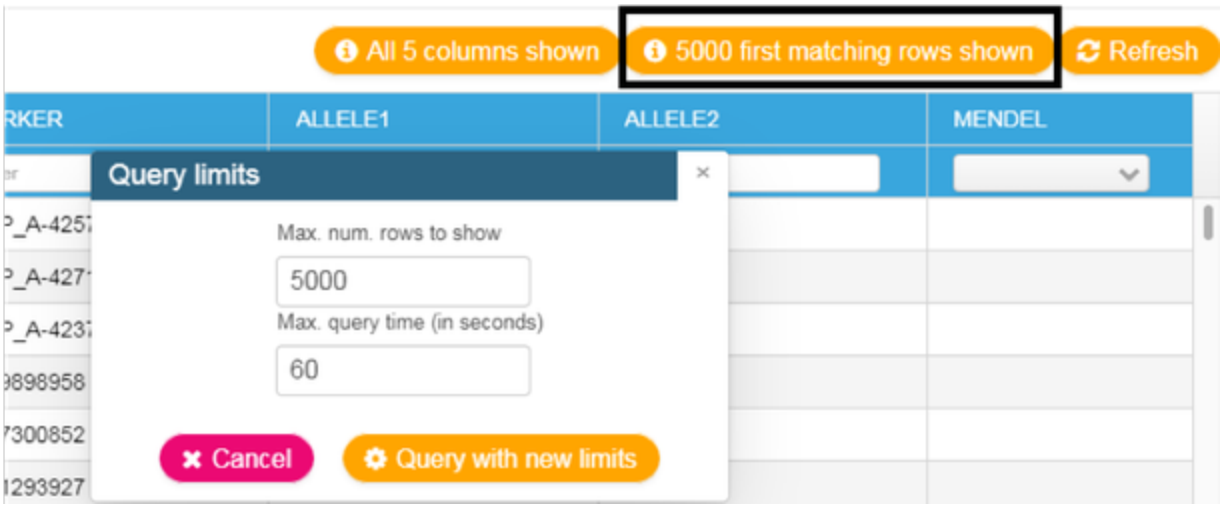


Image 2. Setting the number of rows, and the maximum SQL query time allowed to filter data.

Column button

The column button ('X / Y columns shown') is used to control visibility of fields in the DATA view. The dialog that opens displays a list of available fields, from which user can select the ones they wish to have visible. By default the first 20 columns are displayed. If any fields are hidden, the 'Subset' button will be visible, provided the user has permission to create subsets in BC|INSIGHT. Fields that belong to the datasets primary key cannot be hidden from view. During data export the hidden fields can be left out from the extracted file.

> **Note**
> If the column list is long, you must scroll down to find the ***Select columns*** button.

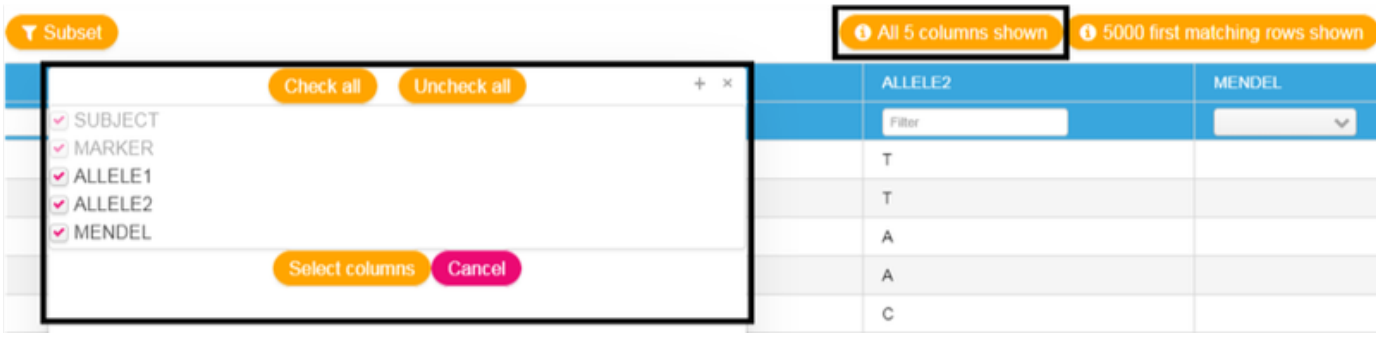


Image 3. Selecting the columns that are displayed.

Refresh menu

Refresh button is also a drop-down menu, where user can clear all filters from the grid. Refresh will update the dataset content, and is commonly used when changes to data are to be expected, for example after data upload, or adding new rows. Clearing filters will automatically also refresh the grid view.

BC|INSIGHT - 3.5.2 Using Subset tool

User roles
Analyst

Table of contents:

- Define a subset
  - Subject and row counts
  - Filtering

- Value distribution
- Preview
- Create

In BC|INSIGHT, it is possible to filter information from one dataset and to join information from several datasets using the ***SUBSET*** tool in ***DATA MANAGEMENT*** .

All available datasets are listed in the hierarchical dataset tree on the left side of the screen. Datasets are dragged and dropped to the canvas on the right side. This is the primary view of the selected datasets in the subset tool. In this view, datasets can be added from the dataset list and manipulated interactively in the subset.

If a subset has at least one dataset, dragging another one to the canvas calls an automatic join candidate generator. For more information about joining dataset information, see section 3.5.3 Joining dataset information in a subset.

For more information about how to use the subset tool, click on the help -icon in the application.

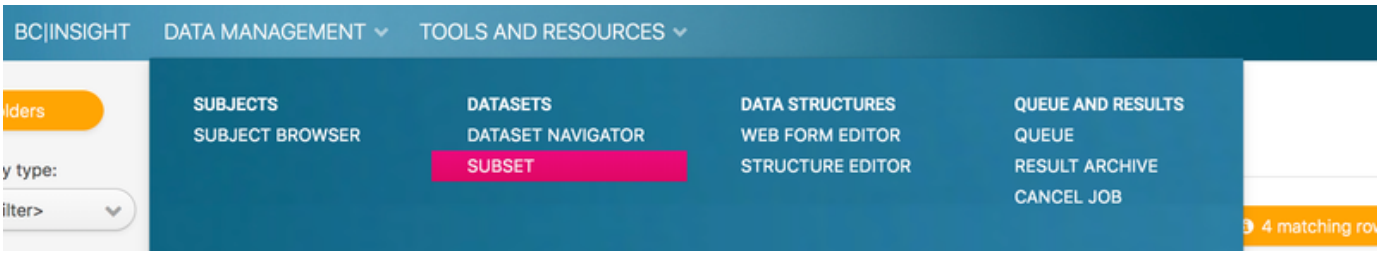


Figure 1. Finding the subset tool and help.

Define a subset

When you open the Subset -tool you will see an area where you can drag and drop datasets from the navigation tree. If you work with multiple datasets, you are able to manipulate the linking of those datasets, the selection of columns, filtering by data values, and also add new custom fields using SQL functions.

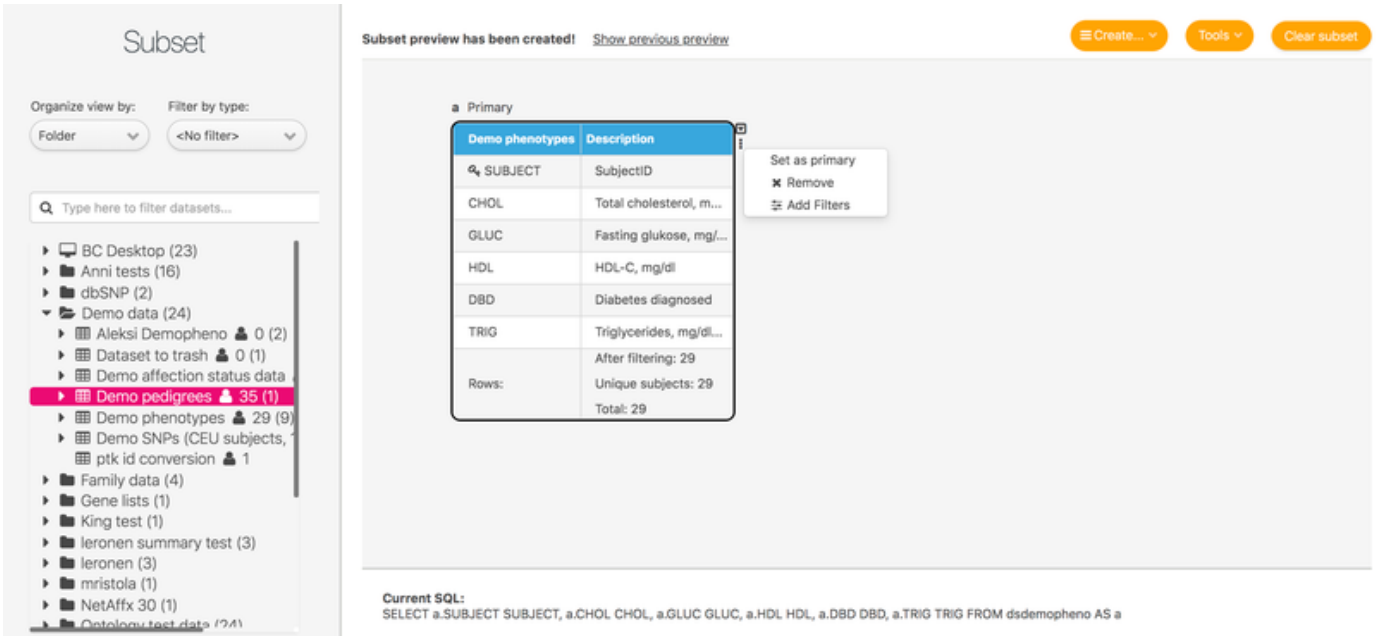


Image 2. The Subset application, and the working area. The dataset menu is open.

You start your subset by dragging and dropping a dataset from the tree to the work area. The dataset information is displayed as a table. Click on the vertical ellipsis on the top right of the table to see the table menu, which gives you Add Filters -option. By selecting that you get to a view where you can set the required data filters. If you want to see the effect of your filters, select a column to display a graph showing the data available after filtering has taken place.

In the subset workspace, you can at any time reset the workspace to empty by clicking the top right button **Clear subset**.

Subject and row counts

The subset workspace shows the currently available total rows in the dataset, applying any filtering. It also shows the number of unique Subjects in the dataset, if the dataset contains Subject identifiers, annotated with BC:subject column type in the dataset form. This is typically the case in all phenotype tables. If the workspace contains joined datasets, the row and subject counts still only apply to the single datasets. IN order to see the result of the join, user needs to generate a Preview and open it.

Filtering

To specify the filtering criteria from Add filters in the ellipsis drop-down menu:

1. Select a variable row to edit the criteria.
2. In the ***Condition*** column, select the condition for the variable.
3. Depending on the condition type selected, in the ***Value*** column, type values to variable text field or choose a value from the value drop-down menu.
4. Repeat the previous steps for all variable rows whose criteria you want to edit.

> **Note**
> Variables can also be removed from the subset by unselecting the check box in the ***Check*** column. By default all variable rows are selected.

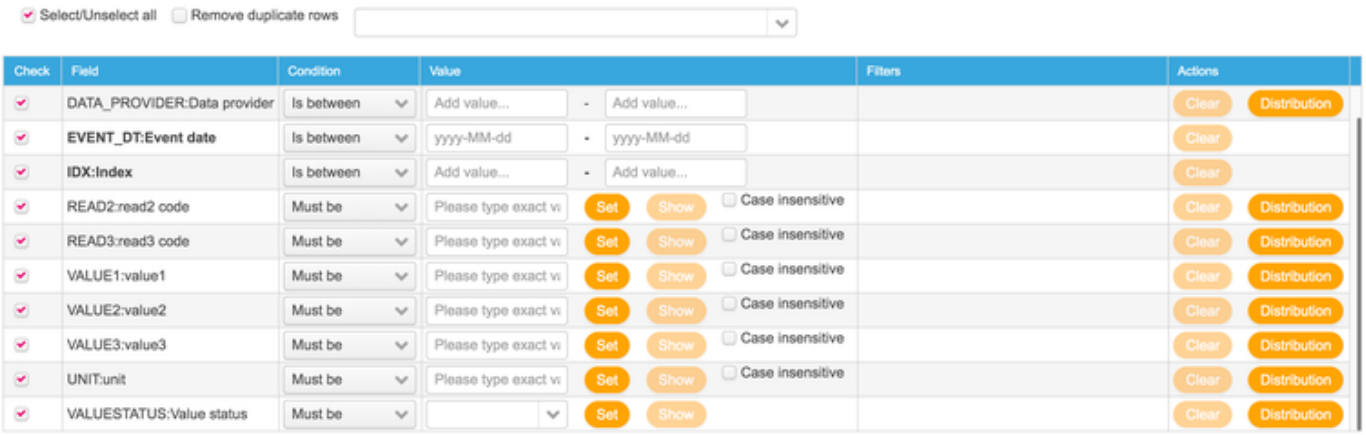


Image 3. Filtering data based on values in different columns, and selecting columns for your subset.

You can clear the filters with **Clear** button, this will reset the field to it's original state.

If the filters include TEXT type values, the default is to make a case-sensitive search. This is the fastest filter option in terms of SQL filter performance. It is possible to toggle any TEXT type field filter into case-insensitive, in which case the search relies on regular expression, and matches all variations in character case. The case-insensitive filter is not as fast to resolve as the case-sensitive filter.

Value distribution

In order to see the distribution of data values in a field, click the **Distribution** -button. This will display the statistical distribution for the field. This information can be used to define the filter ranges or lists. After a filter is set, the Distribution button will display updated graph based on filtered data.

Preview

To create a preview of your subset select **Create > Preview.** This gives you **Show preview** -link, which opens the content of the filtered dataset in a data grid. Click any non-key variable column to view the column's value distribution.

Create

Once you're happy with the content of your filtered dataset, you can create a permanent subset by selecting ***Create > Subset*** and giving your subset a name. You can also specify the folder where the subset is to appear. If you create the subset in the same folder where the original dataset is, your subset will appear in the navigator tree under the original dataset.

BC|INSIGHT - 3.5.3 Joining dataset information in a subset

User roles
Analyst

Table of contents:

- Creating joins
  - Joining non-genotypic data
  - Different logical join types

Creating joins

If a subset has at least one dataset, dragging another one to the canvas calls an automatic join candidate generator. When you join information from several datasets, the subset tool utilizes that join candidate generator, which suggests the joining variables based on the pre-set variable annotation information specified by BC support. The chosen option is shown in the canvas after you confirm your choice.

> **Note**
> You can join dataset to datasets, subsets to datasets, datasets to subsets, and subsets to subsets.
>
> The speed of joining reduces when you join subsets to datasets/subsets as the query uses the original dataset to retrieve the information.

Joining non-genotypic data

You start joining datasets together by selecting your starting dataset from the navigation tree in the Subset -tool, and dropping it into the work area. You then select the other dataset, which you'd like to join to the first one. Make sure that these two datasets have some common key or other field, through which they can be linked. This could be a subject or sample identifier, a genetic marker, gene name, or similar information. The point is, these two datasets must have something in common and share some data, in order to be linked together.

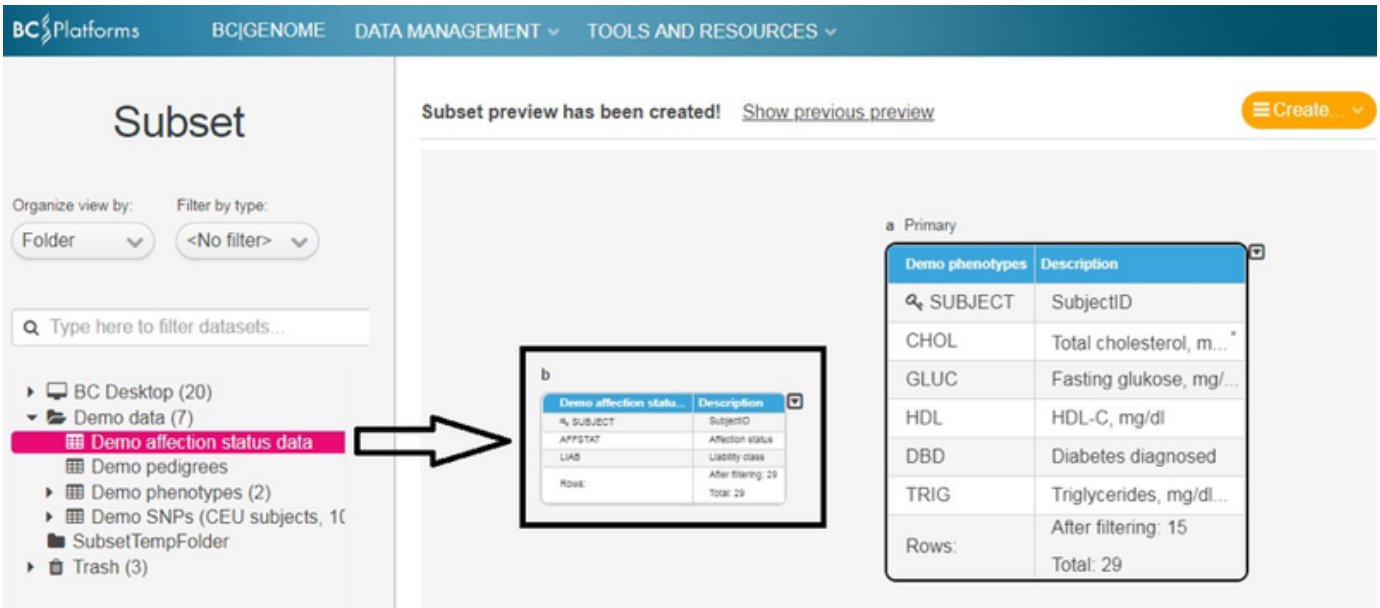


Image 1. In the example above, the first selection for the joined datasets is the Demo phenotypes, and the second in a table containing the affection status information for all subjects in the Demo phenotypes.

If the selected join candidates have fields that have been annotated in the same way, the Subset tool will automatically pick those fields and present them as candidates to be used to join the datasets together. If no such candidates are found, you can select the common fields by simply clicking first on the starting dataset's field, and then on the second dataset's equivalent field. This will give you a prompt dialog explaining how these two fields will be used to join the two datasets. You need to confirm this to continue.

> **Note**
> You can make several different joins between two datasets. Each separate join is highlighted with different colour.

By default the join is created as 'Inner join', see in the next section more ways to control how data is joined.

Different logical join types

If the subset contains at least one join, a join type can be selected. The number of different join types depends on the number of datasets in the subset as well as conditions of joins.

- **Inner Join:** Retrieves dataset values that fulfil the join condition in both datasets. Inner join is always available.
- **Full Outer Join:** Retrieves dataset values that fulfil the join condition in both datasets. In addition, Full Outer Join generates values for cases where the condition is not met, by setting values of the other table to null. Full Outer Join is available for subsets with two datasets only, without namespace mapping join condition.
- **Left Outer Join:** Retrieves dataset values that fulfil the join condition from the primary (left) dataset. In addition, Left Outer Join retrieve values from the primary dataset that do not meet the condition by setting values from the secondary (right) dataset to null. Left Outer Join is available for subsets with two datasets only, without namespace mapping join condition.
- **Right Outer Join:** Retrieves dataset values that fulfil the join condition from the secondary (right) dataset. In addition, Right Outer Join retrieves values from the secondary dataset that do not meet the condition by setting values from the primary (left) dataset to null. Right outer join is available for subsets with two datasets only, without namespace mapping join condition.

Subset tool also includes two unconventional join type options:

- **Semi Join:** Retrieves single dataset values from primary dataset which have at least one instance in the secondary dataset. Semi join is available for subsets with **2 datasets only**.
- **Anti Join:** Retrieves single dataset values from primary dataset which have no matching instances in the secondary dataset. Anti join is available for subsets with **2 datasets only**.

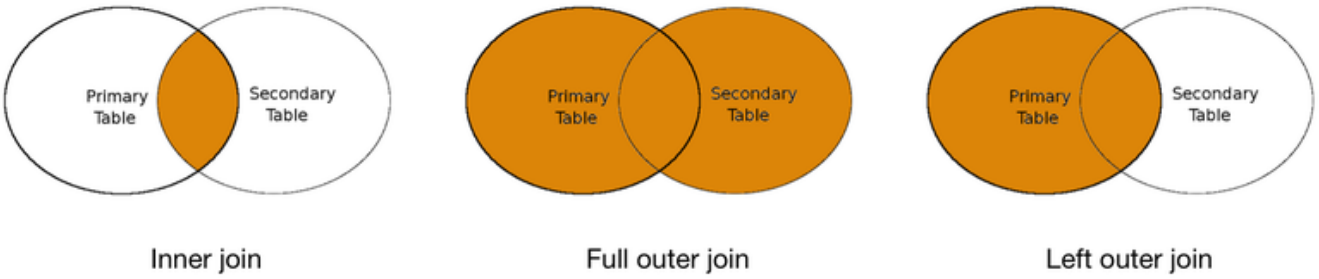


Image 2. The different join types explained. The left side circle represents the primary, or starting, dataset in the two-table join.

BC|INSIGHT - 3.5.4 Advanced subsets

User roles

Analyst

Table of contents:

- Subset Tools
  - Dynamic value fields
  - Concatenating fields
  - Uppercase text
  - Custom constraints

> **Note**
> The advanced Subset tools require some familiarity with SQL language, and available SQL methods on the platform.

Subset Tools

The ***Subset > Tools*** dialog allows you to operate on the structure and content of the subset in more detailed level. The ***Structure*** -tool shows all the fields in the joined form. The list cannot be edited and is for information only. The ***Advanced SQL*** -tool provides customisation options with SQL language and functions. The tool displays the tables and fields constructing the subset, and allows addition of new columns, and constraints to column values.

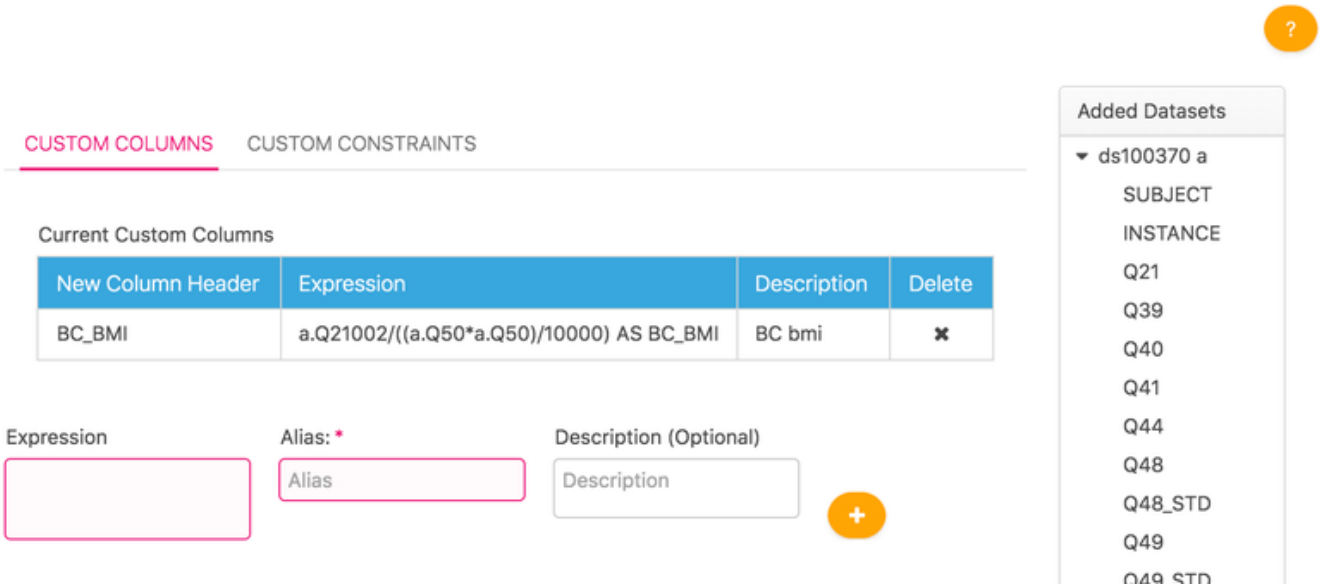


Image 1. The Subset Tools give you the option to add customisation to the subset by using SQL language and functions.

Dynamic value fields

In the above image example a user has generated a simple algorithm to calculate a dynamic value to a new field. The Expression field contains the definition of the value, (you should use standard SQL mathematical symbols), the Alias field defines the name for the new field, and Description defines the user-friendly long name for the dynamic field. Make sure both are unique within the context of your dataset. Clicking the '+' button will evaluate the expression and move it to the top in parsed format, where you can remove expressions from the dataset. If evaluation is not successful, for example there is an SQL syntax error, an error will be displayed and the expression is not added to the Custom columns.

Note that the easiest way to get the field names correct, together with the table aliases, is to drag and drop your wanted field from the right hand side list to the fields. All field names need to be specified with the table alias, by default this is 'a', and any joined tables will receive aliases in alphabetical order starting from 'b'.

Concatenating fields

It is possible to create a new field in the joined or subset table, which concatenates values from two other fields. The SQL method used for concatenation is called CONCAT, and an example of this can be seen in Image 1. One can combine typically text fields, or numerical fields together, and the resulting new field is always considered to be text -type. In case of large numbers or decimal numbers it is important to transform the values properly, in order to create a text presentation that is suitable for the purpose of the field. In the example in Image 1 a large numerical value (chromosomal distance) would otherwise be displayed with scientific notation for large numbers (like 10^6), which will not be useful for the purpose of the new field, unless the value was 'cast' to the INT data type. This cast forces the concatenation method to treat the large numerical value differently, and creates a proper text output, like 21:3322112.

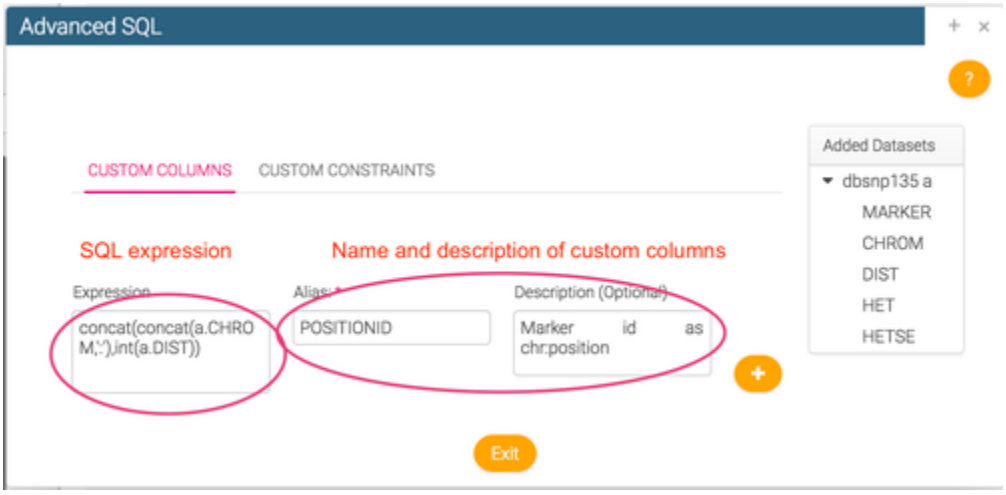


Image 2. How to create chromosomal location based ID for genetic markers, by using CHROM and DIST fields in dbSNP marker annotation table. After navigating the to Subset tool in the top level menu, and selecting the dbSNP table as subset starting point, one can choose from Tools Advanced SQL. This opens the options for adding Custom Columns. In this a concatenation of CHROM, ':', and DIST is created by first

concatenating CHROM and ':' values, and then the DIST value as INT data type. New field goes by name POSITIONID, with appropriate column description, and is after creating the subset, ready to be used as alternative ID for genetic markers. This particular example is handy for renaming markers from coordinate based to RS code based system.

Uppercase text

It is sometimes necessary to convert text, like genetic marker or probe names, to uppercase, in order to match them with existing mapping information. Typically in BC|INSIGHT map datasets present genetic marker identifiers in uppercase format and this may cause unexpected results when joining different tables together. To convert in a dataset any text field to uppercase, you need to effectively create a new column on the fly, which provides the uppercase version of the text variable. This column can then be used to match to marker identifiers when creating joins. Image 3 shows the SQL element needed to do this conversion.

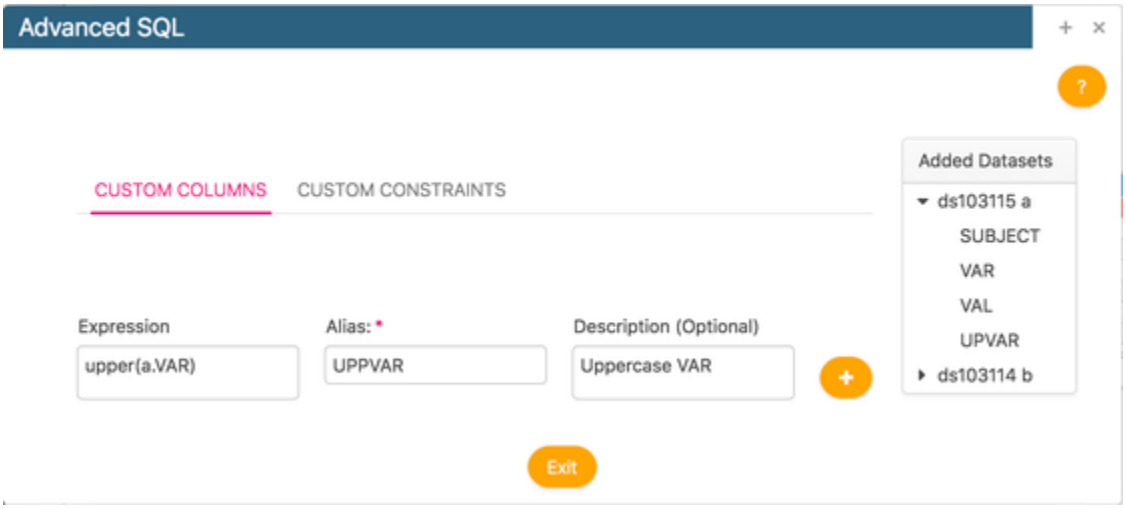


Image 3. To convert a text field to uppercase, you can use the upper() method. In this example we're converting the VAR field in multiQTL table to uppercase, under new column named UPPVAR, and with description "Uppercase VAR". This UPPVAR field can now be used to create joins with tables where the multiQTL variable VAR is in uppercase format.

Custom constraints

The Custom Constraints page allows user to define filtering of data based on more complex SQL expressions, for example conditional evaluation of values from multiple fields, or values based on arithmetics between fields. The below example describes how a mathematical value (relational score) generated from two fields can be used to limit the values in the table. In addition the user has introduced a simpler evaluation, which could also be generated using the Subset tool's filtering functions.

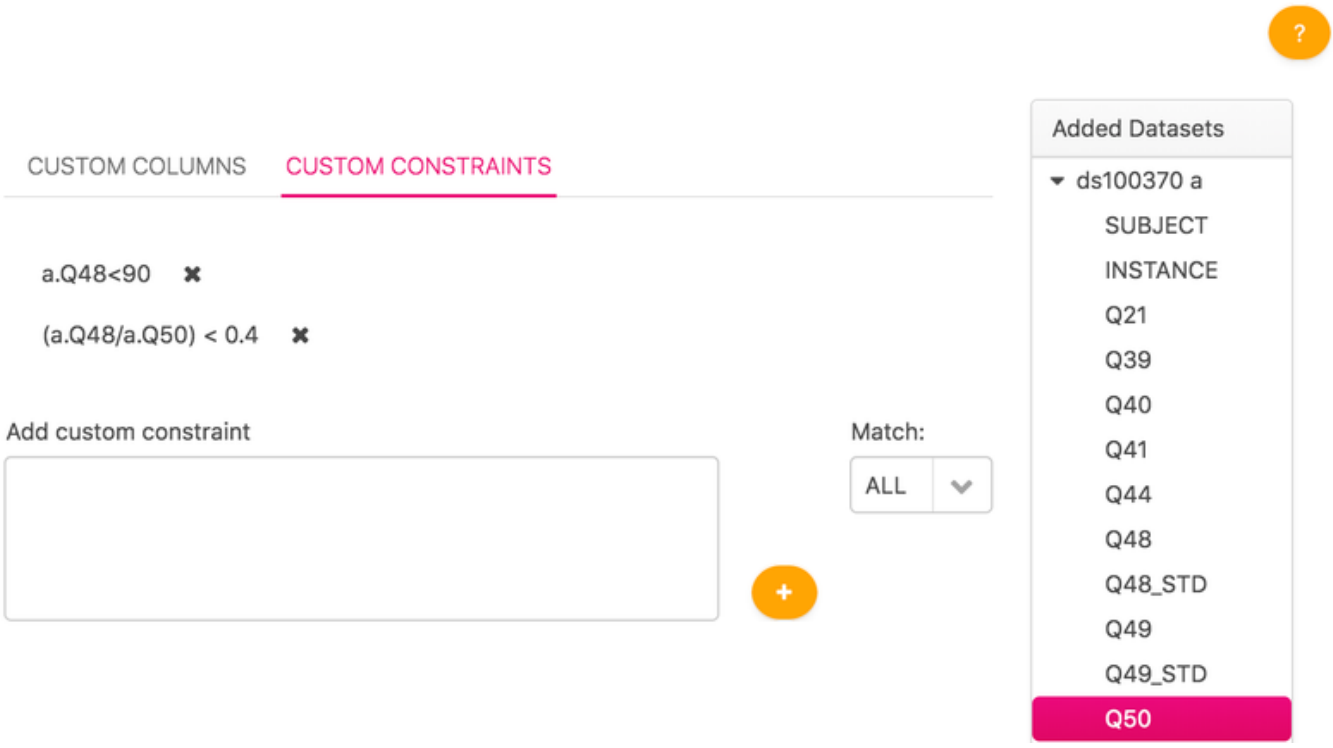


Image 4. Definition of Custom Constraint to filer values in the dataset. The constraints can be stacked together, and removed.

A Custom constraint is simple SQL standard expression that is evaluated and added to the top of the dialog view, after user clicks the '+' button. If evaluation is not successful, for example there is an SQL syntax error, an error will be displayed and the expression is not added to the constraints.

User can stack multiple different constraints and select the logical evaluation between all constraints. Selecting in the Match field option 'ALL' means that all constraints are evaluated with AND logical operator, i.e. all constraints must be true in the final subset. Selecting option 'ANY' means that if any of the constraints is true for a value row, it is included in the subset.

Note that the easiest way to get the field names correct, together with the table aliases, is to drag and drop your wanted field from the right hand side list to the fields. All field names need to be specified with the table alias, by default this is 'a', and any joined tables will receive aliases in alphabetical order starting from 'b'.

BC|INSIGHT - 3.5.5 Gene range annotations

User roles
Analyst

Table of contents:

- Annotating a genetic map with genes
- Using gene map with genetic features
  - Genotypes and methylation markers
  - Markers in MultiQTL dataset

Genetic range in BC|INSIGHT

Genetic range is used to determine locations of genomic structures, most commonly genes. In BC|INSIGHT any structure that has a genetic range can be used to annotate features within those ranges. A typical use case is combining information from a marker map (like dbSNP) with an annotation data set that has gene names, and possible other information, presented with their genetic range. BC|INSIGHT uses special inbuilt annotation for genetic range, and this annotation is present in forms like "Genome regions". The range is build from chromosome and position information, like start and stop position.

Typical use case of this feature are annotation of marker or feature maps with chromosomal positions, to match ranges of genes. This information is then combined with the experimental data like genetic variants and methylation markers, to create a dataset that can be filtered for genes and gene lists.

Annotating a genetic map with genes

At the moment the BC|INSIGHT provides an annotation tool for map data that has genomic range annotation. When you select a dataset, like dbSNP map, you will be able to launch the tool from DATA page, ***Tools and Export Dataset tools Annotation***. The dialog then displays a selection tree for the dataset (Image 1), which contains information about your markers with genomic ranges. The system will check the compatibility of your choices and indicates if it is happy to continue by displaying a short message after the check. In our example we have Gene list information in hg38 build, and we want to combine our SNP list from dbSNP with the Gene list, to get markers matching with gene regions.

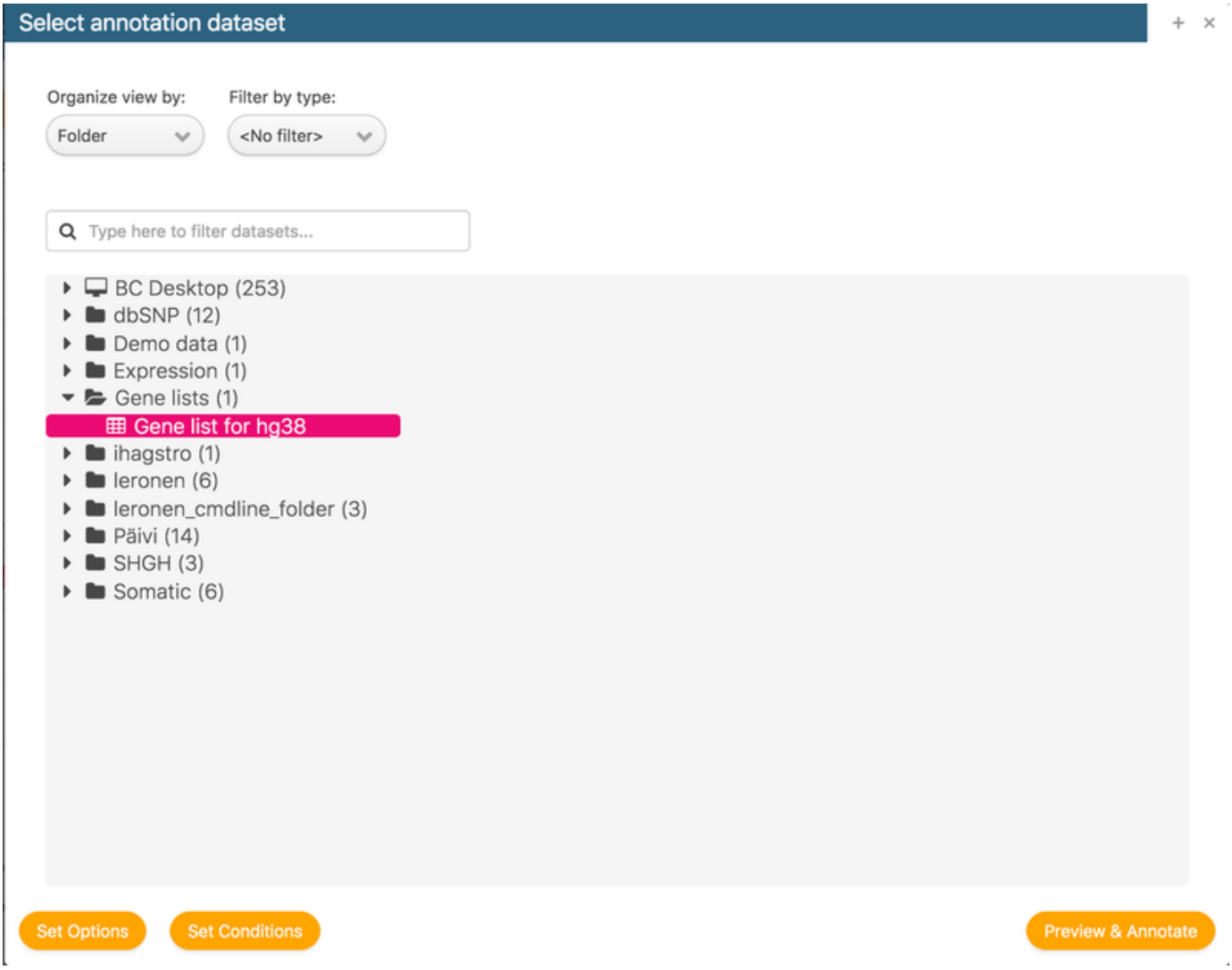


Image 1. Starting point for the annotation. You choose here the genomic information or annotation dataset you want to apply to you marker list.

You can proceed to create the annotation dataset directly without any additional options. The Preview & Annotate -button will provide you with the conditions that are used to combine these two datasets into a new one. In most cases this is what you're after, and you get all of your markers annotated with gene names and other information based on the location of the markers (Image 2).

Image 2. The preview of the criteria used to generate the new annotated dataset. Note that there are no additional conditions defined in this example.

However, sometimes you may wish to create a more restricted view, include only some data fields, or only some chromosomes. Set options -tool allows you to select the chromosomes you wish to include, change the range overlap criteria used to create the genetic matching, and choose the data columns from either dataset (Image 3).

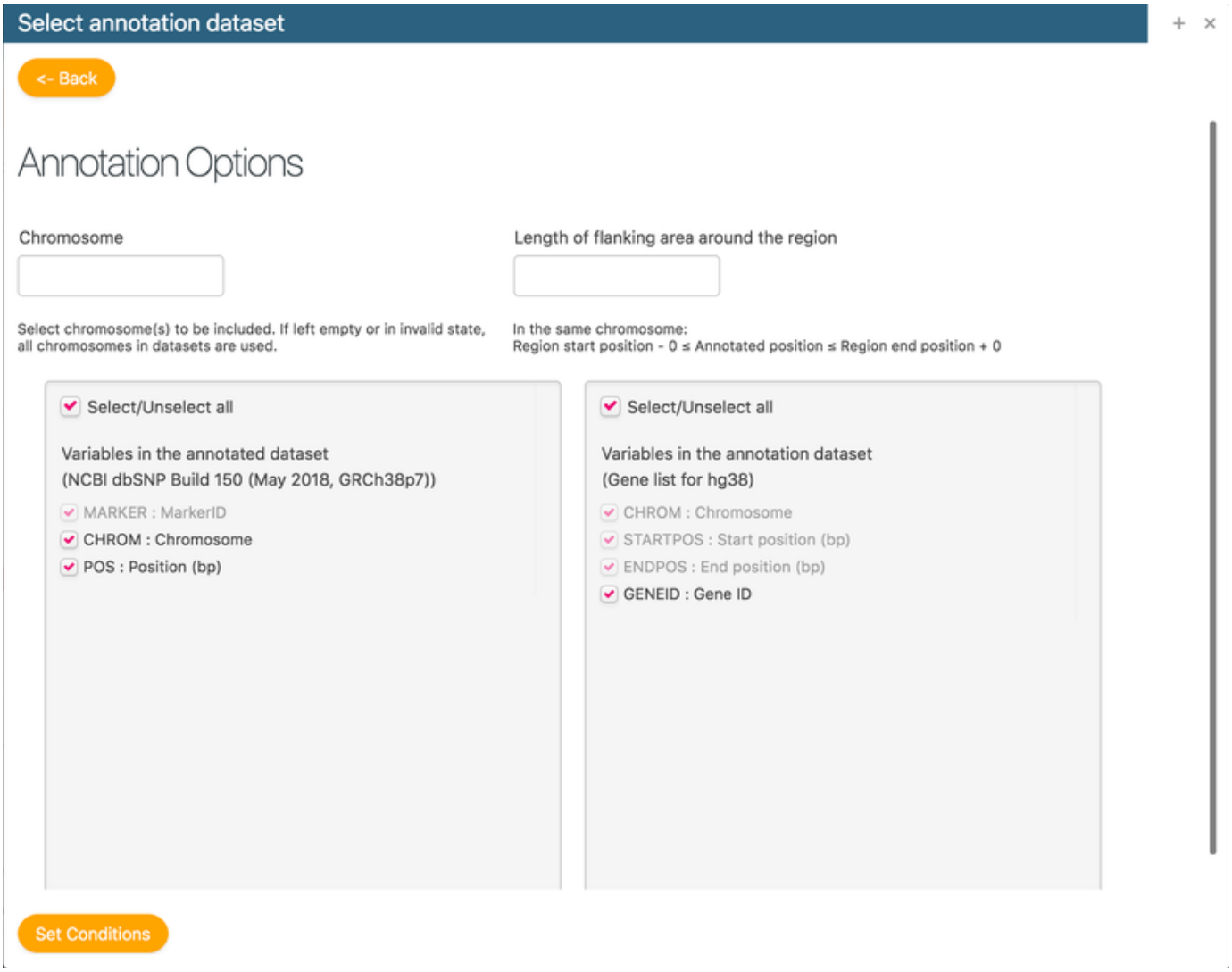


Image 3. In the options you can exclude data columns from either dataset, include only certain chromosomes, or tweak the flanking region of ranged features to increase your capture.

Annotation tool allows in the Conditions -dialog more detailed restrictions based on data column -specific conditions (Image 4). You can choose to only include certain data items that are missing, not missing, or match a selected list of values. In the example here we have selected to only include genes matching a list of selected gene IDs.

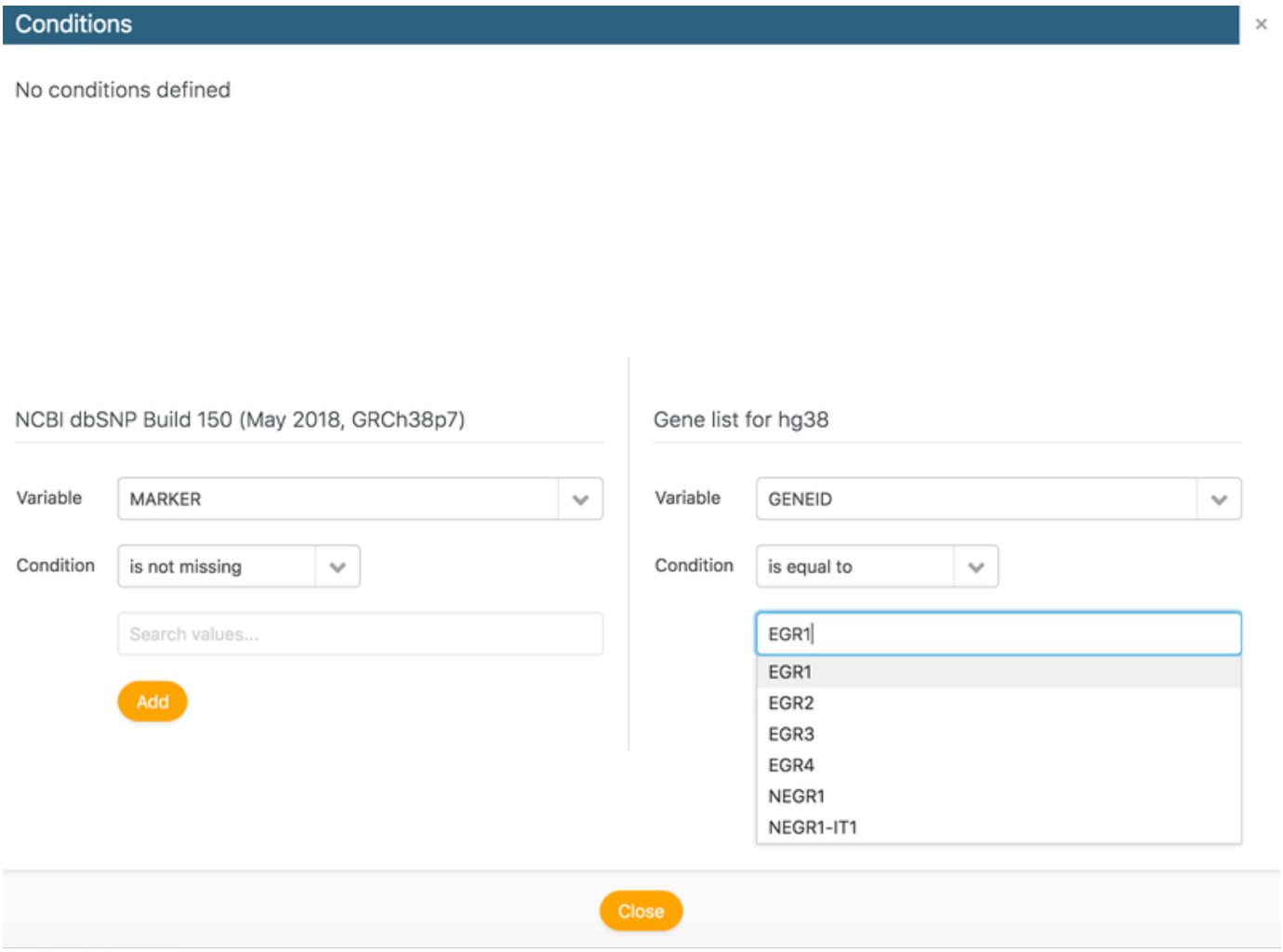


Image 4. Setting more specific conditions to the content of the annotated dataset.

After you have changed your options and settings to your liking, the Preview page will again give you a summary of the conditions for the new annotation dataset. Click OK to create the subset, which will appear in the data navigator under the starting dataset.

Using gene map with genetic features

Once the annotation of the genetic markers (variants, methylation sites) with respective gene ranges has been made, the resulting gene map can be further combined with the data of interest. Note, however, that with large amounts of data these types of joins between three tables will eventually become slower to resolve and read, and that your final dataset may suffer from slowness due to this.

Genotypes and methylation markers

To annotate genotypes with generated gene range map, you need to select in the Subset tool the MARKER (BC:marker) from the genotype dataset and the gene range map dataset. Subset tool will automatically suggests this join condition, if the genotype dataset marker field has correct BC:marker ontology. This type of join will then effectively provide you with the following fields:

SUBJECT	MARKER	CHROMOSOME	POSITION	GENE
dummy1	RS123456	20	19547580	SLC24A3

This structure allows you to filter in Data grid (or further in Subset tool) genetic variants based on their existence in the range of specific genes.

Markers in MultiQTL dataset

To annotate genetic or methylation markers in a MultiQTL dataset, you need to first make sure that the marker names match between the gene range map and the MultiQTL dataset. For example be aware of possible uppercase-lowercase differences, which you may need to resolve using the technique described in Advanced subsets. You need to reject any automated join choices the Subset tool may offer, and manually select VAR (BC:variable_name) from the MultiQTL dataset, and match it with the respective BC:marker field in the gene range map dataset (typically MARKER). Image 5 displays a typical join situation. You will generate the following structure for further filtering:

SUBJECT	VAR	UPVAR	MARKER	CHROMOSOME	POSITION	GENE
dummy1	cg00210842	CG00210842	CG00210842	20	19547580	SLC24A3

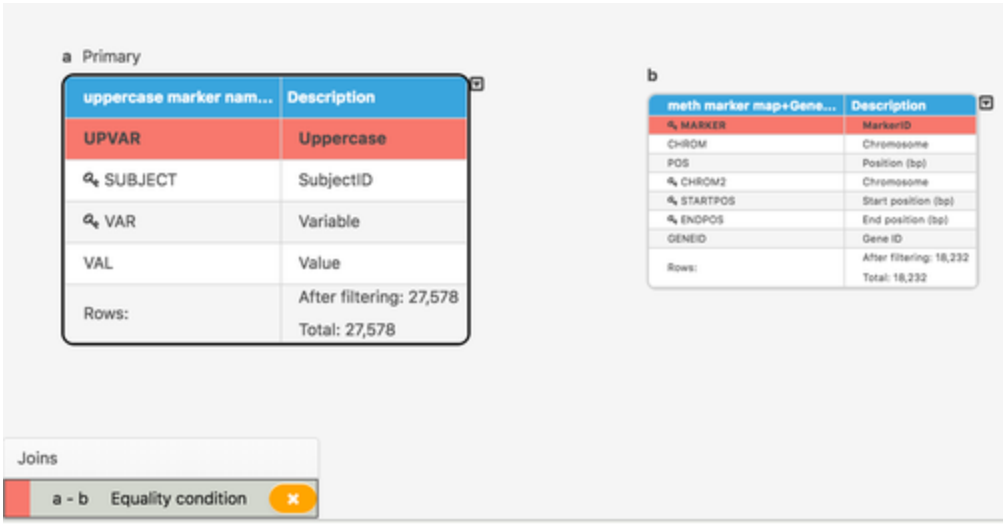


Image 5. In the above example the multiQTL dataset VAR field had to be first converted to uppercase in a new column called UPVAR. This uppercase column was then used to join the gene range map via MARKER column to generate filterable MultiQTL dataset.

BC|INSIGHT - 3.6 Uploading data

User roles

Data manager

Child pages:

- BC|INSIGHT - 3.6.1 Update an existing subject
- BC|INSIGHT - 3.6.2 Add a new subject to the dataset
- BC|INSIGHT - 3.6.3 Upload a single file
- BC|INSIGHT - 3.6.4 Upload a file using the upload wizard
- BC|INSIGHT - 3.6.5 Upload files on server
- BC|INSIGHT - 3.6.6 Save files to datasets as objects
- BC|INSIGHT - 3.6.7 Sample-Subject ID conversions
- BC|INSIGHT - 3.6.8 Revert accidental changes

There are several ways to enter data into BC|INSIGHT datasets. Supported file formats for all data types are TSV and CSV, and in addition various other formats are supported depending on the target datatype, form example VCF for genomic data. Data manager can upload phenotype data files either directly without conversion of data, or by using Upload Wizard -tool to remap file headers to dataset columns, and control the data conversion parameters. For many other datatypes specific converters are available. File uploads can be done in insert and update mode.

In addition to bulk file uploads it is possible for Data manager to add new data row by row, in which case data input takes place through a dialog. Similarly, editing single row of data takes place via an editor dialog. User must have Write permission to the dataset for any data upload actions.

BC|INSIGHT - 3.6.1 Update an existing subject

User roles

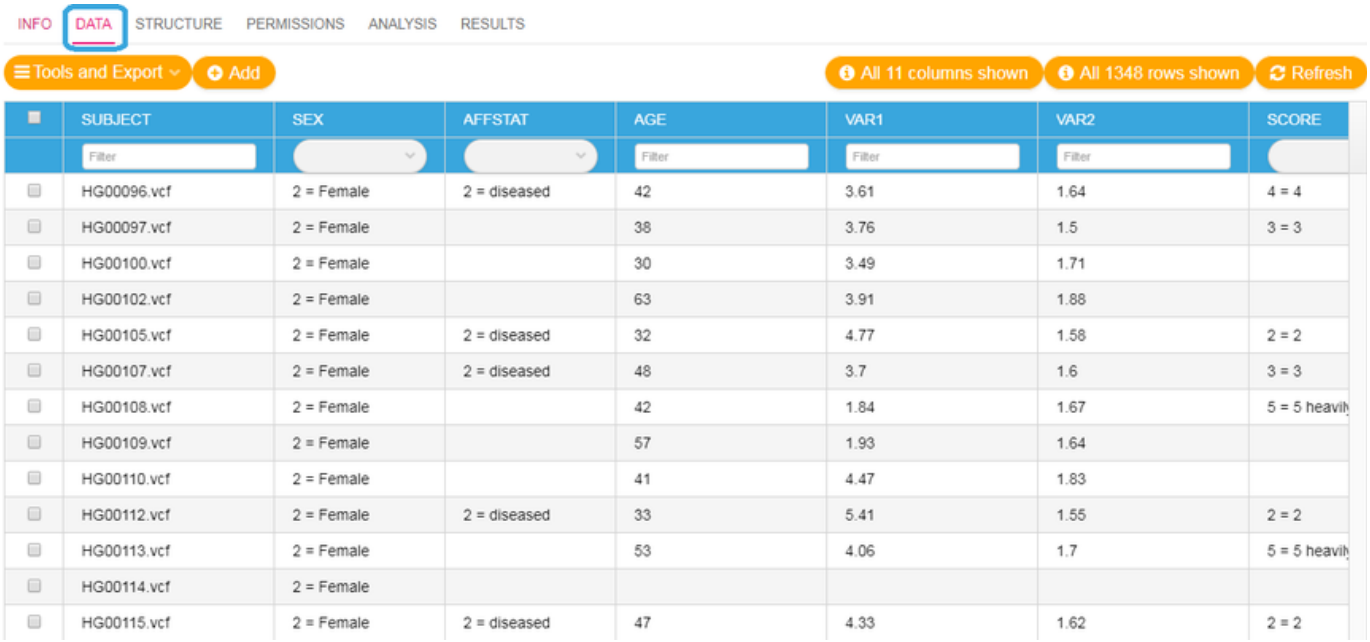
Data manager

Updating an existing subject is most often performed by research nurses, who are entering data into the database while interviewing a subject.

To update data of an existing subject you need to have 'write' permission to the dataset. If you do not have 'write' permissions, you will only see 'View' button in the top of the data grid, and will not have access to 'Add' and 'View/Edit'.

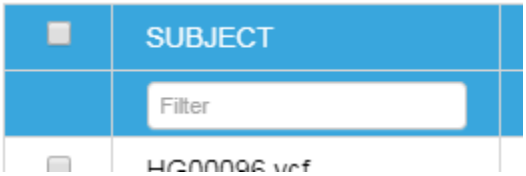
'View' -button allows you to browse the individual record and access the values in fields, but you will not be able to save any changes to values.

1. Select the ***DATA*** tab to open the DATA page.



SUBJECT	SEX	AFFSTAT	AGE	VAR1	VAR2	SCORE
HG00096.vcf	2 = Female	2 = diseased	42	3.61	1.64	4 = 4
HG00097.vcf	2 = Female		38	3.76	1.5	3 = 3
HG00100.vcf	2 = Female		30	3.49	1.71	
HG00102.vcf	2 = Female		63	3.91	1.88	
HG00105.vcf	2 = Female	2 = diseased	32	4.77	1.58	2 = 2
HG00107.vcf	2 = Female	2 = diseased	48	3.7	1.6	3 = 3
HG00108.vcf	2 = Female		42	1.84	1.67	5 = 5 heavil
HG00109.vcf	2 = Female		57	1.93	1.64	
HG00110.vcf	2 = Female		41	4.47	1.83	
HG00112.vcf	2 = Female	2 = diseased	33	5.41	1.55	2 = 2
HG00113.vcf	2 = Female		53	4.06	1.7	5 = 5 heavil
HG00114.vcf	2 = Female					
HG00115.vcf	2 = Female	2 = diseased	47	4.33	1.62	2 = 2

2. Enter the subject ID in the ***SUBJECT Filter*** field.

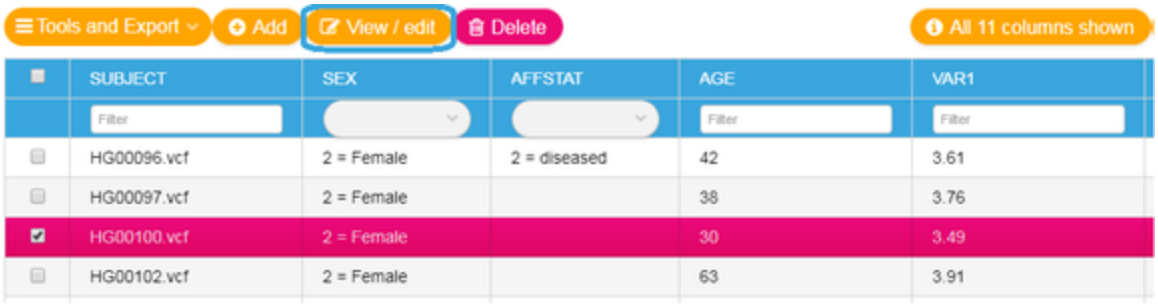


-OR-

Scroll through the list to find the subject you want to update.

SUBJECT	SEX	AFFSTAT	AGE	VAR1
HG00096.vcf	2 = Female	2 = diseased	42	3.61
HG00097.vcf	2 = Female		38	3.76
HG00100.vcf	2 = Female		30	3.49
HG00102.vcf	2 = Female		63	3.91
HG00105.vcf	2 = Female	2 = diseased	32	4.77
HG00107.vcf	2 = Female	2 = diseased	48	3.7
HG00108.vcf	2 = Female		42	1.84

3. Select the required subject check box. The ***View > Edit*** button displays.

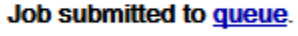


4. Select the ***View > Edit*** button to display the subject's details.

5. Make the required changes.

6. Select ***Save***.

7. Click the queue link to follow the progress of the job in the BC|INSIGHT queue system



8. See chapter *Viewing results and reports* for more detailed information on reports generated upon the data entry job.

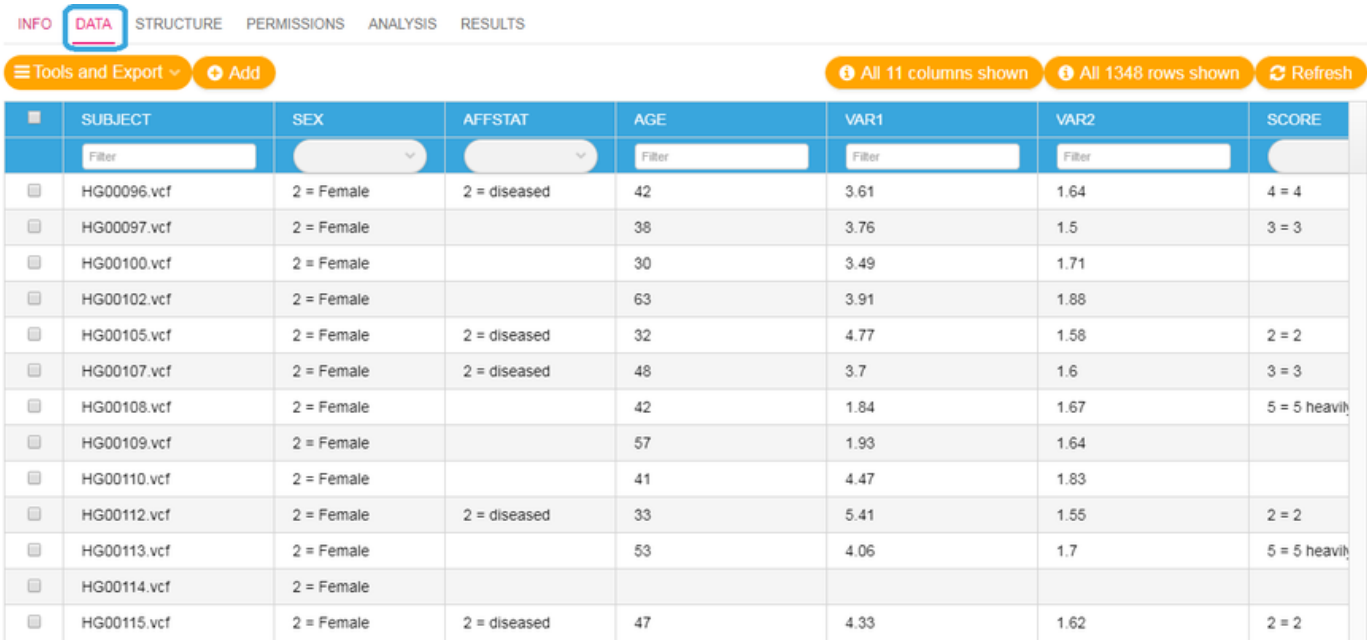
BC|INSIGHT - 3.6.2 Add a new subject to the dataset

User roles
Data manager

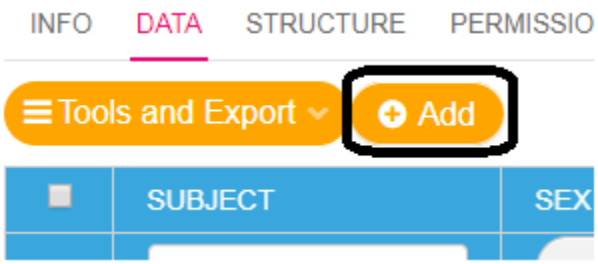
A new subject is added to the database when required. This task is most often performed by clinical research nurses when interviewing a subject.

To add a new subject:

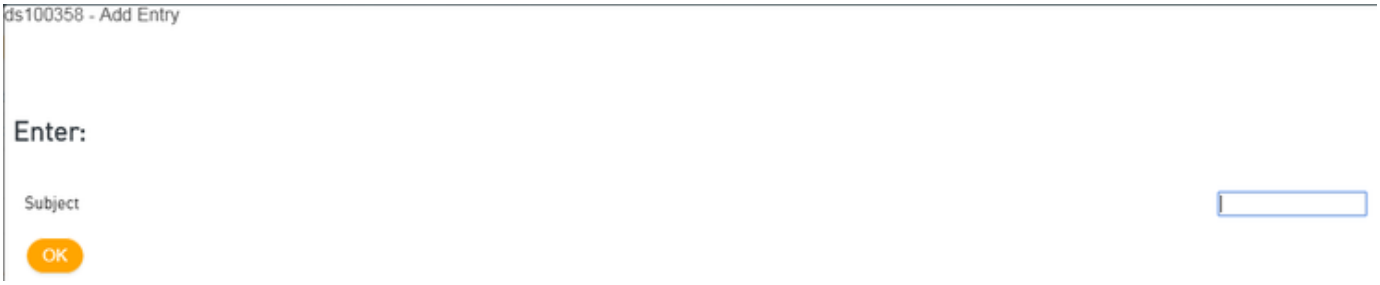
1. Select the ***DATA*** tab to open the DATA page.



2. Select the ***Add*** button.

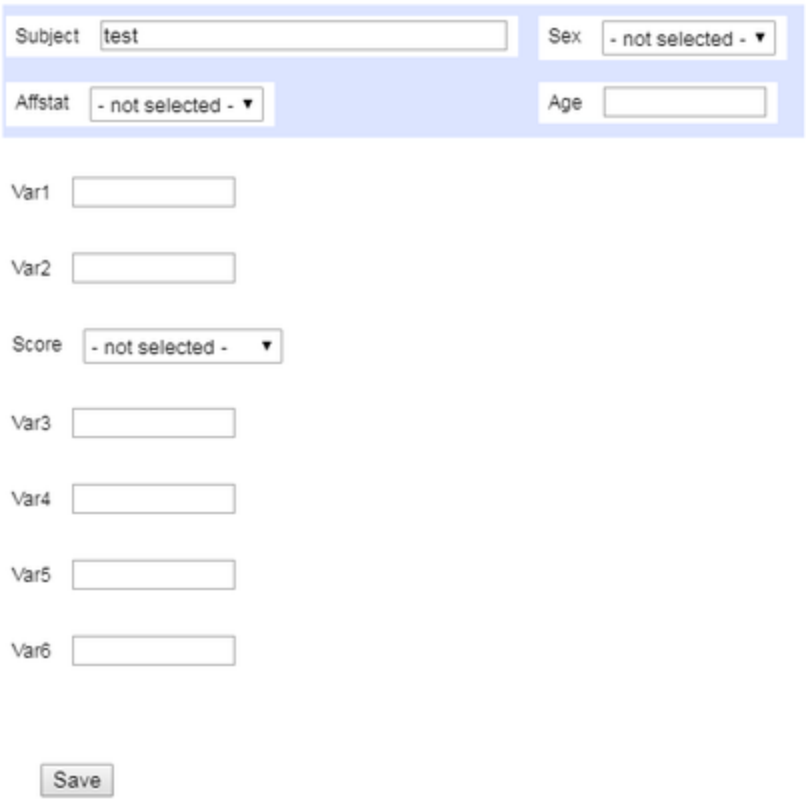


3. Enter an ID for the subject.



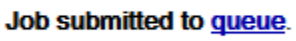
4. Click ***OK***.

5. Fill in the subject details.



6. Select ***Save***.

7. Click the queue link to follow the progress of the job in the BC|INSIGHT queue system



8. See chapter *Viewing results and reports* for more detailed information on reports generated upon the data entry job.

BC|INSIGHT - 3.6.3 Upload a single file

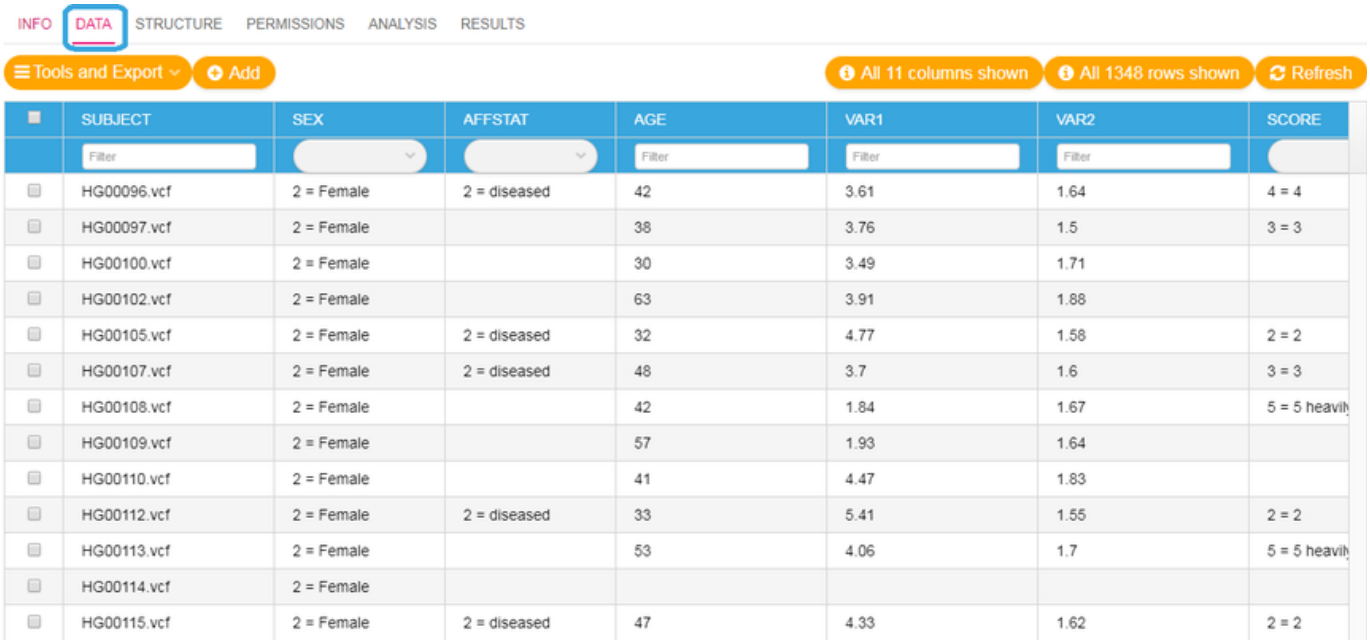
User roles
Data manager

The ***Single file*** **-**function is used when data is imported to a database from a file on user's PC. By default BC|INSIGHT accepts tabulator delimited text files with headers matching the field names (in other words, variable IDs) of the dataset, but with the data import wizard tool you can import other kinds of text files.

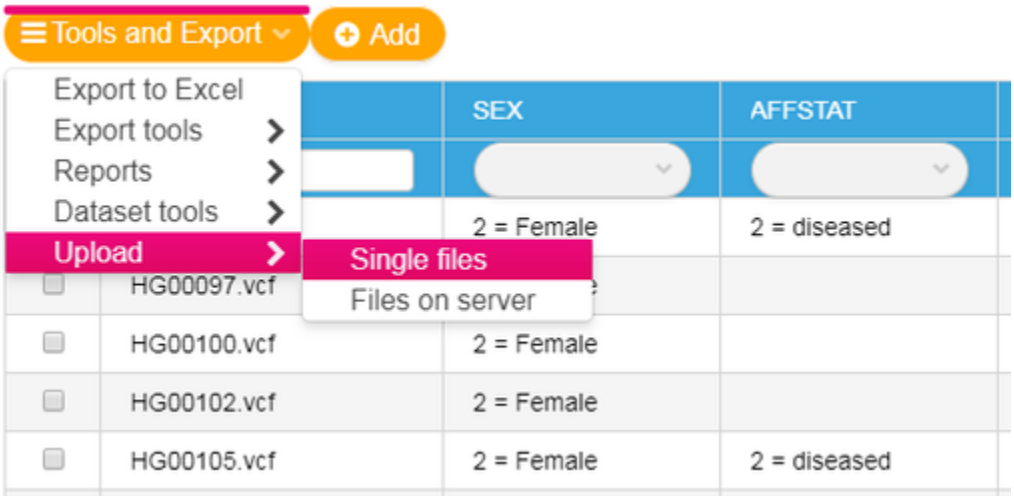
> **Note**
> Check for any duplicate key variables before you upload an input file. If there are duplicate key variables in the input file, only the values from the last row of duplicates are inserted to a dataset.

To upload a single file:

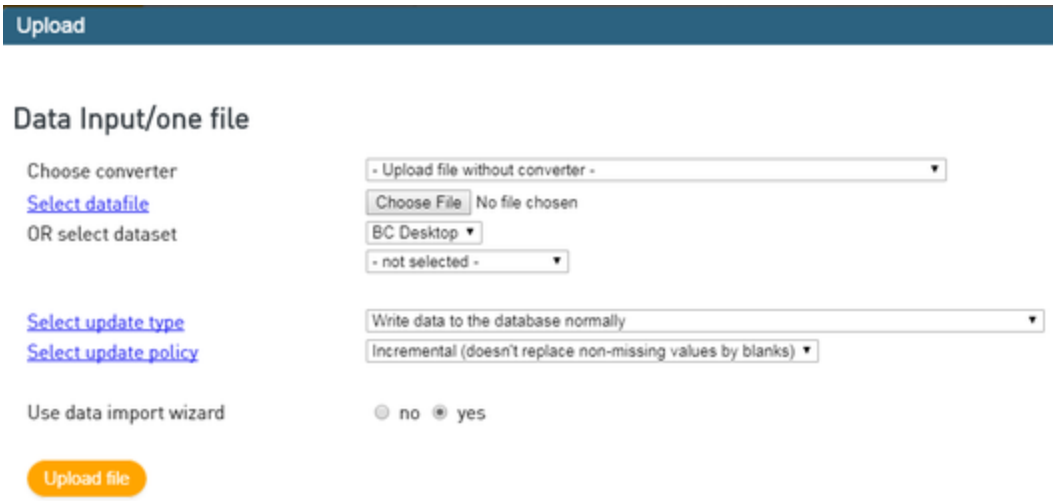
1. Select the ***DATA*** tab to open the DATA page.



2. Select ***Tools and Export > Upload > Single files***.



The ***Data Input/one*** file dialog opens.



3. Choose converter (if required)

> **Note**
> Choose a converter based on the input file format: Variation data always requires a converter specification. However, in the case of other types of dataset (in other words, annotation, pedigrees, phenotype, multiQTL), only choose a converter when the input file is not in BC format as described here.

-OR-

Select a dataset with the same form to copy dataset information to a new dataset.

4. ***Select update type*** to specify how overlapping information is treated

i. Select ***Write data to the database normally*** to upload new rows to a dataset.

The overlapping key value(s) information in the input file replaces the existing information and is reported in the ***RESULTS >*** results folder.

ii. Select Do NOT write data to the database, but generate report which values update would change to run a test report.

The database values are not updated but the test report in ***RESULTS >*** result archive shows the projected changes between the original values and the new values.

iii. Delete rows defined in the file for removing data rows that have duplicate key value(s) with existing data. By default, this action is only allowed to be done by the dataset owner, but BC Support can also grant this permission to database administrators.

5. ***Select update policy*** to specify the management of empty values.

i. Select ***Incremental (doesn't replace non-missing values by blanks)*** if you do not want empty values in the input files to replace the existing data values if there are overlapping key and variable value(s).

ii. Select ***Direct (allows to replace non-missing values by blanks)*** if you want empty values in the input files to replace the existing data values if there are overlapping key and variable value(s).

6. Use data import wizard "yes" to check the match between your input file and dataset form. Anwser "no" if you want to skip this step.

> **Note**
> If you select any converter other than ***Upload file without converter***, the upload data import wizard is not in use.

7. Select ***Upload file***.

8. Select queue.

A screen displays where you can follow the progress of the job in the BC|INSIGHT queue system.

See chapter *Viewing results and reports* for more detailed information on reports generated upon the data entry job.

BC|INSIGHT - 3.6.4 Upload a file using the upload wizard

User roles
Data manager

The ***Single file*** tool as described in 3.6.3 Upload a single file supports for using data import wizard tool when the text file used in the upload does not have an exact match in

- field names
- options of multiple choice questions, or
- missing values have been specified with some values instead of "null", or
- non-database format date values (YYYY-MM-DD) are specified to the form.

> **Note**
> The data import wizard can be used to upload a text file to a dataset of type: phenotype, annotation, sample IDs or pedigree.

To select the upload options using the upload wizard:

1. Follow the instructions to upload a single file as described in section 3.6.3 Upload a single file
2. Select ***yes*** in ***Use data import wizard***.
3. Select ***Upload file***.

The ***Upload options*** dialog is shown.

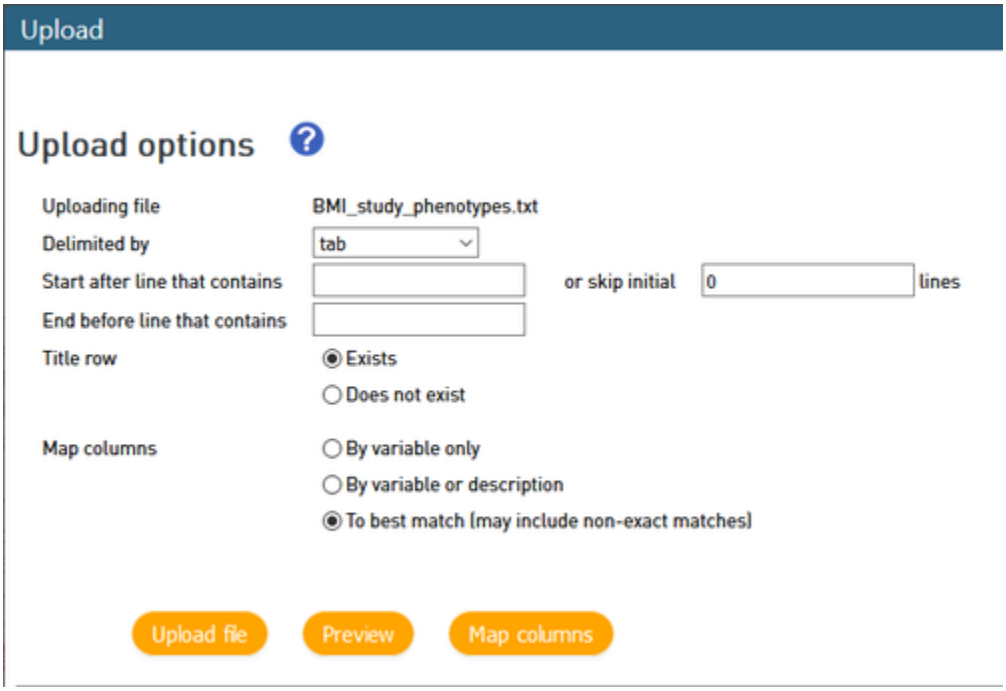


Image 1. Upload options.

> **Note**
> The Upload options dialog contains two sets of buttons for ***Upload file***, ***Preview*** and ***Map columns***. These perform the same function, but are included twice to make it easier for the user to select them as the dialog is long.

4. Select a file delimiter from the drop-down list in the ***Delimited by*** field:

- tab
- comma
- semicolon
- space
- multiple spaces
- space aligned

5. Exclude rows either at the top or end of a file.

6. Select whether or not an input file has a ***Title row***:

- Exists
- Does not exist

7. Select how to map the columns in the ***Map columns*** field:

- By variable only
- By variable or description
- To best match (may include non-exact matches)

8. Select ***Preview*** to check the variable matches between the input file and the form variables.

9. Select ***Upload file*** if you do not need to make any changes to the mapping.

Mapping variables

If you need to make changes to the mapping, use the columns shown in Figure 8. The first two columns, ***Variable*** and ***Description*** show the information that is in the form that was used to create the dataset. The remaining columns are editable and can be used to map the variables in the input file to the variables in the dataset form.

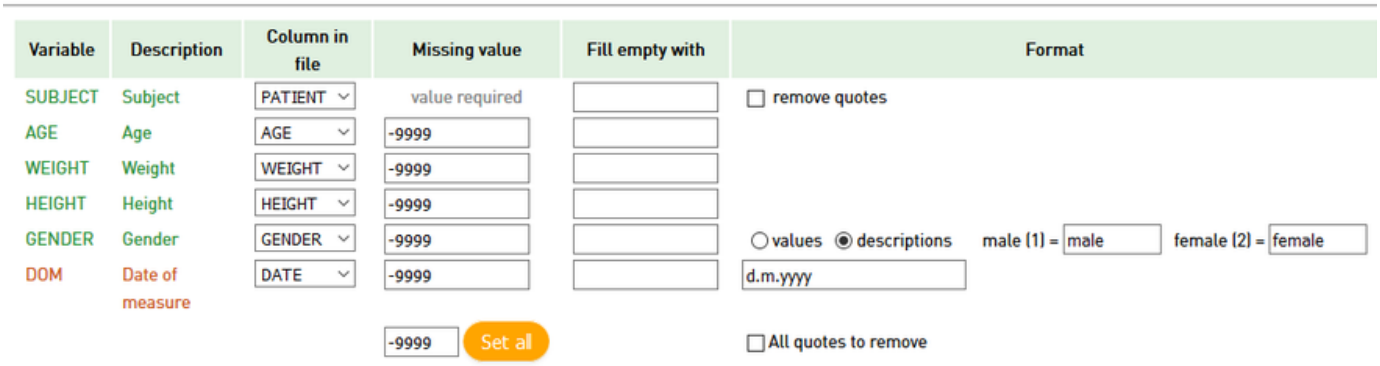


Image 2. Mapping options for variables.

1. In ***Column in file***, select variables from the input file to map to the variables in the dataset form.
    a. Green indicates a perfect match – the input file and the dataset form have exactly the same variable name.
    b. Orange needs a manual check – the dataset form and the input file have a similar variable but it is named differently. For example, DOM is the name of the variable indicating the date in the dataset form and it is named DATE in the input file. They are both the same variable but named differently, so can be mapped.
    c. Blank indicates there is no match – the same variable does not exist in the input file.
2. In ***Missing value***, type a value for a variable if required.
    a. If you want to apply the same value to all variables, type a value in the ***Select all*** field.
3. Select ***remove quotes*** if there are any quotation marks in the input file.
4. For a DATE variable, type the complete date format to match your data (default is yyyy-mm-dd):
    a. d.m.yyyy -> 25.12.1987
    b. d/m/yyyy -> 25/11/1999
    c. m/d/yyyy -> 11/25/2001
    d. m-d-yyyy -> 11-25-2001
5. Select ***values*** or ***descriptions*** for multiple choice questions in the dataset form to match those in the input file.
    a. If you select ***descriptions***, you can type text in the description fields.
6. Select ***Preview*** to check the data values that will be imported.
    a. The values that will not be imported are shown in red or the column is missing in the preview.
7. Select ***Upload file*** when you are ready to proceed and a message displays.

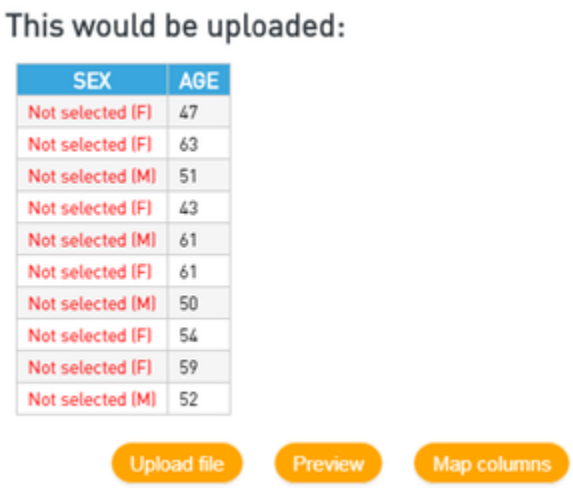


Image 3. Preview that shows problems with the import of F/M choices for SEX value.

BC|INSIGHT - 3.6.5 Upload files on server

User roles
Data manager

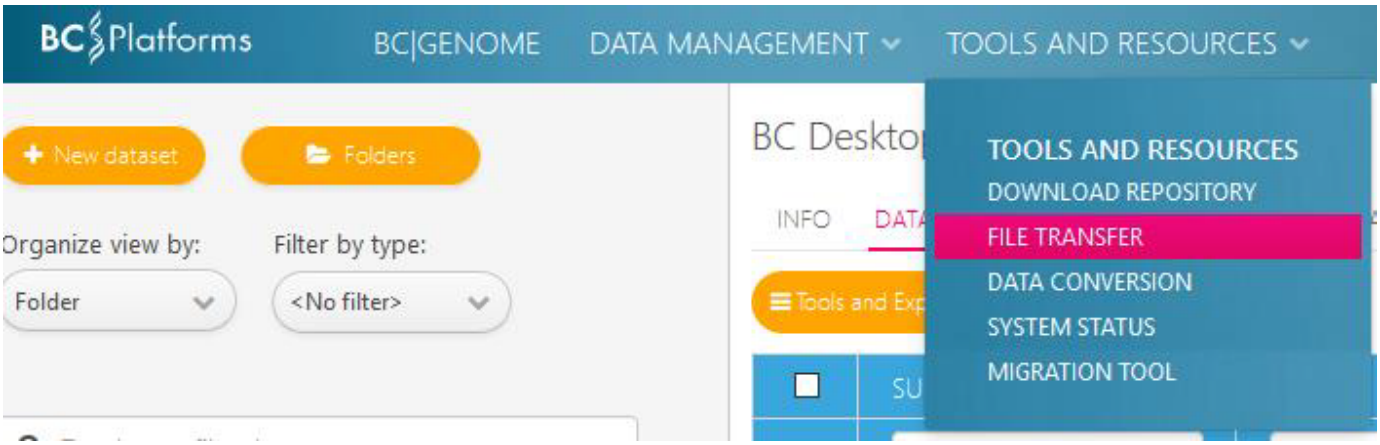
Table of contents:

- Adding files to server
- Uploading files from server
- Uploading either FASTQ or BAM files
  - FASTQ
  - BAM files
  - Uploading FASTQ/BAM files

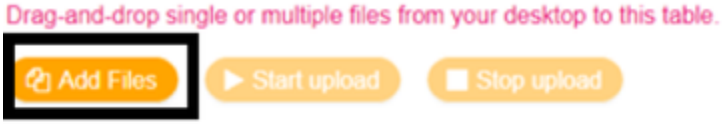
Many files can be uploaded to the dataset simultaneously using the ***Files on server*** -function. Before you start the upload procedure, you must copy the data files you want to upload to a server (usually in a folder on the server).

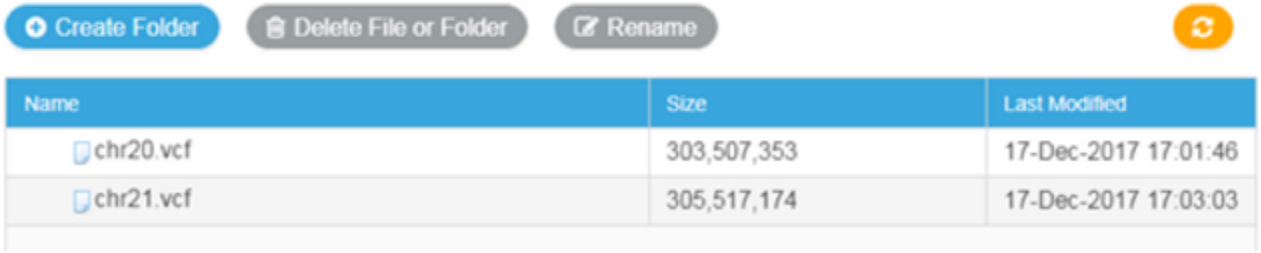
Adding files to server

1. Select ***TOOLS AND RESOURCES > FILE TRANSER***.



2. Select ***Add Files***.





3. Browse to the file(s) you want to add and select ***Open***. The chosen files are uploaded to the server folder.

You can add files to an existing folder or create a new folder.

To create a new folder:

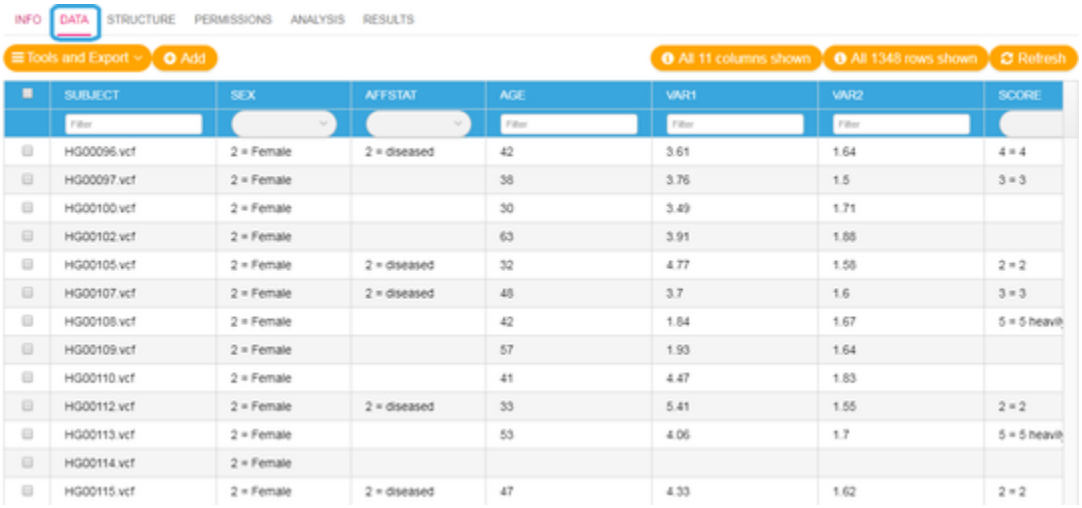
1. Select ***Create Folder*** to add a new folder.
2. Type a name for the new folder.
3. Select ***Create***

Once your folder is available, drag and drop the file(s) from the table to the required folder in the table (either the newly created folder or an already existing folder).

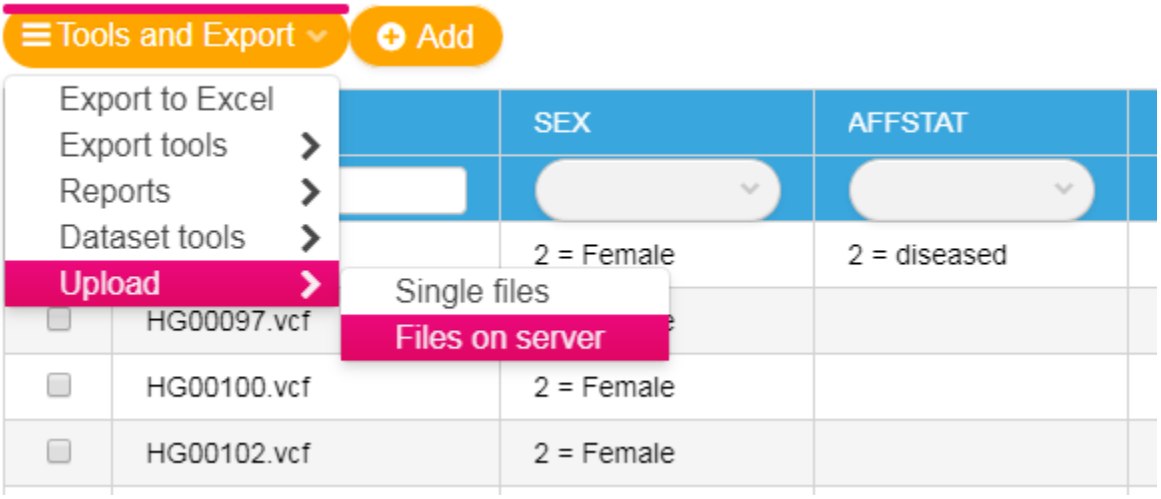
You can also delete files and folders by selecting them first and using Delete File or Folder. You can only delete one file at a time. You cannot delete a folder until the folder is empty. Renaming files and folders works with the Rename button.

Uploading files from server

1. Select the ***DATA*** tab to open the DATA page.



2. Select ***Tools and Export > Upload > Files on server***.



3. Choose a converter (if required)

> **Note**
> Choose a converter based on the input file format: Variation data always requires a converter specification. However, in the case of other types of dataset (in other words, annotation, pedigrees, phenotype, multiQTL, pedigrees), only choose a converter when the input file is not in BC format as described in section 3.2.3.
> If BC|INSIGHT uses a special reference file to find files stored in a dataset, you must choose a converter. By default upload file(s) without any converter in the case of phenotype data.

4. ***Select update type*** to specify how overlapping information is treated:

   i. Select ***Write data to the database normally*** to upload new rows to a dataset.

   The overlapping key value(s) information in the input file replaces the existing information and is reported in the ***RESULTS >*** results folder.

ii. Select Do NOT write data to the database, but generate report which values update would change to run a test report.

The database values are not updated but the test report in ***RESULTS >*** result archive shows the projected changes between the original values and the new values.

iii. Delete rows defined in the file for removing data rows that have duplicate key value(s) with existing data. By default, this action is only allowed to be done by the dataset owner, but BC Support can also grant this permission to database administrators.

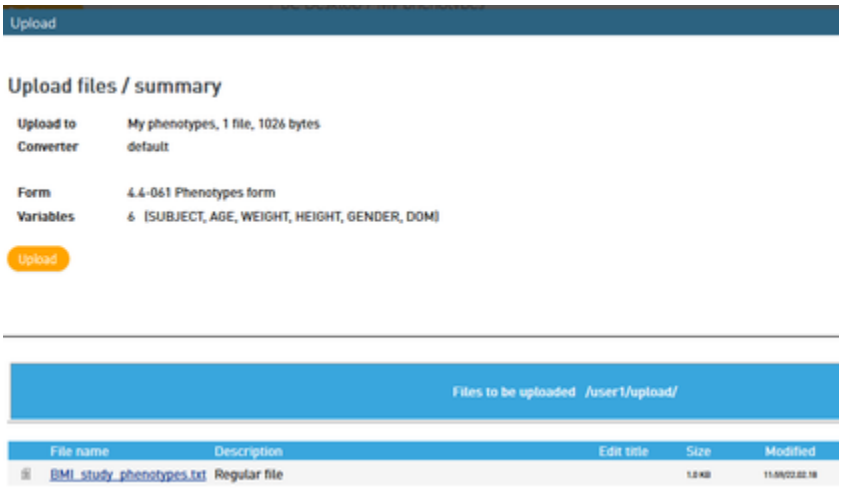
5. In ***Select upload directory*** to specify where you have your transferred file(s).

6. In ***Type search string***, type a string to match the files you want to upload.

i. If you type a ***** or leave the field blank, all files are shown.

To narrow the search, type, for example, ***b**** to show all file names that start with the letter ***b***.

7. Select ***Continue*** and make sure that the files listed are those you want to upload



For more information about viewing results and reports, see chapter 3.3 Viewing results and reports.

Uploading either FASTQ or BAM files

FASTQ

The FASTQ format is a text-based format representing sequencing in single-letter codes. The format represents also the quality scores that is used as input for BWA alignment and mapping in the BC|INSIGHT system.

FASTQ files are stored in the BC|INSIGHT database as such. The *FASTQ Files* datasets contain reference file describing the data. The variables in FASTQ dataset are listed and described in Table 1.

Table 1. Variables in the FASTQ Files form.

VARIABLE	DESCRIPTION	SOURCE	MANDATORY
SUBJECT	Subject ID		Yes
ID	File ID		Yes
FASTQ1	FASTQ file #1	Folder path and file information of FASTQ files (folderID(s)/FASTQ file)	
FASTQ2	FASTQ file #2	folder and file information for paired-end reads in a FASTQ file	
PLATFORM	Sequencer	Platform used for creating reads: 1 = 454, 2 = LS454, 3 = Illumina, 4 = Solid, 5 = ABI_Solid, 6 = CompleteGenomics (GC)	

BAM files

The BAM format is a compressed binary version of the Sequence Alignment/Map (SAM) format that describes nucleotide sequence alignment. BAM files are often companied with BAI and BAS files: BAI files contains references for tools reading the BAM files and BAS files show statistics about each alignment. For more information on BAM format, visit http://www.1000genomes.org/category/bam.

BAM files are stored in the BC|INSIGHT server as such, but the *BAM Files* dataset is needed to tell the system, what data BAM files contain. The BAM files dataset are created using the *BAM Files* form, which variables are listed and described in Table 2.

Table 2. Variables in the BAM files form.

VARIABLE	DESCRIPTION	MANDATORY
SUBJECT	ID of the subject, who has been sequenced.	Yes
ID	ID of the BAM file	Yes
CHROM	Chromosome of sequence data (if a BAM file includes data from several chromosomes leave this field empty)	No
BAM	folder path and file information of BAM files (folderID(s) /BAM file)	Yes

Uploading FASTQ/BAM files

For uploading either FASTQ or BAM files you need to perform the following steps

1. Create a csv or tab separated reference file to inform BC|INSIGHT about the location and format of your files
2. Transfer your files into your upload folder of BC|INSIGHT server using Tools and Resources / File transfer
3. Create a dataset for your FAST or BAM files
4. Upload your files

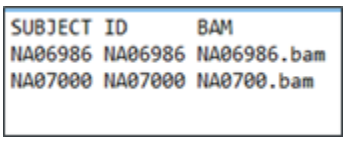
The next sections illustrates the steps in more detail:

1. Create a reference file

1.1. For FASTQ and BAM files form the combination of Subject ID and ID needs to be unique.

1.2. In the case of external file system enquire the folder path information from your server IT team.

1.3. In figure below both BAM reference file and BAM files have been transferred into the root of user's upload folder.



1.4. In figure below the reference file has been transferred into the root of user's upload folder but the FASTQ files can be found in the folder of FASTQ (the latter figure describes the reference file for paired-end reads)

Single-end reads

SUBJECT	ID	FASTQ1	PLATFORM
NA06986	SRR006142_1.filt.fastq	fastq/NA06986_SRR006142_1.filt.fastq	3

Paired-end reads

SUBJECT	ID	FASTQ1	FASTQ2	PLATFORM
NA06986	SRR006142_1.filt.fastq	fastq/NA06986_SRR006142_1.filt.fastq	fastq/NA06986_SRR006142_2.filt.fastq	3

1.5. Save the reference file either in the tab-delimited txt or csv format.

2. Transfer the files into the upload folder of BC|INSIGHT

2.1. Navigate to Tools and Resources / File transfer

2.2. Transfer your files into user's upload folder

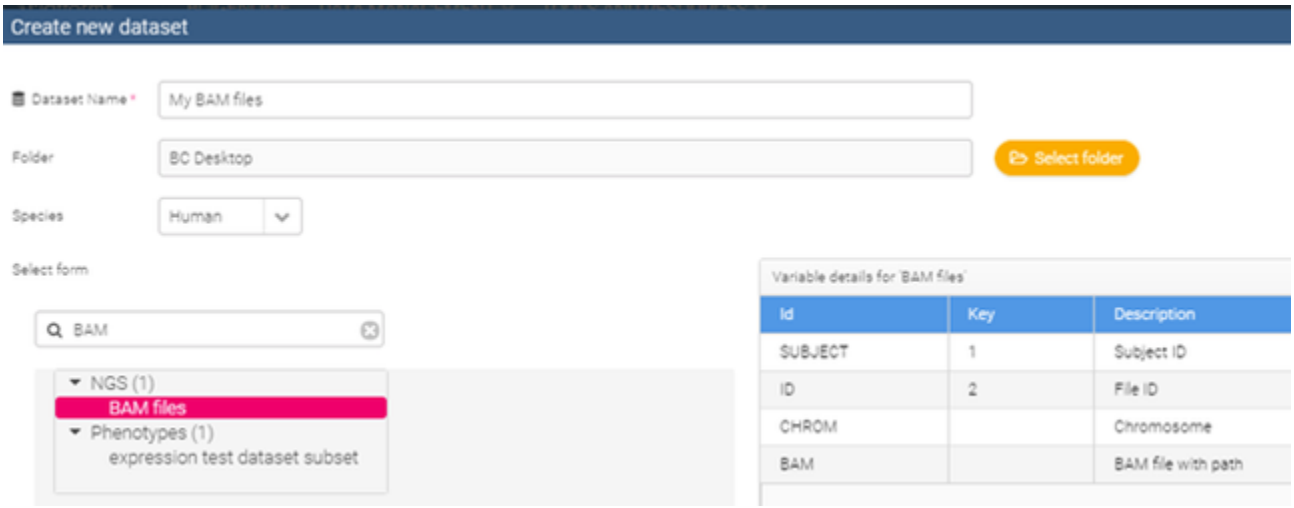
3. Create a dataset either for your FASTQ or BAM files

3.1. Select New dataset

3.2. Type a unique name of a dataset

3.3. Specify either the form of FASTQ or BAM files matching to your file type

3.4. Select Create dataset for a new dataset



4. Upload reference information to a dataset

4.1. Check you have selected the dataset for either BAM or FASTQ files

4.2. Go to DATA > Tools and export / Upload / files on server

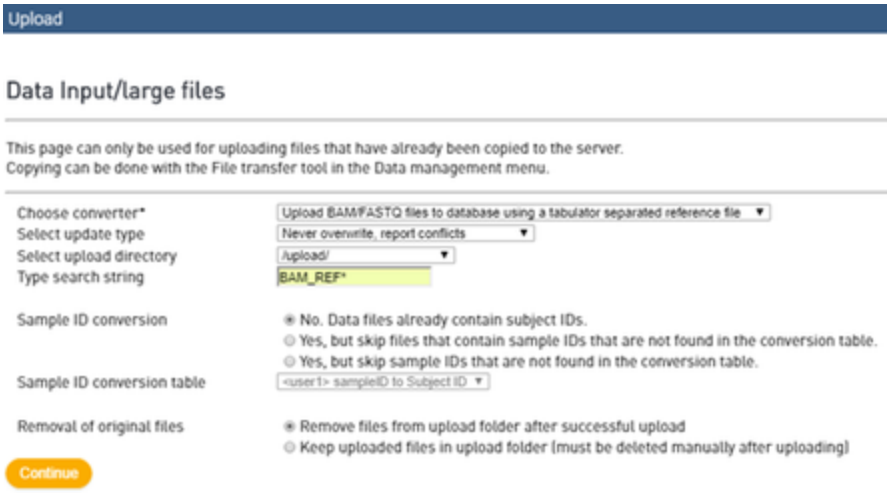
4.3. Choose the converter e.g. *Upload BAM/FASTQ files to database using a semicolon / tabulator separated reference file* depending on the column separator in the file.

4.4. Check for the directory of your reference file, the folder of /upload/ refers to root

4.5. Type the search string either for the BAM / FASTQ reference file

4.6. Check the *Removal of original files* option (if using external file system you need to skip this step)

4.7. Press Continue to check the upload summary



4.8. Check your reference file is shown in the summary list.

4.9. Press Upload to submit the job to the queue

4.10. When the job has completed check the upload report in RESULTS.

BC|INSIGHT - 3.6.6 Save files to datasets as objects

User roles
Data manager

Table of contents:

- Upload using file reference

One important aspect in BC|INSIGHT data model is the ability to save actual files per subject. These could be images, PDF reports, other documents, or genetic data files like BAM and FASTQ collections.

BC|INSIGHT has inbuilt templates for creating FASTQ (also paired-end sequencing) collections, and BAM file lists, which can be further filtered and analysed using the available tool interfaces in the system. To upload FASTQ and BAM files, please follow the specific instructions for these file types in the further chapter.

To store files as part of phenotypic data, like pathology images or reports, users are able to simply attach FILE-type variables into their phenotype forms. If you add data to these collections subject by subject, the normal Add -function will work by simply providing a file browse and upload option, when you open the data entry form. In case you need to upload many files for multiple subjects in bulk, you should follow the specific instructions provided here.

Upload using file reference

In order to upload files in bulk to phenotype table, you need to first create a separate folder for you files in the File transfer space. Move the files to that folder and create a reference file with the following structure:

SUBJECT	<FILE COLUMN NAME>
NA100001	myfolder/a_pathology_image_NA100001.TIFF

Move the created reference file to the same upload folder where you have your files to be saved to the dataset.

Start the file upload from ***Tools and Export Upload Files already copied to server***. In the opening dialog select the folder where you have saved both the reference file, and the actual files to be stored in the phenotype table. Start the upload and check the progress in queue, and finally in the result folder.

BC|INSIGHT - 3.6.7 Sample-Subject ID conversions

User roles
Data manager

Table of contents:

- Use cases

- See also
- ID conversion table creation
- VCF data upload and sample ID conversion

Use cases

BC|INSIGHT file import tools give the option for converting original identifiers in the files to new identifiers during data upload. Typical use cases for this feature are:

- Creation of pseudonyms at file import
- Conversion from sample IDs to subject IDs
- Transformation of long IDs to shorter IDs, if required due to dataset key constraints
- ID harmonisation across multiple cohorts

In this chapter we use an example of uploading VCF data to BC|INSIGHT Composite VCF dataset and simultaneously converting sample IDs to subject IDs. The process includes two steps 1) ID conversion table creation and 2) VCF data upload using the conversion option.

See also

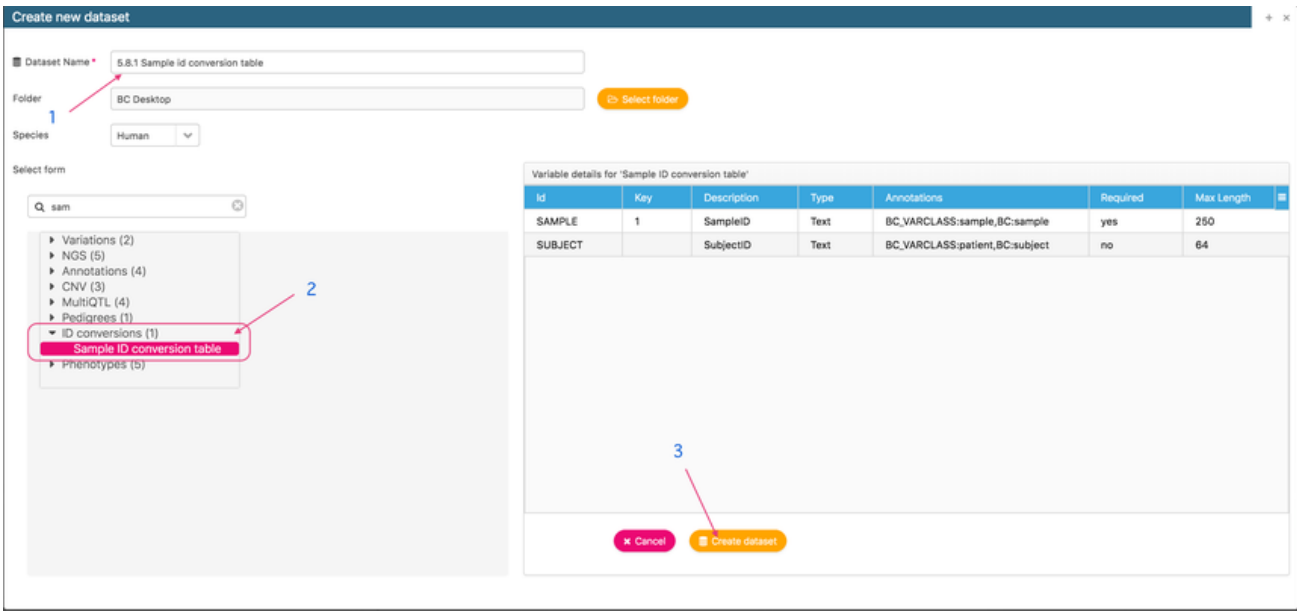
- ALIAS VIEWS - BCOS Lookup feature
- BCGENOME 5.8.1 Sample Id conversion
- Sample subject conversion table

ID conversion table creation

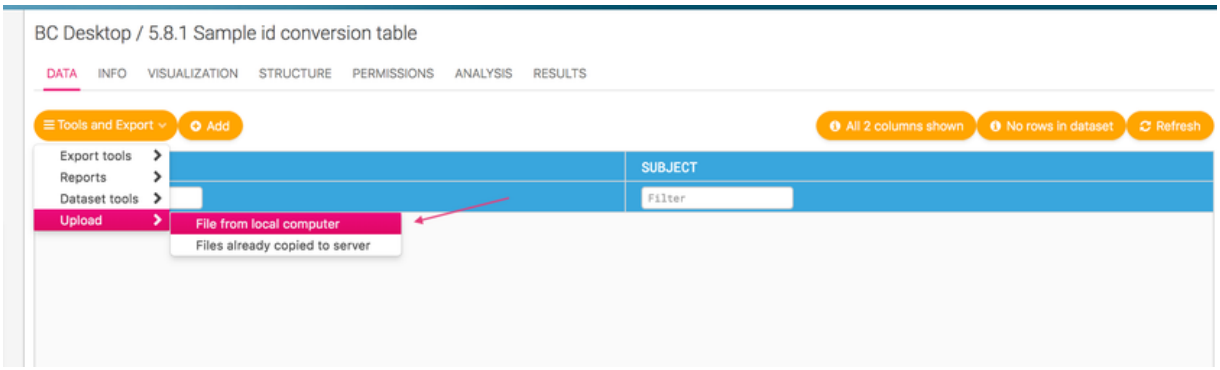
- Generate sample ID conversion file with SAMPLE (original ID) and SUBJECT (new ID) column names as follows

```
SAMPLE          SUBJECT
HG03168          GENOME_SUBJ_03168
HG03169          GENOME_SUBJ_03169
HG03172          GENOME_SUBJ_03172
```

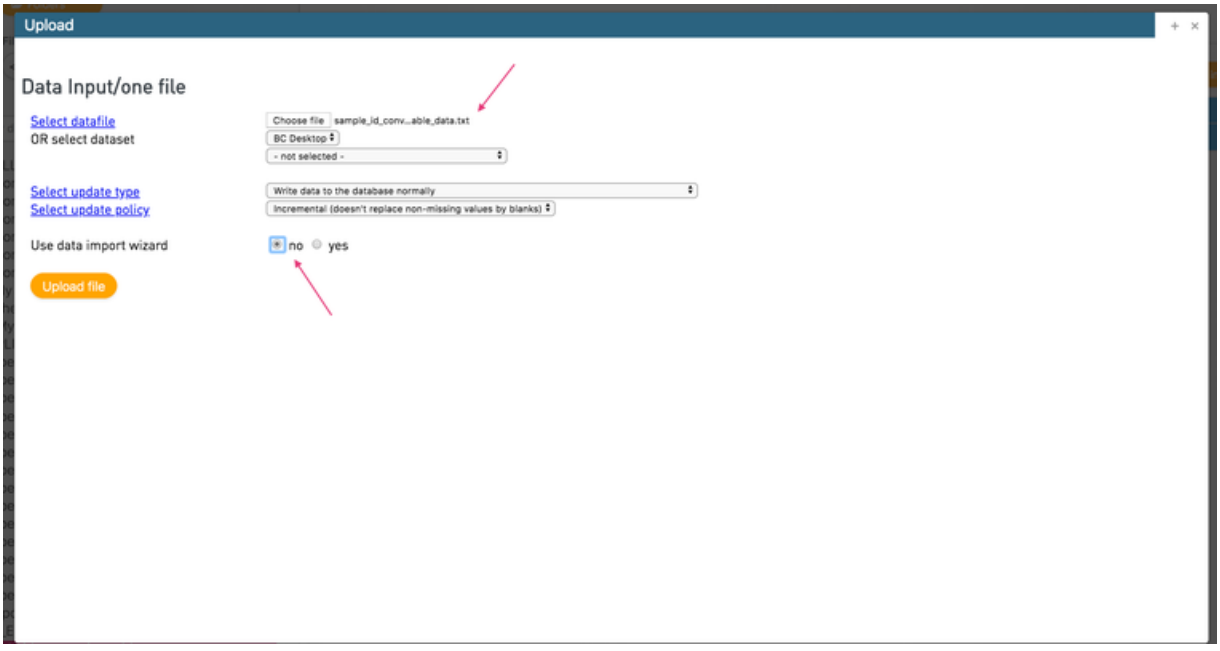
  - Save the file
- Create sample id conversion table dataset:
  - From main page click "New dataset" -button
  - In the dialog window:
    - Write Sample ID conversion table name
    - Select "Sample ID conversion table" as a form for the dataset, it is in "ID conversions" folder
    - Click "Create dataset"



- Upload ID conversion table data from local computer
  - In sample ID conversion table DATA tab, select "Tools and Export" Upload File from local computer



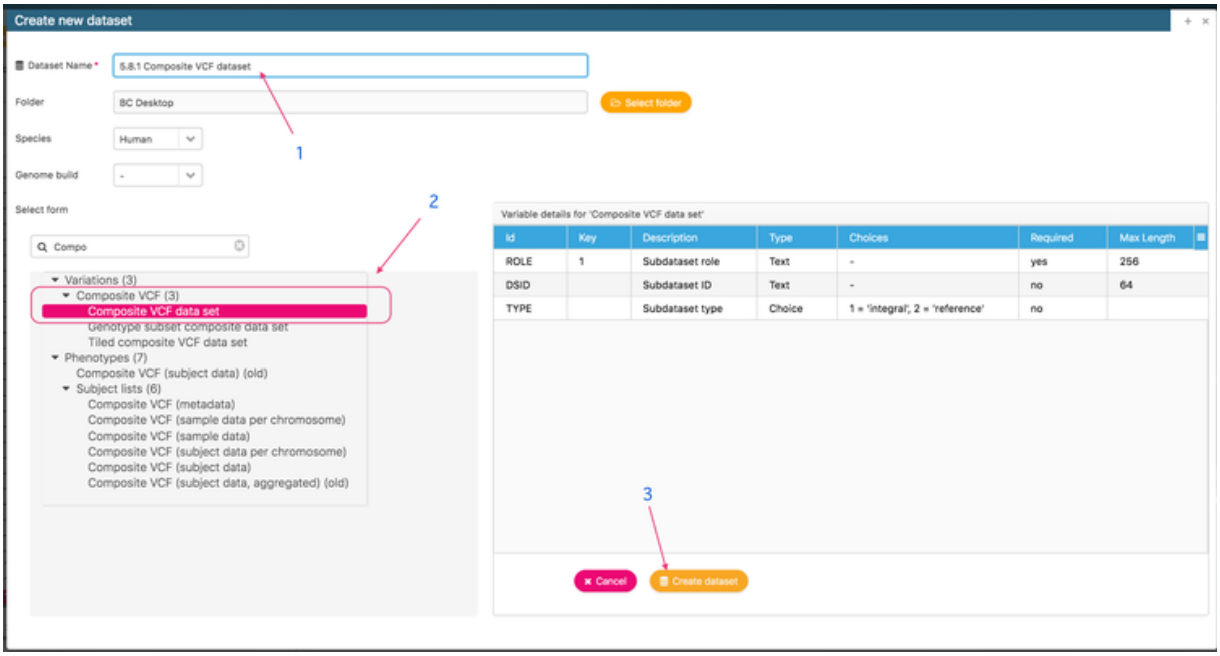
  - Select the ID conversion data file you saved previously and select "Use data import wizard: 'no'" and click "Upload file":



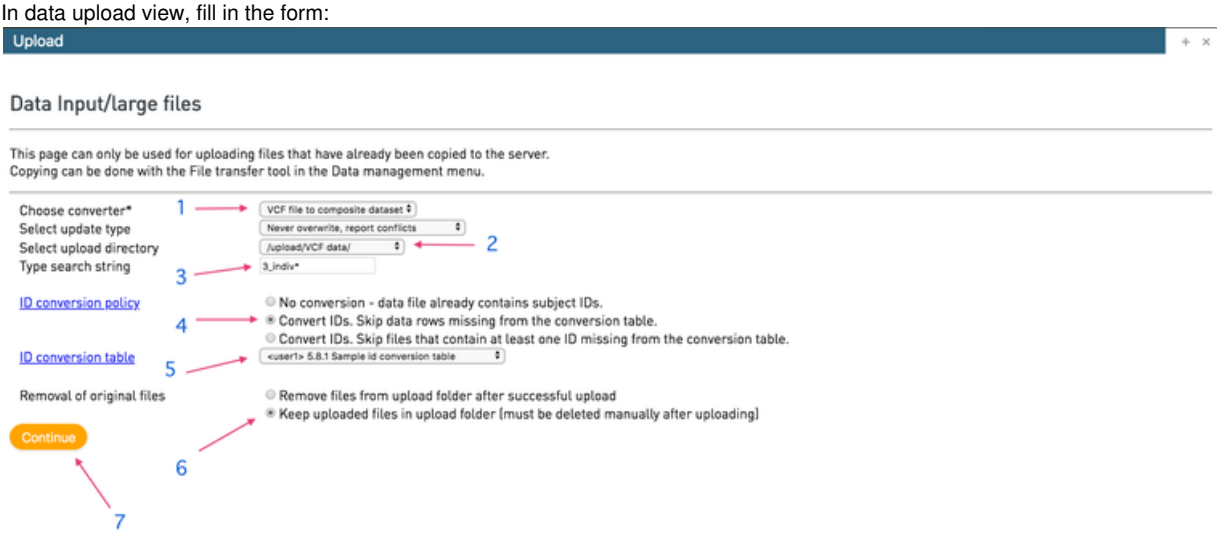
- Wait until ids are uploaded to id conversion table dataset

VCF data upload and sample ID conversion

- Create Composite VCF dataset from "New dataset"-button in the BC|INSIGHT main page.
- In the New dataset dialog:
  - Write name for the composite VCF dataset
  - Select "Composite VCF data set" as the form, it is in the "Composite VCF" -folder
  - Click "Create dataset"



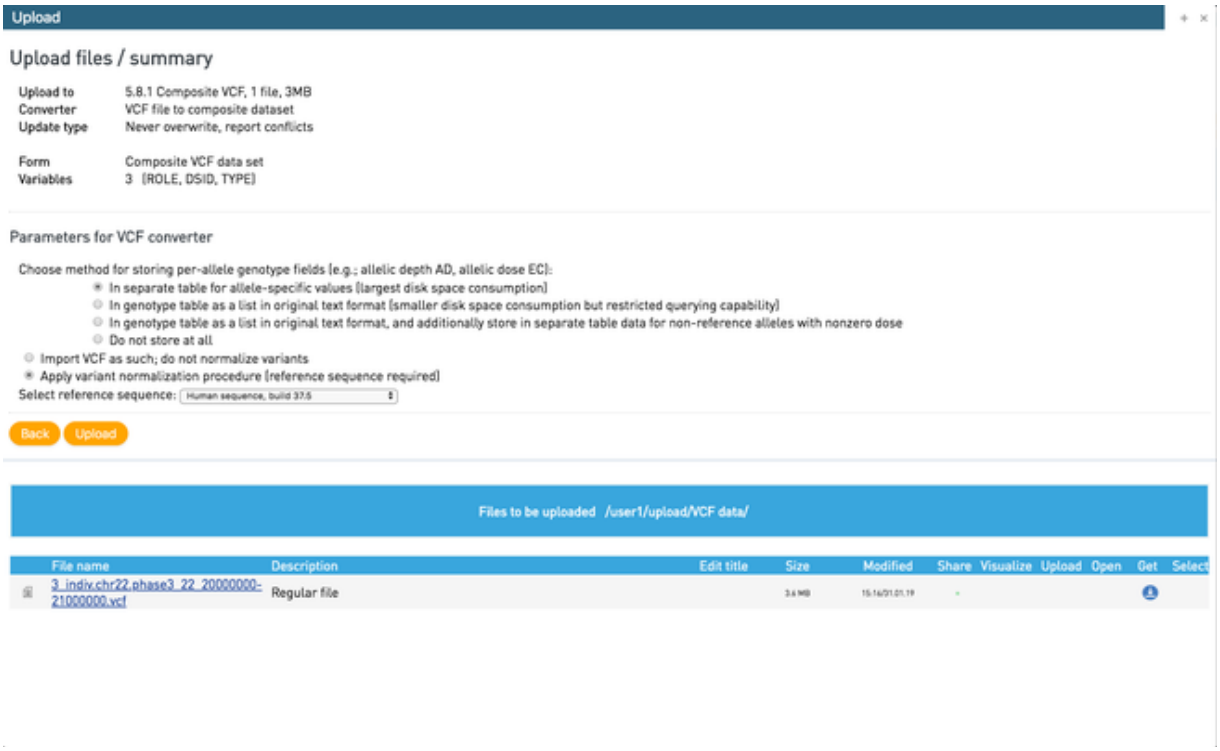
- Import your VCF file, like "example.vcf"
  - In case of a large file, copy it to using the "File transfer" tool to your upload folder
  - From the Composite VCF dataset DATA tab select "Tools and export" Upload File already copied to server
  - In data upload view, fill in the form:



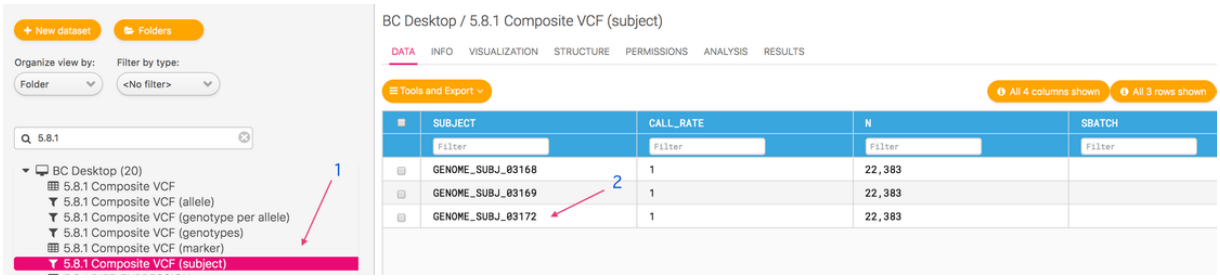
1. VCF file to composite dataset
2. Select upload directory

3. type the data file name or prefix + '*' (asterix)
4. Convert IDs. Skip data rows missing from the conversion table.
5. Select the sample id conversion table created in the previous chapter
6. Keep uploaded files in upload folder
7. Click "Continue"

- In the next view select to upload with or without normalization:



- Click "Upload"
- Go to following the "Queue" link and when the upload job is done go to the DATA tab.



1. Select the "subject" sub dataset from your VCF composite
2. Check that the sample ID conversions to subject ID follow the mapping in the sample ID conversion table

BC|INSIGHT - 3.6.8 Revert accidental changes

User roles
Data manager

In BC|INSIGHT there is no direct 'Undo' or rollback method to revert erroneous changes. If you find you have accidentally changed data and you wish to revert the change, you have two options to go about this, depending on the scale of change.

Undo data entry change of a single data row

When you find there is a need to revert changes in a single data entry row of the dataset stored in the database table proceed with the following steps:

1. Select a dataset from the dataset navigator
2. Navigate to INFO of your dataset and browse for the section of Latest data updates.
3. Click the source link open to verify the changes based on jobID and timestamp
4. In the DATA tab select the data row needing editing
5. Use the View / edit button to edit values
6. Save the changes using the Save button

Revoking multiple data entry rows in a dataset

If you have accidentally uploaded wrong data into your dataset, contact support@bcplatforms.com for their support of reverting the dataset into its original stage. Please indicate your user name, dataset name and jobID found in DATA MANAGEMENT / RESULT ARCHIVE. BC|INSIGHT stores full change log in the database for datasets, which is used to roll back changes. The retraction of changes will also become visible in the change log.

BC|INSIGHT - 3.7 Genomic data management

User roles
Data manager

Child pages

- BC|INSIGHT - 3.7.1 VCF data management
- BC|INSIGHT - 3.7.2 Composite VCF in SQL structure
- BC|INSIGHT - 3.7.3 Tiled composite VCF

Table of contents

- Genomic data
- SNP data
- Imputed genotypes
- Sequenced genotypes

Genomic data

This chapter aims to help the user to understand the various ways of storing genomic data in BC|INSIGHT. Genomic data can be high in volume, consuming disk space, and provides potentially millions of data points per individual subject. The generation of the genomic data can have significant effect to the content and quality of the subject-level data, and to the structure of the overall dataset of multiple subjects. These characteristics can make storing and analysing genomic data challenging. BC|INSIGHT offers inbuilt structures for storing genomic data from different kinds of production pipelines. Depending on the total volume of data, and the specific applications for the data, the data managers should decide on the optimal storage technology.

A short introduction to commonly used data formats and genomic data production processes is needed to fully understand, how these characteristics may affect the choices for genomic data management.

SNP data

SNP (Single nucleotide polymorphism) arrays are commonly used for quick and economical genotyping. Typically SNP array manufacturers design the composition of SNP markers to cover specific portions of the individual genome, often concentrating on specific highly diverse areas, or otherwise targeting markers with well known associations with health problems and responsiveness to certain drugs. Many SNP arrays provide room for thousands of customisable markers, specified by the customer. SNP genotypes are therefore often patchy in giving information, and interpretation of genotypes' consequences is based on association, rather than causality. For many SNP array projects the logical next step is often imputation of the genotypes.

Data integrity requires careful documentation of the SNP manifest for each cohort genotyped using the same array. SNP genotyping ensures that all individuals have had relatively high confidence probing of well documented list of markers, and any quality problems are quickly spotted.

Typically a very small percentage of genotypes produced this way are set to missing per each individual, due to quality problems, or in some cases deletions in the individual's genome. SNP array data is produced often in manufacture's own format, usually convertible into flat text files or VCF format. Recommended storage type is usually **Composite VCF dataset or ACGT dataset**.

Imputed genotypes

SNPs in human genome can be used to predict other known variants in the same genomic region. These predictions are based on known lists of variants (like 1000 Genomes project) and the assumed genomic haplotype of the individual. Commonly known imputing algorithms are for example IMPUTE, and Minimac. Typically an imputation process includes large number of individuals being analysed at the same time (thousands). The resulting dataset of the predicted genotypes is homogenous within that group of individuals, in the sense that all subjects in the dataset have a probability genotype for the exact same list of genomic markers. This allows imputed data to be stored in highly compressed manner (structure of data is known in detail), and makes it possible to use statistical shortcuts in filtering and analysing the data, because there is no need to consider deviances for uncommon markers or alleles, present only in a small number of individuals.

In BC|INSIGHT it is possible to upload imputed data as VCF and CHIAMO files, and the recommended storage type is **Tiled VCF dataset**.

Sequenced genotypes

In comparison to SNP-based genotyping methods, sequencing technologies read the whole nucleotide composition of the genomic region of interest. Sequencing can be targeted to a panel of genes, to the exome, or cover the whole genome. Consequentially the volume of data from these different targeting strategies are significantly different, and have to be taken into account in their storage. Targeted and exome sequencing tend to produce moderate amounts of data, and with thousands of individuals the information is still easily managed in a **Composite VCF dataset** . Whole genome sequence data is considerably larger in volume, and **Tiled VCF dataset** is recommended for storage even with relatively few individuals (hundreds).

Sequencing technology reads all nucleotides within the scope of the used library, thus increasing significantly the likelihood of rare variants or multiple alleles being observed amongst large population of individuals. This makes it challenging to create scalable compression and statistical methods for the genotype data, still allowing for a very fast read access, as each new individual is potentially carrying previously unseen variants. It is therefore highly recommended to introduce new individuals in the datasets in larger batches, so that compression and calculation of internal statistics of the genotypes is as least disruptive as possible.

BC|INSIGHT - 3.7.1 VCF data management

User roles
Data manager

Table of contents:

- Composite datasets explained
  - VCF file format
  - VCF files for import
  - Variant normalisation
  - Export VCF Data

Composite datasets explained

In BC|INSIGHT VCF format data can be stored in a composite dataset. The name 'composite' describes the way how VCF data content is split into more accessible and readable sub-compartments within the dataset, allowing easier filtering for subjects, markers, and alleles. **Composite dataset can store VCF data in either SQL tables, or in Tiled file structure**. In order to create pure SQL dataset, the system form "Composite VCF data set" is used. To create a Tiled dataset, the system form "Tiled composite VCF data set" is used. The SQL storage works well when the final dataset contains less than ten billion genotypes (refers to 100 - 200 full human genomes). In the case of whole genome variation data or large number of subjects with exome sequence or imputed data the Tiled dataset should be used.

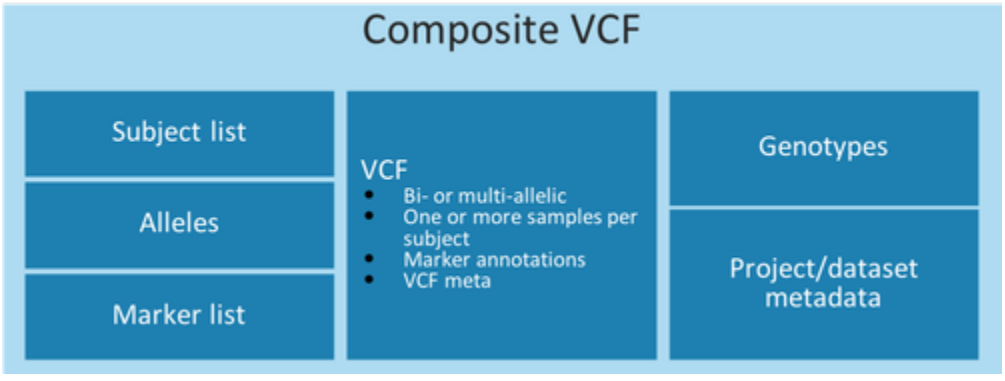


Figure 1. Schematic presentation of concept of Composite VCF dataset.

VCF file format

A VCF file represents variants in which an individual's genome differs from a reference genome. In addition, it contains meta information about variants. You can familiarise yourself with the VCF file content in the blog post of https://gatkforums.broadinstitute.org/gatk/discussion/1268/what-is-a-vcf-and-how-should-i-interpret-it. The composite VCF support in BC|INSIGHT has been created to optimise database disk consumption, improve upload speed and allow filtering of both variants and variants' meta data efficiently: Instead of storing all data to a single database table the composite data upload creates one composite dataset with subsequent datasets or dynamic views, i.e. subsets. Since a VCF file content may vary, the forms for storing VCF information are generated during the upload process in BC|INSIGHT to match to provided VCF information. It is therefore important to keep the same VCF format for all input data to the same dataset.

BC|INSIGHT currently supports VCF specification version 4.3.

VCF files for import

BC|INSIGHT system supports upload of VCF files that adhere to the following rules

- all variants stored in one vcf file either in the vcf or vcf.gz format,

or

- per chromosome vcf files where all subjects are in the same order in all the files.

If your files are in some other VCF format than above please contact BC support for assistance in reformatting your files.

Variant normalisation

The same underlying genomic variant can be represented in many different ways across different VCF files. Thus, it is mandatory to normalise the representation of the variant in order to determine when two representations are the same or different variants. More information about normalisation can be found in https://genome.sph.umich.edu/wiki/Variant_Normalization.

BC|INSIGHT composite upload workflow offer the possibility to normalise variant information. The workflow utilises the left-normalisation method: The start position of a variant is shifted to the left until it is no longer possible to do so, the smaller the number, the better. In BC|INSIGHT in-house developed normalisation procedure has been applied but it obeys principles of VT – Variant Tool (http://genome.sph.umich.edu/wiki/Vt *). Only normalised allele information can be used to annotate alleles against normalised reference information.

Export VCF Data

You can export data out from Composite VCF dataset in VCF format by choosing 'Tools and export' 'Export in VCF format'. You can further specify the subject IDs and markers you wich to include in the resulting VCF. The export job runs through the queue system and you will receive the files in the results folder.

You can specify subjects as simple text list of IDs, or by selecting another dataset containing the subjects. You can filter markers also by simple list, or by Gene name. VCF export provides also statistics-based filtering including by MAF and other frequency thresholds, but also by variant type.

BC|INSIGHT - 3.7.2 Composite VCF in SQL structure

User roles
Data manager

Table of contents:

- Composite VCF data set (SQL)
  - Genotype per allele
  - Genotypes
  - Upload workflow

Composite VCF data set (SQL)

Uploading VCF data into an SQL composite dataset consist of a few steps, which are explained in more detail later in this document. Shortly:

1. Transfer your VCF files to an upload folder
   a. It is recommended to create a dedicated folder for the VCF files going to the same dataset
2. Create a dataset using Composite VCF data set -form
3. In the data upload tool, specify the upload and genotype normalisation options
4. Submit upload job to queue

5. Check results for report on upload events

> **Note**
> You can transfer either an uncompressed or compressed file. If the file is compressed, it must be in the format ***.gz***.

Table below gives you a summary on datasets created upon the composite VCF upload process, based on the data content of the VCF files. The table gives examples of tools available for each sub-table in the composite dataset.

Composite table	Subdataset name	BC|INSIGHT tools available
Composite VCF		• VCF data upload • NGS analysis tools (e.g. Variant tools, SNPeff) • VCF export using the ANALYSIS tab
allele	allele_ext	• Annotating allele information using e.g. VEP and EXAC information
genotype per allele	geno_allele_ext	• stores information on annotated (e.g. using EXAC and VEP information) variants of subject, supports extracting *dose* information from variant data
genotypes	geno_ext	• GWAS analysis tools (e.g. PLINK, R) • In-build BC|INSIGHT tools in DATA: Tools and Export / Reporting tools
marker	marker	• Annotated marker info
subject	subj_aggr	• List of all subjects in the composite dataset

Genotype per allele

"*Genotype per allele*" contains key columns (SUBJECT, MARKER, AINDEX) and genotype fields that have values for each allele. It joins information from the "*genotype per allele*", "*Marker*" and "*Allele*" tables: allele label, normalised allele information and mapping information. A use-case is creating a joined view with an annotation dataset using CHROM, GREGION and ALLELE as joining variables in the subset tool. This enables queries like "*Give me all subjects with a variant annotated as LoF in my annotation database*".

- In "genotype per allele", SUBJECT and MARKER rows are shown multiplied by the number of distinct alleles.
- AINDEX refers to the allele labels, 0 being reference
- ALLELE refers to the reference and alternate allele of chromosomal position
- DOSE_INT refers to actual genotype

Genotypes

This shows information joined from tables Genotypes, Marker and Allele. Use this view for analyses in Variations.

- Both ALLELE1 and ALLELE2 refer to actual genotype of a subject.
- AINDEX1 is the first allele index; 0 for reference call, 1 for alternative
- AINDEX2 is the second allele index
- ALT_DOSE_INT shows the dose of alternative genotype. In other words, a dose of homozygous reference genotype is 0, and 1 for heterozygote and 2 for homozygous alternative genotype.

Upload workflow

1. Create folder for composite VCF dataset. This eases the management of composite datasets in BC|INSIGHT system.
2. Create composite VCF dataset:
    a. Open New dataset
    b. Select folder created in step 1
    c. Select the Composite VCF data set form
    d. Specify both species and dbSNP build
    e. Click 'OK'
3. Transfer the VCF files from the local machine to BC|INSIGHT
    a. Select 'Tools and resources' 'File transfer'
    b. Create a folder for your VCF files, or select an existing one
    c. Browse the files from the local machine and select them for transfer
4. Upload data from the upload folder to a composite VCF dataset
    a. Select 'Tools and Export' 'Upload' 'Files already copied to server'

b. Choose converter: ‘VCF file to composite dataset’
c. Select update type: ‘Never overwrite, report conflicts’
d. Select upload directory
e. Type search string e.g. *vcf*
f. Sample ID conversion, choose dataset stored in SampleIDs if you have sample – subject ID conversion pair information stored in the SampleIDs tab
g. Removal of original files: If you want to keep the file(s) in the upload folder after the successful upload then ‘Keep..’ else ‘Remove…’
h. Click ‘Continue’
i. Upload files summary
j. Parameters for VCF converter:
    i. Store per-allele genotype fields: Check if (this generates the genotype per allele dataset)
    ii. Apply variant normalisation procedure: Check if you wish that the allele position information, so chromosome, reference allele position and length, is normalised with reference genome and stored. This means that the data can be combined to other annotation information datasets.
    iii. Select reference sequence – only applies if you choose do normalize. If there are no references sequences shown in the drop-down menu, you must contact support@bcplatforms.com to provide the correct reference files. You can specify one of the following normalizing files:
        1. Human sequence, build 37.5
        2. Human sequence, build 37.5 (1000G version)
        3. Human sequence UCSC hg195
k. Click 'Upload'

> **Note**
> Select the reference file that matches your VCF file information. If your VCF file has chromosome IDs prefixed with ***chr*** use the **UCSC hg195** file, but if the prefix is missing, use one of the build 37.5 files.

The upload work goes to queue, where it can be followed. Check the result report for any potential issues with the VCF format.

BC|INSIGHT - 3.7.3 Tiled composite VCF

User roles
Data manager

Table of contents:

- Tiled composite VCF data set
    - VCF format prerequisites for upload
    - Upload workflow for Tiled dataset

Tiled composite VCF data set

BC|TILING creates a fast-read filesystem for either imputed genotypes or VCF data. The genotype and variant data is stored in indexed 'tiles' in the file system, where each tile can be read independently from others accelerating both data access and analysis workflows.

Tiles are ordered by both number of subjects and variants. Marker tiles and markers within a tile must be ordered according to the genome build. Subject order does not matter. Data stored in tiles significantly speeds up read-access to large collections of genotype or VCF data enabling efficient and flexible data analysis and result collecting. Storing data in tiles is optimal for population data where the number of genotypes exceeds ten billion (100-200 full human genomes).

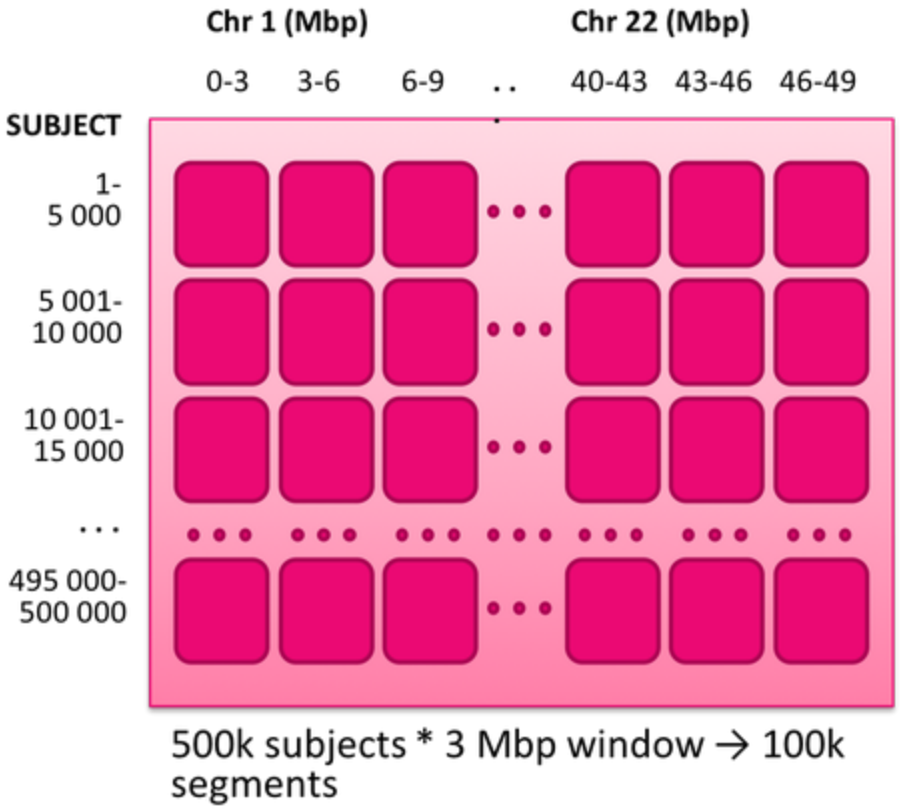


Figure 2. Representation of marker – subject tiles.

Storage saving with a tiled dataset, in comparison to storage in SQL structure, is linear to the number of markers / subjects comparing to data stored in the SQL table: When data of 10 000 subjects and 100 million variants totalling of 1 000 billion data rows is stored to a SQL table 88 000 GB will be consumed while tiled data consumes about from 127 GB to 375 GB depending on the input data format. The storage saving has been achieved by storing variant information into the BC file system (BCFS) instead of all data stored in a single SQL table.

VCF format prerequisites for upload

Before uploading your VCF files please check the following prerequisites:

1. All VCF variant data is called in one batch
2. You have chromosome specific VCF files
3. All subjects are included in all VCF files
4. Order of subjects is the same in all VCF files

If you have any questions considering your VCF file format, or you need help with data conversion please contact suppport@bcplatforms.com.

Upload workflow for Tiled dataset

This section describes the steps needed to store VCF files into the tiled dataset. As is the case with the composite datasets also in the tiled upload process BC|INSIGHT auto-generates forms matching to your VCF meta-information content. It is therefore important to make sure all VCF files follow the same conventions.

As default, VCF files need to be transferred from user's local computer to user's BC|INSIGHT upload folder before proceeding to the data upload step. If you have a file system mounted to your upload contact BC support for details if needed. However, before starting the VCF file uploads phase check your folder only contains the necessary VCF files to ease the upload effort.

> **Note**
> You can transfer either an uncompressed or compressed file. If the file is compressed, it must be in the format ***.gz***.

1. Create the tiled dataset
    a. Open 'New dataset' and select 'Tiled composite VCF data set' form
    b. Specify dataset name

c. Select folder
d. Select both Species and Genome build from the Menu
e. Click 'Create dataset'

2. Transfer of VCF files to upload folder
   a. Select ‘Tools and resources’ 'File transfer'
   b. Create a folder for your VCF files, or select an existing one
   c. Browse the files from the local machine and select them for transfer
3. Upload data from upload folder to dataset
   a. Check you have the tiled VCF dataset open
   b. Select 'Tools and export' 'Upload' 'Files already copied to server'
   c. Choose converter: ‘VCF files’
   d. Select update type: ‘Never overwrite, report conflicts’
   e. Select upload directory
   f. Type search string e.g. **vcf* to identify your all vcf files in an upload folder
   g. Click ‘Continue’
   h. Apply variant normalisation procedure by selecting the reference sequence
   i. Select reference sequence – only applies if you choose do normalize. If there are no references sequences shown in the drop-down menu, you must contact support@bcplatforms.com to provide the correct reference files. You can specify one of the following normalizing files:
      i. Human sequence, build 37.5
      ii. Human sequence, build 37.5 (1000G version)
      iii. Human sequence UCSC hg195
   j. Click 'Upload'

> **Note**
> Select the reference file that matches your VCF file information. If your VCF file has chromosome IDs prefixed with ***chr*** use the **UCSC hg195** file, but if the prefix is missing, use one of the build 37.5 files.

The upload work goes to queue, where it can be followed. Check the result report for any potential issues with the VCF format. The report typically contains information about inserted subjects and markers.

BC|INSIGHT - 4. Analysis and tools

User roles
Analyst

Child pages:

- BC|INSIGHT - 4.1 Visualising distribution of data values
- BC|INSIGHT - 4.2 Conversions and reports
  - BC|INSIGHT - 4.2.1 Pivot datasets
  - BC|INSIGHT - 4.2.2 Aggregate statistics
  - BC|INSIGHT - 4.2.3 Reports
- BC|INSIGHT - 4.3 Running embedded analyses
  - BC|INSIGHT - 4.3.1 Queue system
- BC|INSIGHT - 4.4 Analysis results
  - BC|INSIGHT - 4.4.1 Results content
  - BC|INSIGHT - 4.4.2 Uploading results to database
- BC|INSIGHT - 4.5 R script interface
  - BC|INSIGHT - 4.5.1 R Data input
    - BC|INSIGHT - 4.5.1.1 R Genotypes
    - BC|INSIGHT - 4.5.1.2 R Imputed data
    - BC|INSIGHT - 4.5.1.3 R Omics and multiQTL data
    - BC|INSIGHT - 4.5.1.4 R Phenotypes
  - BC|INSIGHT - 4.5.2 R script Data output
  - BC|INSIGHT - 4.5.3 Storing and sharing R scripts
    - BC|INSIGHT - 4.5.3.1 External R libraries
  - BC|INSIGHT - 4.5.4 R script Examples
- BC|INSIGHT - 4.6 Genome browsers
  - BC|INSIGHT - 4.6.1 LocusZoom
  - BC|INSIGHT - 4.6.2 Manhattan and QQ plots
  - BC|INSIGHT - 4.6.3 UCSC genome browser

- BC|INSIGHT - 4.6.4 Embedded IGV
- BC|INSIGHT - 4.6.5 Data service for IGV and LocusZoom
- BC|INSIGHT - 4.7 Embedded analysis tools

Principles of data analysis in BC|INSIGHT

The BC|INSIGHT Data Warehouse comes equipped with visualisation tools for different types of data (genetic, statistical, omics), and with some inbuilt tools for simple statistics about you data, like cross-tabulation, marker allele frequencies, phenotypic filtering linked to genotypes, and so on. These BC Platforms considers household items that should be available for all data managers on BC|INSIGHT. The Data Warehouse provides also interfaces to various external tools like scripts, and third-party Notebook applications, which will effectively require some proficiency in common coding and scripting languages typically used for Data Science tasks. On the other hand the script interface, and external tools are great for creating shared in-house analytics collections for less coding-savvy people in the projects.

In addition, BC Platforms maintains a short-list of available packages that are deemed appropriate to the typical data volumes and workflows amongst BC|INSIGHT users. It is, however, possible to request additions of new packages and tools by contacting either BC support or the Sales organisation.

Many operations in the BC|INSIGHT produce result files. These are simple folders of data and reports about the tasks being done, and what the end results were. Data uploads and large exports leave their reports and results in the Report -page of the user, and most analytical processes drop their output and log files there as well. The results can be shared between users, uploaded to dedicated datasets for further examination, or visualised on the spot.

BC|INSIGHT - 4.1 Visualising distribution of data values

User roles
BC|INSIGHT user

Child pages:

Table of contents:

- Generic visualisation of data
- Heatmaps

Generic visualisation of data

BC|INSIGHT provides various ways to visualise your data points in a dataset. These tools can be accessed in the ***VISUALIZATON*** page when a dataset has been selected in the data navigator. The available graph types (Comparison, Trend, Correlation), depend on the type of the selected dataset.

In charts, the maximum number of different data points is 1 million. If you try to visualise more than one million data point values, or the query takes more than 60 seconds, a timeout message is shown.

In the pie graph, up to ten sectors are used to visualise the data and the rest of the values are concatenated into a sector named **Others**.

> **Note**
> Not all graph types may be available. The options depend on the dataset chosen.

You can start by selecting a chart type, and the columns you wish to use to set the X and Y values, or grouping, depending on the type of chart. Some chart types give you more control over the display of the data, than others.

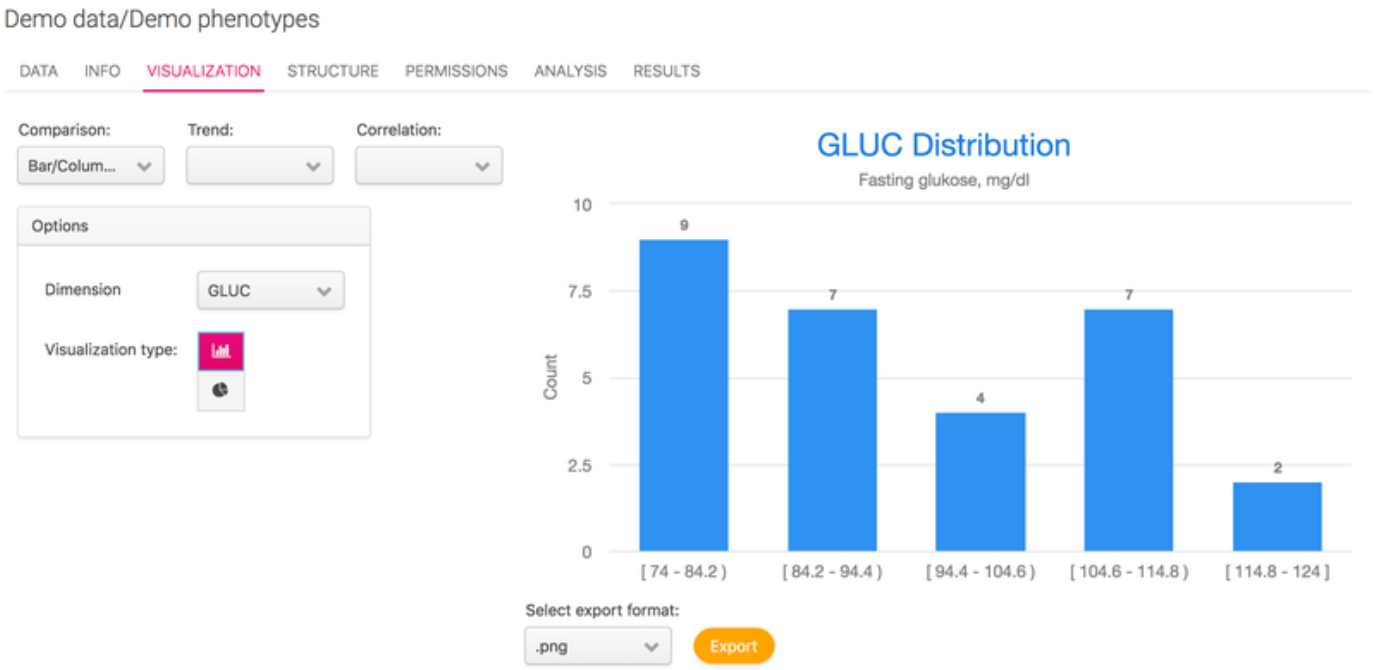
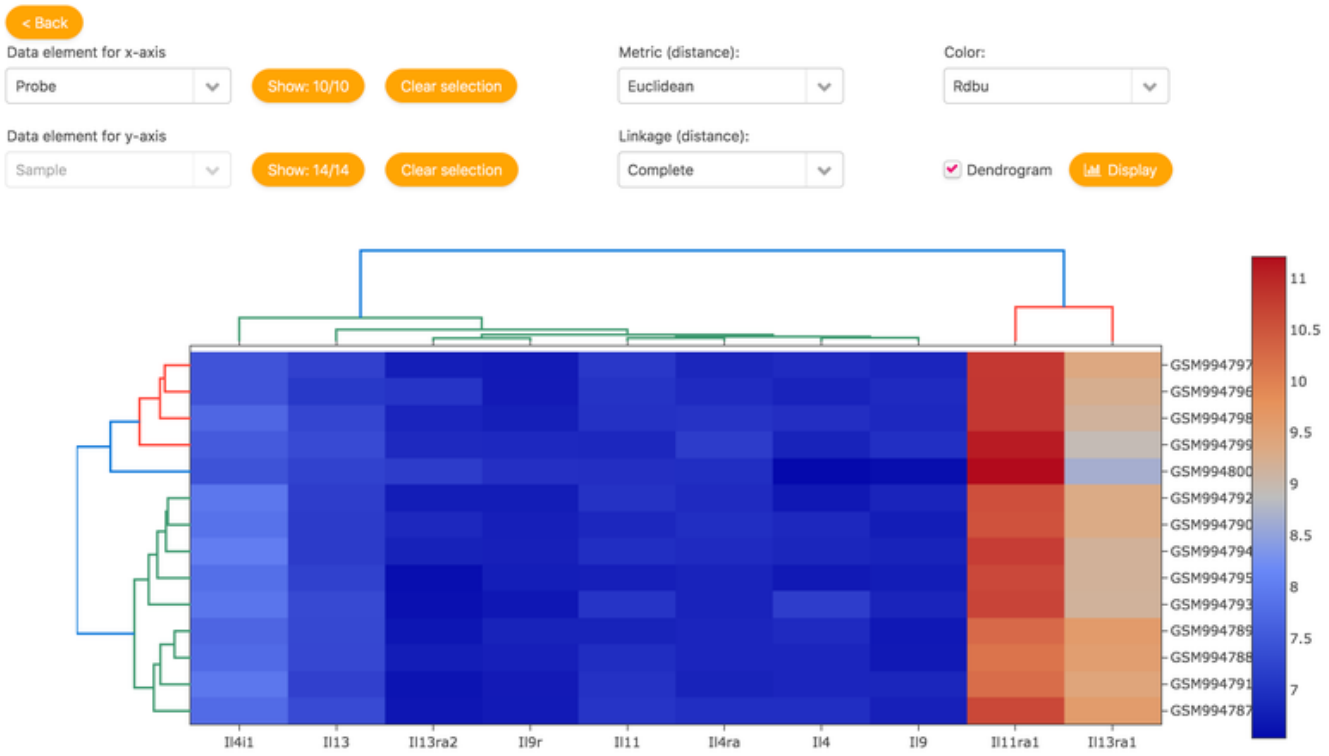


Image 1. In the example above the user is creating a Bar chart of Fasting glucose distribution, which is automatically binned into value ranges.

You can download your generated chart image as PNG, JPG, SVG or PDF using the Export feature under the chart.

Heatmaps

In the current version heatmap visualisation of data is available only for differential expression data. This will change in the future versions. The heatmap is defined as vectors of samples/subjects (y-axis), and observed variables (x-axis, typically genes or probes in expression data). These observed variables then provide values for the heatmap, which can be grouped using various inbuilt choices for metric distance, and linkage.



User roles

Analyst

Child pages:

- BC|INSIGHT - 4.2.1 Pivot datasets
- BC|INSIGHT - 4.2.2 Aggregate statistics
- BC|INSIGHT - 4.2.3 Reports

Views into data

In many situations it is beneficial or even necessary to look at the data from a different angle, or in different format. BC|INSIGHT provides tools for transformation of data into various formats, also structurally, and tools to create more detailed reports about specific data items, and their relationship to information in other datasets. Some of these tools are for very specific use-cases, and some are more generically appliable to different situations.

BC|INSIGHT - 4.2.1 Pivot datasets

User roles

Analyst

Table of contents:

- Restructuring phenotype table
  - Pivot
  - Unpivot

Restructuring phenotype table

Pivot and Unpivot -tools are only available for phenotype tables with subject data, where a subject identifier field has been defined by using *BC: subject* column annotation.

Pivot

The pivot functionality translates tables into a "wide" format in which the values of the key columns the user selects are transposed and become columns in the resulting view. In other words, the pivot tool rotates a table-valued expression by switching the row values to multiple columns in the subset table.

Original data:			**Pivoted on VAR:**		
SUBJECT	*VAR*	*VALUE*	*SUBJECT*	**X**	**Y**
S1	**X**	1	**S1**	1	2
S1	**Y**	2	**S2**	3	4
S2	**X**	3			
S2	**Y**	4			

Table 1. An example in principle mechanism of pivot -tool.

SUBJECT	DATE	VARIABLE	VALUE
NA10839	2000-04-05	VAR_X	20.454985
NA10839	2002-12-04	VAR_Y	-0.043008
NA12003	2000-07-03	GLUC	135.232653
NA12003	2002-02-10	TRIG	310.235178
NA12004	2004-06-13	GLUC	133.374616
NA12004	2001-03-03	VAR_Y	0.652801
NA12004	2004-08-14	GLUC	140.744509
NA12005	2013-01-24	TRIG	71.695676
NA12005	2005-09-24	GLUC	101.690309
NA12006	2004-06-12	HDL	30.184064

SUBJECT	DATE	VAR_X	VAR_Y	VAR_Z
NA07034	2000-01-04		-0.228287	
NA07345	2000-01-04			
NA07034	2000-01-05			
NA07022	2000-01-15			
NA10838	2000-01-19			
NA12005	2000-01-24	16.106099		
NA07022	2000-02-02			
NA07034	2000-02-02		-0.052792	
NA10830	2000-02-11		-0.40627	
NA06985	2000-02-12	20.694231		

Image 1. In the data table above, SUBJECT, DATE and VARIABLE are the 1 st , 2 nd and 3 rd keys, respectively and the dataset has been pivoted on variable values VAR_X, VAR_Y and VAR_Z:

Tip: The pivot tool can also be used for transposing any column values to columns with expressions from the specified columns, e.g. multi-choice values.

Caution! When you use multiple choice variable types as pivoting variables, make sure that the choice descriptions only contain letters from A to Z in a combination of upper and lower case letters, numbers from 0 to 9 as well as underscores and empty spaces.

You start pivoting data from the DATA -page of your dataset. Choose ***Tools and Export Dataset tools Pivot***. The dimensions available for pivoting are the text and multi-choice fields in the dataset. Use preview to see the results of your choices in the dialog.

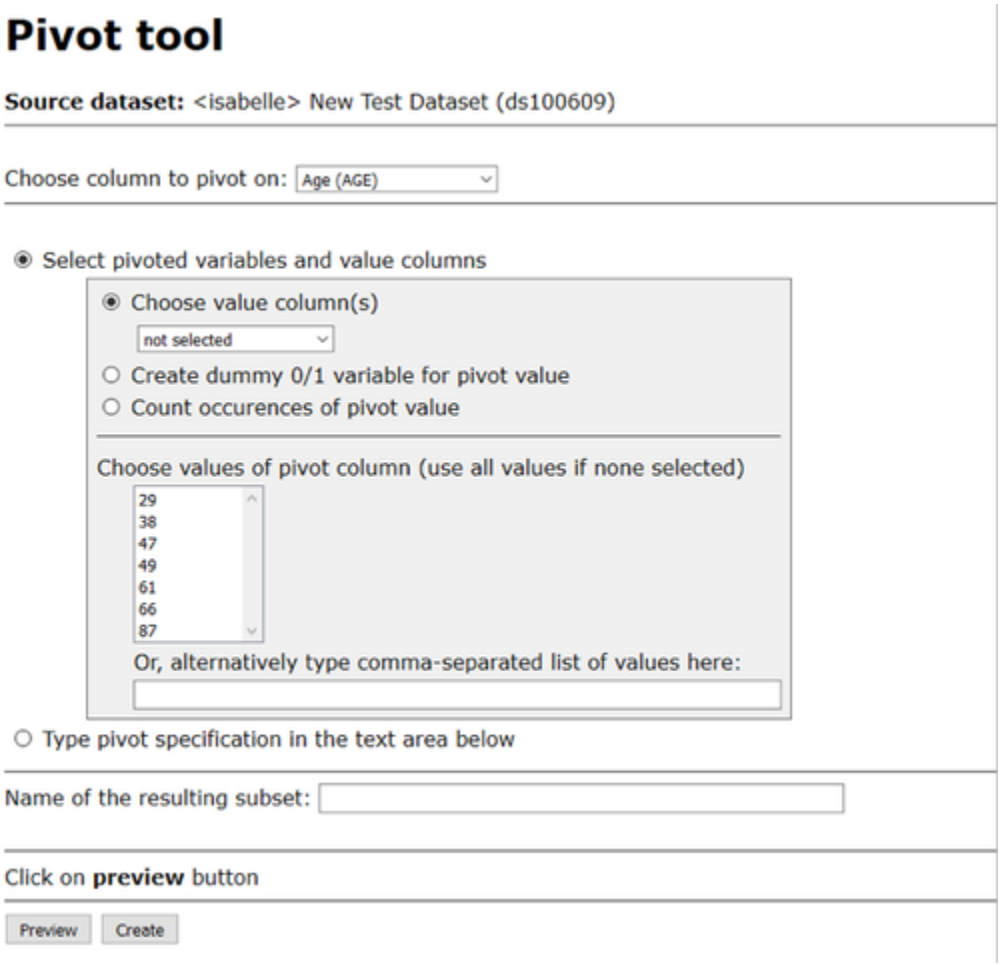


Image 2. The Pivot tool dialog where user chooses the column to pivot on, and then the other dimensions.

In the "Choose the column to pivot on" specify the variable for which the values should be displayed as columns. **Note:** The available variables are displayed in alphabetical order, and the first in the list is automatically displayed when you open the Pivot tool (in this example it happens to be the required one).

Then select the option Select pivoted variables and value columns, and then select the required options:

- Choose value column(s): Select the variables for which the values will be shown in the pivoted table.
- Create dummy 0/1 variable for pivot value: Specify 0 (=no) or 1 (=yes) if the values of the pivoted variable exist.
- Count occurrences of pivot value for showing a number of occurrence of subject ID specific column value (when longitudinal data entries)

Choose values of pivot columns (use all values if none selected) for specifying one or more values of the pivoted column by selecting the keyboard combination of CTRL + click. Alternatively, values can be specified as a comma-separated list.

> **Note**
> Please contact BC support regarding the option - our developers are happy to help you to create the required pivot expression

Name of the resulting subset for specifying the unique pivot subset name.

Content preview:

PATIENTID	N29	N38	N47	N49	N61	N66	N87
1111		65					
2222	72						
3333			68				
4444						72	
5555		76					
6666						60	
7777					85		
8888							62
9999				95			

Image 3. The preview shows the AGE in the example as columns. Note the preceding 'N' character, which is simply to bypass the SQL requirement for column names to start with an alphabet. The weights are then listed in the pivot table.

If the query results displayed meet your expectations, click the Create button to create the pivot subset for further data management tasks. The system displays a confirmation message, and you can open the newly created dataset.

Unpivot

The unpivot functionality translates tables into a "wide" format in which the values of the columns the user selects are transposed and become rows in the resulting view. In other words, the pivot tool rotates a table-valued expression by switching the column values to multiple rows in the subset table.

Original data:			**Unpivoted on VAR:**		
SUBJECT	**X**	**Y**	*SUBJECT*	*VAR*	*VALUE*
S1	1	2	**S1**	**X**	1
S2	3	4	**S1**	**Y**	2
			S2	**X**	3
			S2	**Y**	4

Table 2. An example in principle mechanism of unpivot -tool.

SUBJECT	DATE	VALUE_VARIABLE_CHOL	VALUE_VARIABLE_GLUC
NA06985	2000-02-12		
NA06985	2000-09-03		
NA06985	2001-01-07		
NA06985	2002-12-05		
NA06985	2004-02-13		
NA06985	2004-11-05		
NA06985	2006-11-05		
NA06985	2006-11-27		
NA06985	2007-03-04	192.602604	
NA06985	2007-11-08		

SUBJECT	DATE	VAR	VALUE
NA07345	2000-01-04	VALUE_VARIABLE_CHOL	246.550002
NA07034	2000-01-04	VALUE_VARIABLE_CHOL	
NA07034	2000-01-04	VALUE_VARIABLE_GLUC	
NA07345	2000-01-04	VALUE_VARIABLE_GLUC	
NA07034	2000-01-05	VALUE_VARIABLE_GLUC	86.642041
NA07034	2000-01-05	VALUE_VARIABLE_CHOL	
NA07022	2000-01-15	VALUE_VARIABLE_CHOL	
NA07022	2000-01-15	VALUE_VARIABLE_GLUC	
NA10838	2000-01-19	VALUE_VARIABLE_CHOL	
NA10838	2000-01-19	VALUE_VARIABLE_GLUC	
NA12005	2000-01-24	VALUE_VARIABLE_CHOL	

Image 4. Unpivoting. In the example a longitudinal table of measurements is transformed into VARIABLE/VALUE pairs.

You can access the Unpivot tool from the same place as pivot: ***Tools and Export Dataset tools*** *Unpivot*. Specify the column(s) to be unpivoted in the 'Choose column to be unpivoted (unpivot all if none selected)' area.

Tip: Use the key combination CTRL+ click to select or unselect columns.

Choose extra columns to be included for specifying variable(s) shown as originally in the unpivoted subset. Specify the name of the generated pivot variable ID and the value column ID, which will contain the data from the column(s) that are unpivoted. Make sure that the variable IDs only contain letters from A to Z, numbers from 0 to 9, or underscores. Note that by default, you can use column IDs (i.e. variable IDs) with up to **16** characters.

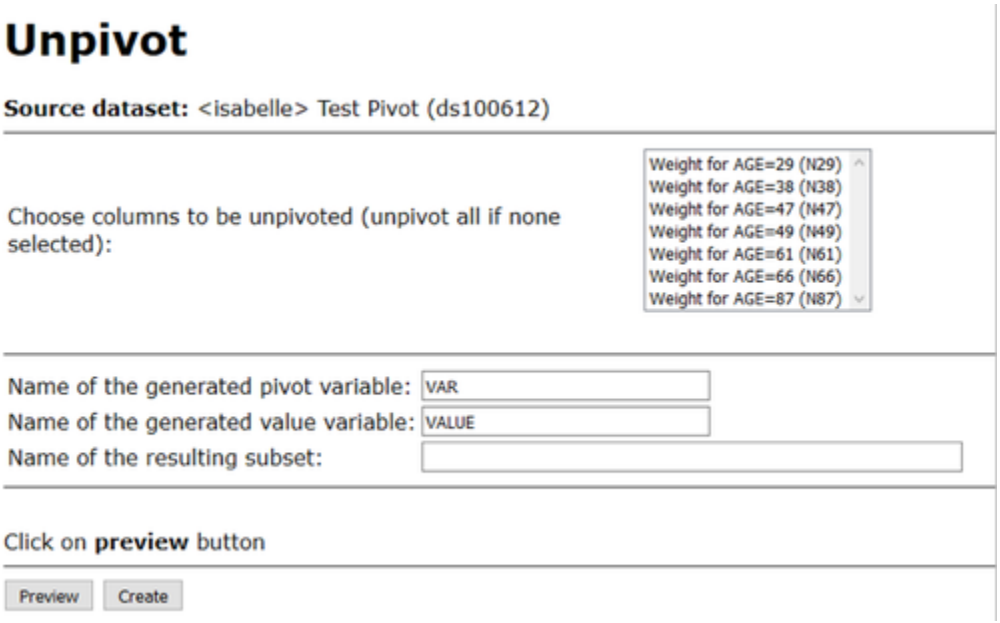


Image 5. Unpivot dialog, where you choose the columns to be transformed, and name the generated new variable and value columns.

Name your transformation subset, and preview the results to make sure all is as it should be.

Content preview: first 10 rows

PATIENTID	AGE	WEIGHT
6666	N66	60
8888	N87	62
1111	N38	65
3333	N47	68
2222	N29	72
4444	N66	72
5555	N38	76
7777	N61	85
9999	N49	95

Image 6. Preview of the table that was unpivoted for age and weight. You can then create the subset, and view it from the Data Navigator.

BC|INSIGHT - 4.2.2 Aggregate statistics

User roles
Analyst

Table of contents:

- Summary statistics of selected fields
  - Using the Aggregate tool
  - Hint - Creating lists

Summary statistics of selected fields

Aggregate -tool is only available for phenotype datasets that contain subject data, where a subject identifier field is defined by *BC:subject* column annotation.

Aggregate tool groups together selected field values based on either a categorical grouping choice (like diagnosis, visit number, or similar categorical value column) or a date column (longitudinal data, see Image 1 for example). User has the choice over the statistics calculated over the values per each group - mean, minimum, and maximum values.

SUBJECT	AVG_VALUE_VARIABLE_CHOL	MIN_VALUE_VARIABLE_CHOL	MAX_VALUE_VARIABLE_CHOL
NA06985	202.278425333333	192.602604	213.93959
NA06991	184.264784666667	153.532204	203.182793
NA06993	209.05477	162.05323	233.083069

Image 1. Example of aggregate statistics from multiple cholesterol value observations per subject.

User can also specify their own SQL functions in the SQL expression column. The different options are described in a separate document applying to the SQL technology used in BC|INSIGHT https://www.ibm.com/support/knowledgecenter/SSEPEK_10.0.0/sqlref/src/tpc /db2z_aggregatefunctionsintro.html.

Using the Aggregate tool

The examples in this document use a dataset illustrated in Image 2.

PATIENTID	GENDER	AGE	HEIGHT	WEIGHT
10	1	38	1.63	65
11	2	29	1.78	72
12	1	47	1.72	68
13	2	66	1.58	72
14	2	68	1.82	76
15	1	81	1.67	60
16	1	61	1.74	85
17	1	87	1.65	62
18	2	49	1.8	95
19	2	38	1.64	72
20	2	29	1.78	80
21	2	47	1.63	55
22	2	66	1.85	90
23	1	38	1.77	68
24	1	29	1.69	65
25	1	47	1.7	81

Image 2. Example dataset used in this chapter, refer to this to understand the generation of aggregate information.

You find the Aggregate -tool in the Dataset Navigator on DATA page of the dataset, in ***Tools and Export Dataset tools Aggregate*** .

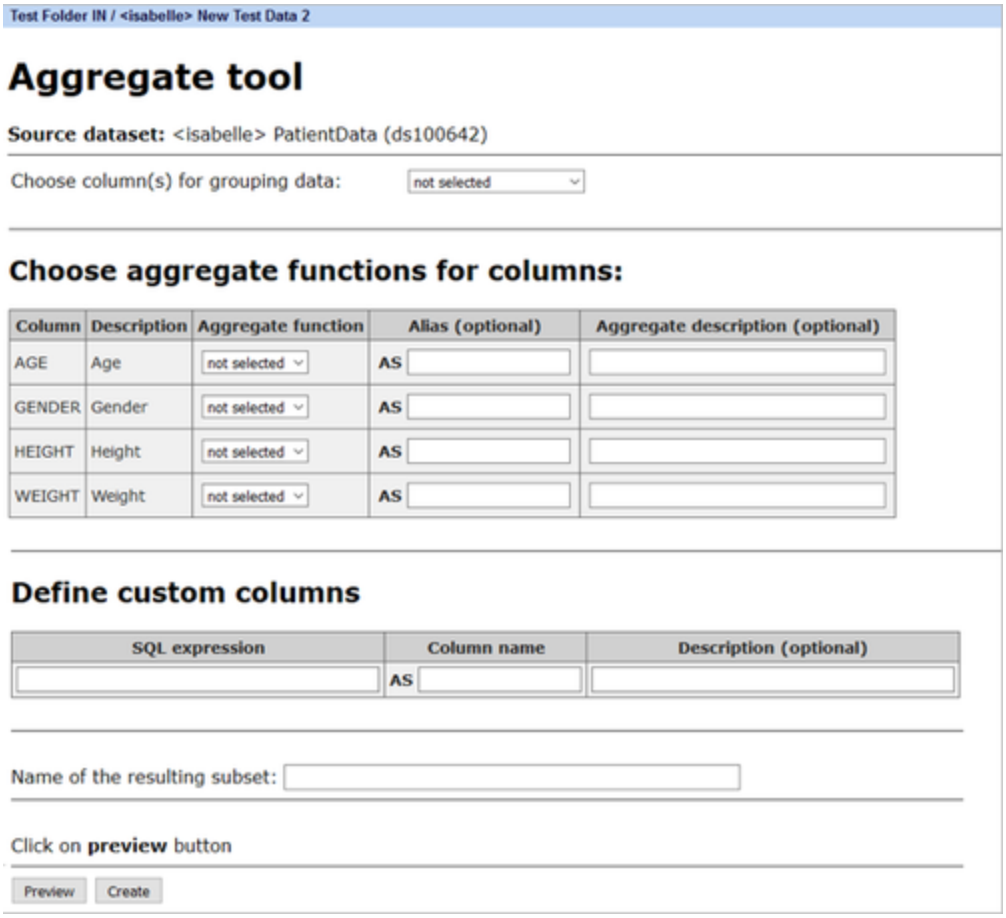


Image 3. After selecting the grouping column (or date/timestamp column) for your statistics, you can choose the fields which are included in the aggregation, and give the new statistical result fields column identifiers.

Select the desired column(s) in the Choose column(s) for grouping data to specify the variable(s) for grouping data in a new subset. Once you have selected a column, the system displays a new field that allows you to select another variable, if it is needed. Enter the settings in the 'Choose aggregate functions for columns' section. In this example, the average height and weight should be determined for each age group.

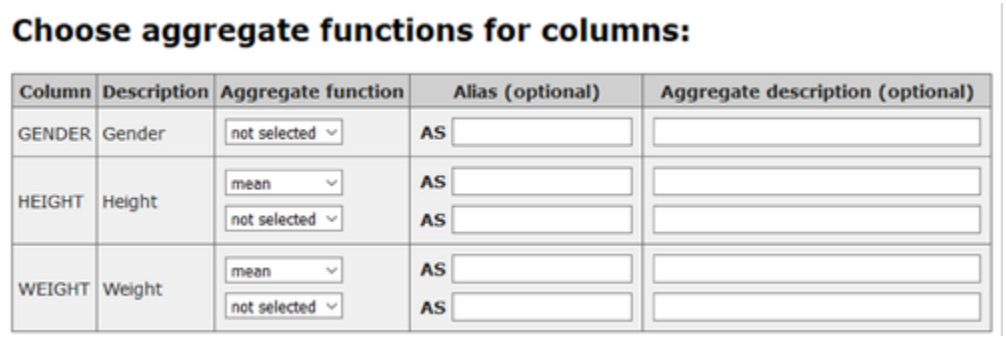


Image 4. Settings to generate mean values for Height and Weight in the example dataset.

You can select to calculate the mean, the minimum, and the maximum. Optionally, you can enter alternative names to the added statistics fields in Alias and Aggregate descriptions -fields. Field names will appear in the final subset in the format <aggregate function><variable>. For example the mean value for height would appear as AVG_HEIGHT. If required, define custom columns for custom aggregating functions (see the document link above). Use unique column names that follow the Variable ID naming convention (Variable IDs only contain letters from A to Z, numbers from 0 to 9 as well as underscores). Note that by default, you can use column names (i.e. Variable IDs) of up to **16** characters.

Name the new subset, and use Preview to check the generated content.

AGE	AVG_HEIGHT	AVG_WEIGHT
29	1.75	72.3333333333333
38	1.68	68.3333333333333
47	1.68333333333333	68
49	1.8	95
61	1.74	85
66	1.715	81
68	1.82	76
81	1.67	60
87	1.65	62

Image 5. The example dataset produces mean values in the defined age groups, as shown.

If you are happy with the results, you can continue to create the new subset, and it is displayed in the Dataset Navigator.

Hint - Creating lists

Aggregate tool can be used to create simple lists of distinct categorical values. If you would need to capture the distinct (unique) categorical values from any dataset, you can use Aggregate tool to group the data by that value column, and leave the other options in the Aggregate generation empty. This will result a single-column subset, i.e. a list of the distinct values.

BC|INSIGHT - 4.2.3 Reports

User roles
Analyst

Table of contents:

- Report tools
  - Text
  - Custom query
  - Statistics - Summary of numeric fields
  - Statistics - Call rates by subject
  - Statistics - Compare to maps
  - Genotype/phenotype cross-tabulation
  - Summary
  - Crosstab all

Report tools

BC|INSIGHT has a collection of little convenience tools that operate in specific data context, or perform a more generic task of collecting and displaying data, according to user's design. All of these tools create either a tabular or graphical view to display the resulting information. Many are also easy to repeat using other tools, like R script, but this collection is in the vicinity of datasets to make convenient short cuts for the user.

If your data volumes are large, some of these tools may speed up filtering and subsetting processes, that might otherwise take considerable amount of time.

You access the report tools from the dataset's DATA page, ***Tools and Export Reports*** menu.

	Tool	Description
All datasets	Text Custom query	Text -tool gives you a table of counts for values in a column. Custom query is free form SQL query interface, which allows you to export the query results as a file. Good understanding of SQL is required.
Genotypes, variants	statistics ->	Genotype specific tools to query call rates and mapping information. Note that you may need to refresh the marker and subject index for your dataset in the INFO -page for these tools to work properly.

	• Summary of numeric fields • Call rates by subject • Compare to maps Genotype /phenotype cross-tabulation	Genotype/phenotype cross-tabulation queries per genotype selected phenotypic attributes for the patients in the genotype dataset.
Phenotypes, Markers	Summary Crosstab all	Summary tools gives you summary statistics across different datasets, by grouping items based on date or text values, and comparing the data between datasets. Crosstab is a cross-table report which gives summaries based on subject grouping.
Omics (MultiQTL)	Summary	Summary tools gives you summary statistics across different datasets, by grouping items based on date or text values, and comparing the data between datasets.

Table 1. The tools under Reports menu can be data type specific. A high-level description for each tool available is provided here.

Text

A text report shows detailed information on a selected question in the dataset that can be filtered either by value or count. Works best for text questions.

- For the **Value** fields either minimum, maximum or both values for filtering values can be given. (For text variables, character-wise comparison is used. Dates should be given in yyyy-mm-dd format.)
- For the **Count** fields either minimum, maximum or both values for filtering counts can be given
- By ticking the **Include missing** box a count of missing values are shown on the last row (otherwise missing values are ignored)
- Press **OK** to view the text report
- The text report can be saved by pressing the **Download as a file** link
- Optionally in the *Genotypes* and *CNV* sections, either the subject or marker counts per occurrence can be viewed by pressing the **Download subject list** and **Download marker list** buttons, respectively

Custom query

This interface allows you to build your own SQL query very freely, and download the results as a file. Access to data depends on you account permissions.

- You can build a SELECT clause
- Add more tables to the query target and define a JOIN strategy
- Apply SQL clause modifiers like WHERE, and GROUP BY

Statistics - Summary of numeric fields

- The report is viewed by the question set in the form
- The *Compare to* field enables the comparison of the dataset with some other dataset that is based on the same form
- Comparison can be made by showing *significant results only*. In this case only statistically significant results are printed:
- N.S.= Statistical significance is not significant ($p > 0.10$)
    - * = Statistical significance is almost significant ($p < 0.05$)
    - ** = Statistical significance is significant ($p < 0.01$)
    - *** = Statistical significance is very significant ($p < 0.001$)
- Summary report displays the symbols of statistical significance in the right top corner of every question
- Pressing the **OK** button displays the report
- The report is able to be printed by pressing the **Print report** button

Statistics - Call rates by subject

- Requires the refreshed index performed in the INFO -page (Clear index -button)
- The report shows both *Missing* and *Total* calls with the *Total SNP calls* derived from the call rates by a subject. Furthermore, a *Percentile* and both the ascending and descending number of subjects are shown
- The report can be saved by copying all cells

Statistics - Compare to maps

Simple tool that takes the genotype or marker dataset, and compares it to any given marker map. It then gives a breakdown of overlap between the too. Used for basic quality control of data.

- Select the map you want to compare to
- The result table appears after the task has run, this may take a while

Genotype/phenotype cross-tabulation

The function of this tool is to compare the break down of genotypes in a marker or markers, based on phenotypic differences between individuals. It gives you allele frequencies for a marker grouped by phenotype.

- Select a genotype dataset and the tool
- Select a phenotype dataset, and from there a phenotype trait (multichoice) which is used to break down the frequency of alleles
- The tool works only for low number of markers, so either provide them as a file, or comma-separate list in the text input
    - You can also have the list of markers picked from another dataset
    - Alternatively you can select exclusion criteria

Summary

The summary function is available in the *Markers*, *MultiQTL*, *Affection*, and *Phenotypes* sections. It gives you the summary of statistical differences between the selected and another dataset.

- A summary of a dataset is reported based on the selected question variables (By pressing simultaneously the **ctrl** button and clicking the questions multiple question can be added)
- The **Compare to** fields enables you to select a dataset in a folder to which the reported dataset is compared to
- Comparison can be made by showing *significant results only*. In this case only statistically significant results are printed:
- N.S.= Statistical significance is not significant ($p > 0.10$)
- * = Statistical significance is almost significant ($p < 0.05$)
- ** = Statistical significance is significant ($p < 0.01$)
- *** = Statistical significance is very significant ($p < 0.001$)
- Pressing **OK** creates the summary report
- The summary report can be printed by pressing the **Print report** button

Crosstab all

With the **crosstab** function the user is able to view the dataset variables in the crosstable report (except in the **Pe digrees** section)

- Select the question/questions that is / are included in the cross-tabulation report (the vertical column in a report). By pressing simultaneously the **ctrl** button and clicking the questions multiple questions can be added
- Select one grouping question (the horizontal column in a report)
- Comparison can be made by showing *significant results only*. In this case only statistically significant results are printed:
- N.S.= Statistical significance is not significant ($p > 0.10$)
    - * = Statistical significance is almost significant ($p < 0.05$)
    - ** = Statistical significance is significant ($p < 0.01$)
    - *** = Statistical significance is very significant ($p < 0.001$)
- Summary report displays the symbols of statistical significance in the right top corner of every question
- Pressing the **OK** button displays the report
- The report is able to be printed by pressing the **Print report** button

User roles
Analyst

Child pages:

- BC|INSIGHT - 4.3.1 Queue system

Table of contents:

- Basic principles
  - General settings
  - Filtering subjects
  - Requirements for genomic analysis

Basic principles

External analysis tools can be found in the dataset's **ANALYSIS** -tab. The available tools are filtered based on the dataset type. So for example genetic analysis tools will not be available for a dataset that contains only phenotype data. The type of dataset is determined by the system, based on the form used to generate the dataset. In case of joined datasets both parent forms are used to select analysis tools.

Analysis tools are categorised but you can also search for them by typing to the filter field (Image 1).

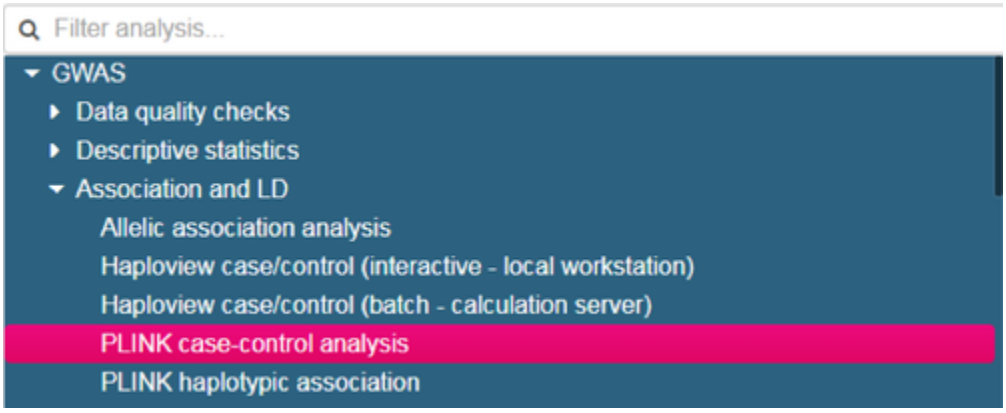


Image 1. Analysis menu shows available analysis tools for the dataset type. You can filter the options by typing to the Filter text box.

Each analysis tool has it's own GUI choices. Some are common (like selecting phenotype or other extra data from other datasets) and some are specific for the tool in question (like parameter selections). The purpose of the analysis page is to collect user's input and choices for filtering of the data, and tool-specific parameters into a job package. The job package will be used to export the data from the database, and execute the tool with correct input.

The executed analysis job is put into a queue for processing. This processing filters erroneous data, or data that is not sufficient to be used in the analysis run. These extra filtering steps are recorded and reported to the user in the final report of the analysis. After the analysis job has run, the reports created by BC|INSIGHT and the executed algorithm are all saved in a result folder.

General settings

Each analysis tool page has a section called **General**. This section allows the user to define at least the following things:

- **Run name**: Use this field to give a name for your analysis run. This will be displayed in the result archive as the job name.
- **Run mode**: With this function you can select the mode of running analysis. When using the normal run mode, the analysis program is run and it produces text report (and possibly graphical) output. When using the data export mode, the system only generates the analysis input file(s) in the format of the analysis program, but does not actually run the analysis.
- **Maximum run time**: This sets the maximum time for the algorithm to run, after which the job is automatically cancelled. Note that the maximum run time does not include either the time used for preparing the input files or the result reporting. It only applies to running the executable itself.
- **Send all analysis directory contents**: Handling of intermediate output of the too; If this is checked, the output report will contain all input files, intermediate results and other files generated for or by the analysis program. This is intended to help in validation of analysis results and solving potential problems.

In addition some tools have package-specific options like alternative versions, and so on. For some tools the run environment can be defined, if a grid-engine is in use in the analysis system.

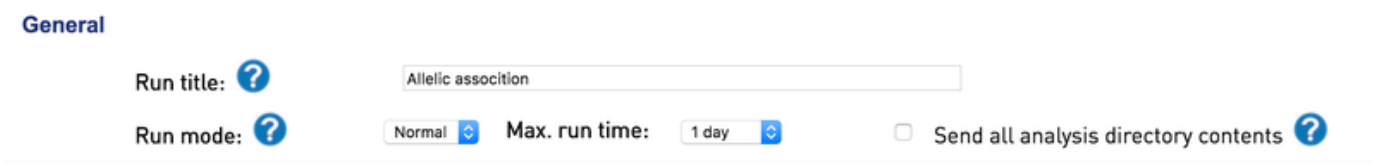


Image 2. Generic analysis option available for most analyses in BC|INSIGHT. User can define a name other than the default provided, and select Run mode and maximum run time.

Filtering subjects

Many analytical tools are subject -based algorithms, and therefore need the ability to filter subjects for which the data will be exported. Subjects -section is used for this purpose. If no selections are made here, all subjects with sufficient data for the analysis are used. You may need to expand the Subjects -section by clicking the blue arrow, if you want to set filters. You can either list the subject identifiers manually, or use another dataset/subset to specify, which subjects you wish to include or exclude.

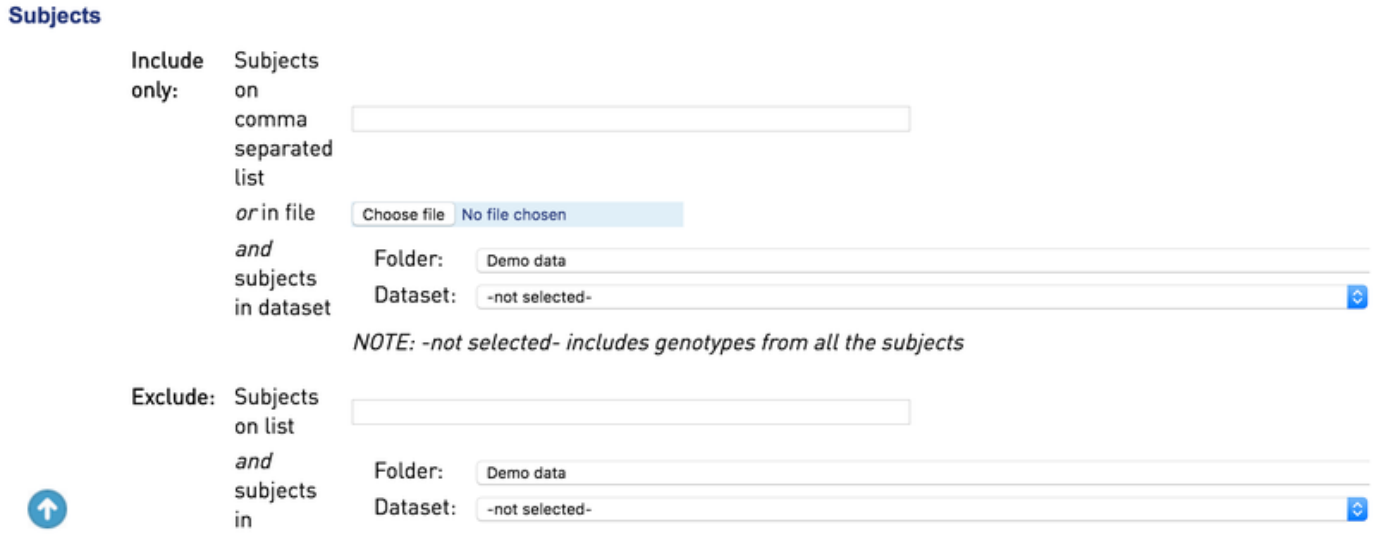


Image 3. Option for filtering subjects in analysis interface. It is possible to either create whitelist (include those specifically listed), or a blacklist (exclude only listed subjects).

Requirements for genomic analysis

BC|INSIGHT analysis drivers run statistical analyses using data stored in datasets or files. When analysing genomic data, before you can submit an analysis job you need to make the following checks:

- Make sure that all datasets or filtered subsets needed for analysis are available,
  - phenotypes
  - maps
  - any other files or datasets specific for the tool
- Make sure that *subject IDs* between phenotype and genotype datasets (and other connected datasets or files) match.
- Make sure that there are no fields highlighted in red on the analysis page before you select ***Run***, or the analysis does not start.
- If you need to specify sex information for your analysis, the *variable ID* named ***SEX*** is specified either in a pedigree or phenotype dataset.
- For case and control analysis you need two datasets/subsets: one that specifies cases and one specifying controls.
- ***Or*** you can use an affection status dataset/subset that specifies a set of cases (categorical value of 2) and controls (categorical value of 1). You must have *variable ID* named ***AFFSTAT*** in the affection status dataset.

For genotype (SNP or variant) analysis:

- In case your genotype data contains *marker IDs,* like RS codes, you must have an annotation dataset with matching map information for these marker IDs, like SNPINFO dataset. This dataset should have the marker position split in chromosome and position, like in SNPINFO datasets as CHROM and POS. These columns have BC ontology terms BC:chromsome and BC:bp_position respectively, and the marker ID is stored in column with BC:marker term.
- In case your markers are named as chr:position (like 21:1234567), you can usually use the option to derive the map information from the marker name (Image 4).

Image 4. In genetic SNP/variant analyses the Markers -section is typically present. Here you need to define where the marker position is coming from. In case you use RS codes, you should select a suitable map. If your marker IDs already provide map location, check the 'Derive map information from marker labels', to take advantage of *chr:position* -type marker identifiers.

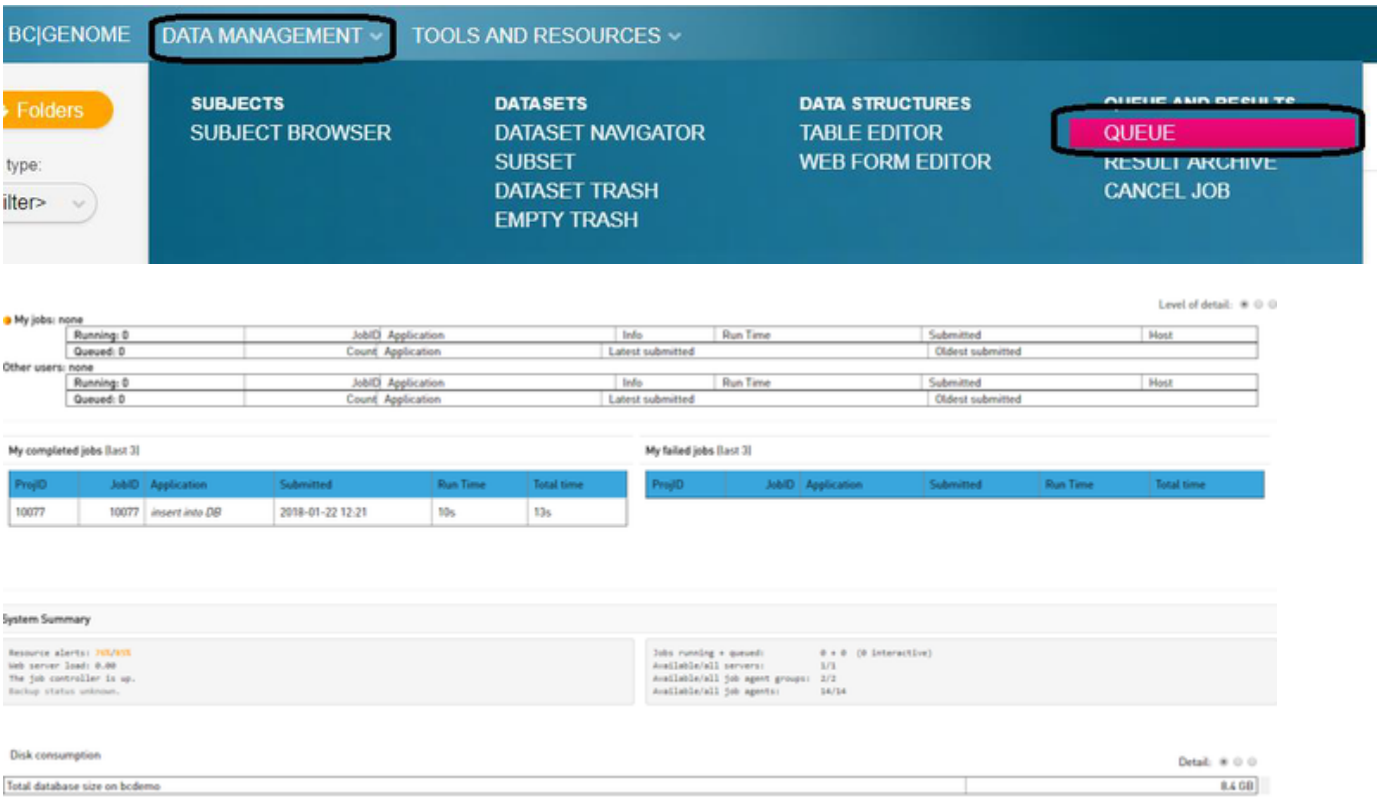
BC|INSIGHT - 4.3.1 Queue system

User roles
Analyst

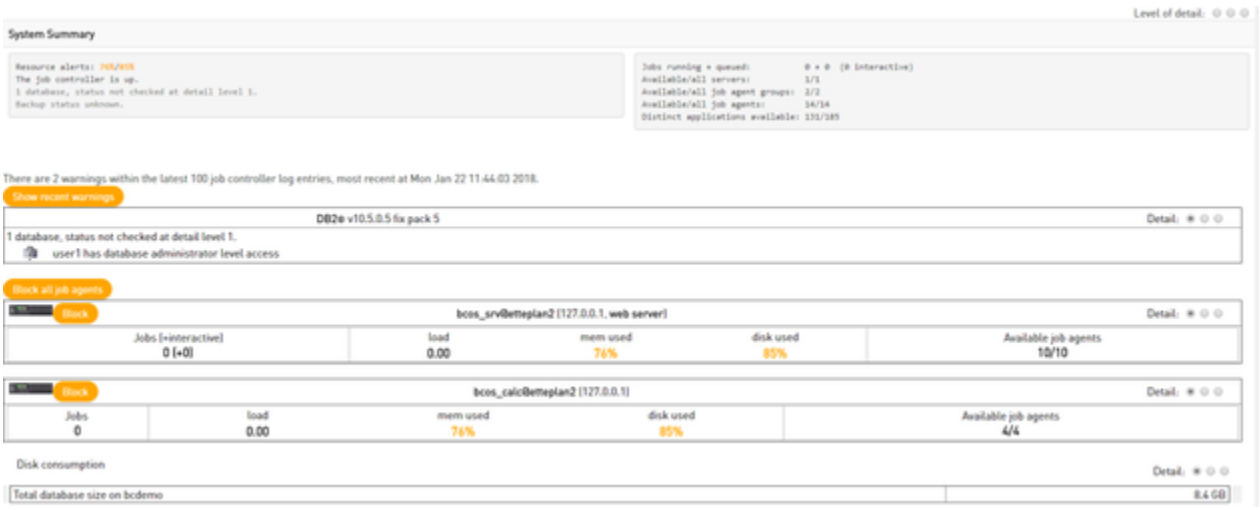
Most data management jobs are handled by the queue system. The status of each job can be seen in the QUEUE page and more detailed information in the SYSTEM STATUS page.

Jobs in one queue are prioritised by the number of jobs: the more jobs users submit to the queue, the lower priority their jobs are given, allowing other users with fewer jobs to occupy the queue. When a job disappears from the queue, the results report can be found in ***RESULTS -> Result archive*** page. Jobs can be removed from the queue by using the ***DATA MANAGEMENT > CANCEL JOB*** function.

To see the queue, select ***DATA MANAGEMENT > QUEUE***.



To see the system status, select ***TOOLS AND RESOURCES > SYSTEM STATUS***.



BC|INSIGHT - 4.4 Analysis results

User roles
Analyst

Child pages:

- BC|INSIGHT - 4.4.1 Results content
- BC|INSIGHT - 4.4.2 Uploading results to database

Table of contents:

- Overview
- Using Result archive
  - Managing single folders
  - Bulk actions

Overview

BC|INSIGHT Result Archive is an interactive way to monitor and handle completed jobs and their report files. When an analysis is run, the generated report folder is given a file name, *jobXXXXXX*.

The result archive contains a list of job folders (each contains a link to the report) and information about each job folders.

In addition, there are other folders listed in the result archive, for example, ***upload*** , ***shared***. These are links to your upload folder content, and to results that have been shared with you, by other users.

Using Result archive

You access the Result archive from the top menu ***DATA MANAGEMENT > RESULT ARCHIVE*** . or you can open the ***RESULTS*** -page in your currently selected dataset's navigator view.

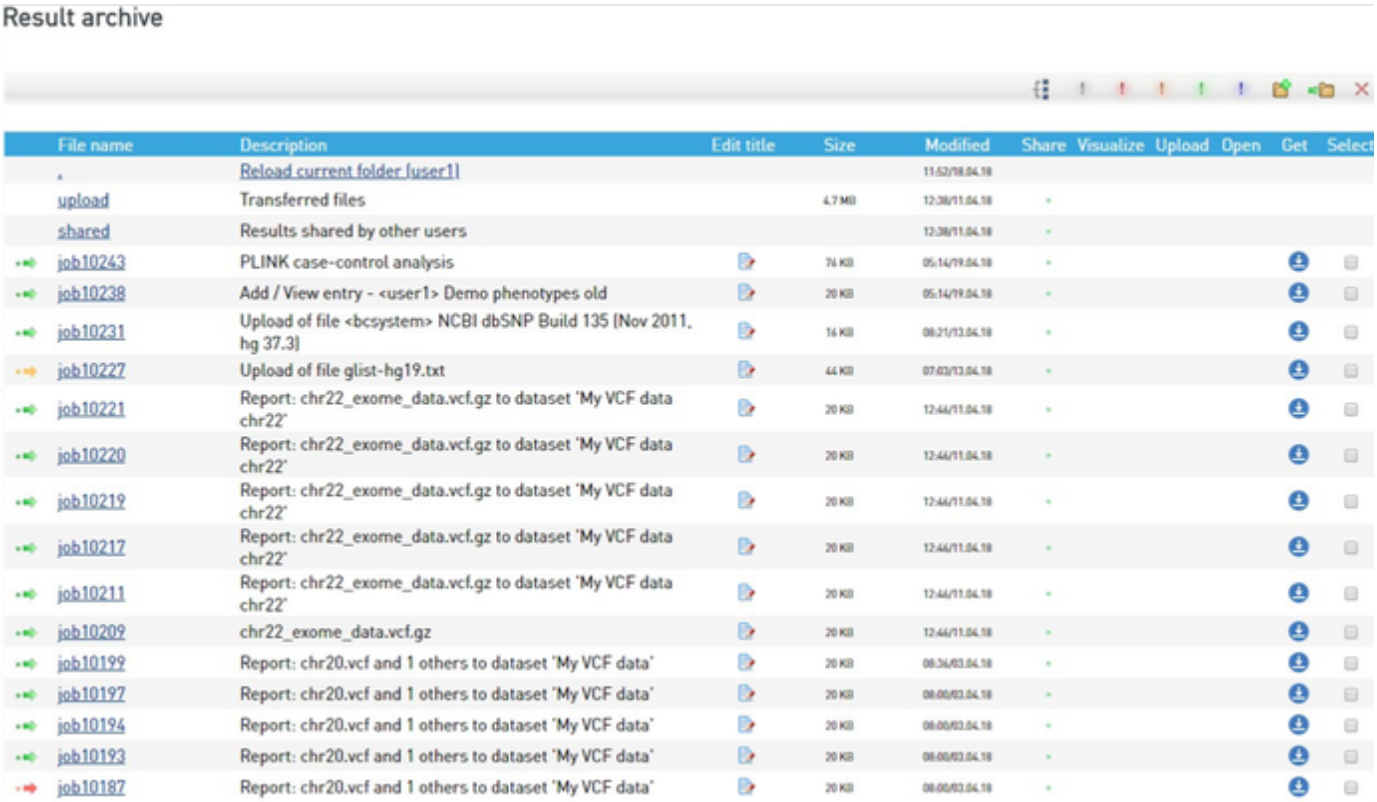


Image 1. View of the Result archive in the BC|INSIGHT.

Managing single folders

In the result archive, there are several tools that you can use to work with your job folders. Some options allow you to manipulate and manage the job folders, while others are for monitoring and viewing the results.

You can edit the job folder name by clicking the editing icon in the ***Edit title -*** column. Type a new name for in the dialog ad click OK.

To share the contents of a result folder, you click on the icon in the ***Share*** -column, and choose from the opening dialog the names of the users you wish to be able to see your results. The icon now changes to indicate sharing, and you can change the scope or remove sharing from the same tool.

The ***Get*** -column has the download tool, which allows you to wrap the content of the job folder into a package and then download that package to your PC desktop. You can choose between .ZIP and .TAR.GZ package formats.

You can create new folders by selecting the icon for new folders in the tool row above the result table.

Bulk actions

There are few operations like Delete, Move, and setting colour codes that you can do in bulk. Select the job folders you want to manipulate using the checkbox on the right hand side of the table.

Bulk -action tools are above the result table. Choosing Delete will remove the folders from the hard drive, and if you do not have regular backups, you may lose data.

> **Warning**
> Delete and Move are file system operations. It is possible to lose data, of you do not have regular backups made of the file system in BC|INSIGHT.

To move the selected jobs to another folder, choose the Move -icon from the tool row, and select the destination folder form the list you are shown.

You can apply different colours from the tool row to your selected result folders as well. You may find this useful in keeping an eye on important results, or to simply classify different kinds of result folders to make them stand out.

BC|INSIGHT - 4.4.1 Results content

User roles
Analyst

Child pages:

Table of contents:

- Results folder
- Content of an analysis results

Results folder

Results folder can be found either in the current dataset's **RESULTS** tab, or in ***DATA MANAGEMENT > RESULT ARCHIVE.*** The folder lists recent reports generated by the job-queue system. These can be data upload, export, and analysis reports, or other items that go through the queue system. The results are organised in job-folders, each named with the job identifier. Some of them are colour coded to indicate possible issues with the result: Green indicates your job has been completed successfully, orange indicates your job process has generated additional report files for checking, and red indicates your job has been failed completely and you should check the reports.

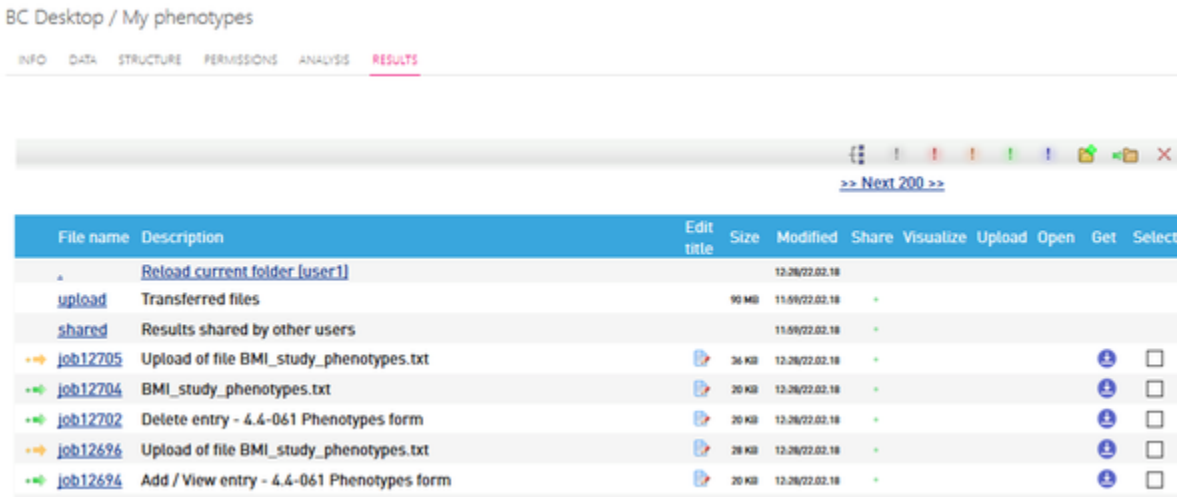


Image 1. Result archive displays the list of recently finished jobs directed in to the system queue.

The result report itself consists of a summary, or a cover page, and the actual report body: The report structure depends on the type of tasks generating the report. If there are output files or other generated files from the task being run, those are found as downloadable links in the report.

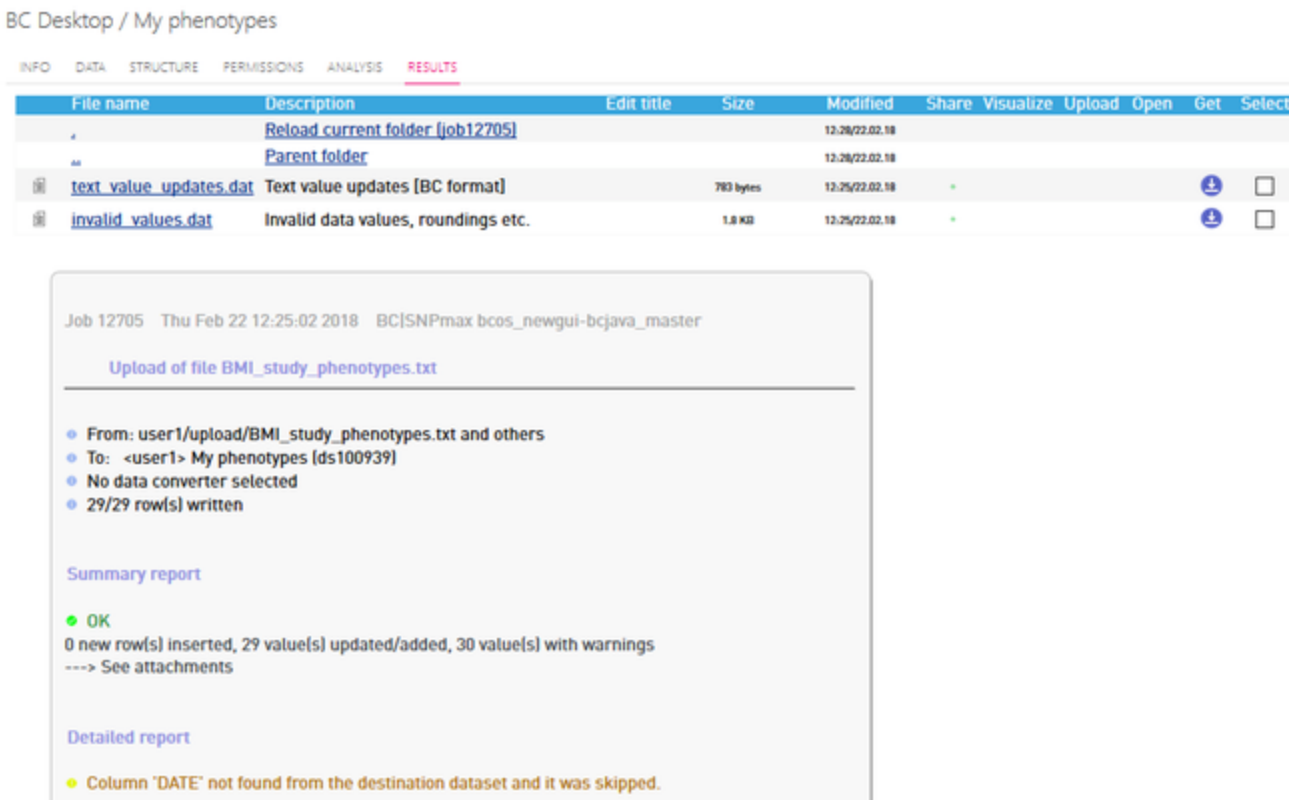


Image 2. Data upload report example.

Content of an analysis results

The output of different analytical packages varies considerably, so it is not possible to provide a comprehensive guide to all possible output and data. Typically the BC system collects any output from these algorithms and packages into an 'stdout.txt' file, if there is any output to capture. Typically many packages create their own report file about their performance during the analysis. These files can all be opened as text files directly from the result folder, or by downloading them first to your desktop (Image 3).

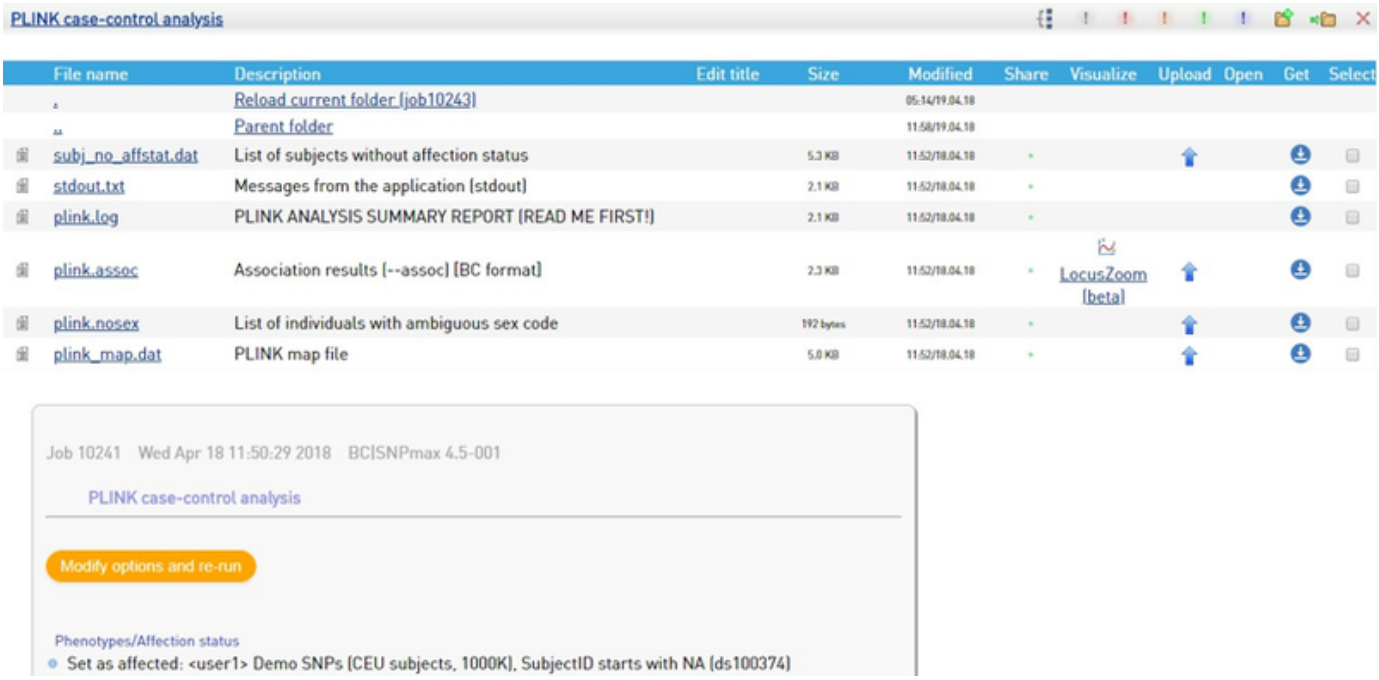


Image 3. Structure of a typical genomic analysis report. The downloadable content of created reports and result files is accompanied with a short summary report describing the flow of the work.

The Results always come with a 'cover letter', or a summary of the analysis flow, and states either success or failure. This summary may give you more information about possible changes to input data due to missing values, or otherwise unsuitable data for the particular tool. The top of the summary report has **Modify options and rerun** -button. This button will take you to the original analysis web page, where it is possible to change the parameters or external data filtering, and rerun the analysis. The rerun will generate it's own, independent result report. This is a handy way to iterate the same analysis with different settings for filtering or other parameters.

The result files come in a format that is modified by BC|INSIGHT to be more easily handled by the system tools, and to be compatible with the existing data structures. Result files may also have the option to visualise the data in some for or another, depending on the result type. If data visualising is available for that data type, an embedded link for an application will appear in the table.

BC|INSIGHT - 4.4.2 Uploading results to database

User roles

Child pages:

Table of contents:

- Uploading results from the archive

Uploading results from the archive

It is often beneficial to upload the result files back to the database for further analysis or comparison with other results. If a file in the result folder has a structure that makes it available for uploading to database, a blue arrow in the column 'Upload' will appear.

To upload data, you need to first have a dataset ready to accept the format you are about to upload, or there needs to be a converter capable of making the required transformation. After you have created or selected a suitable dataset from the Data navigator, you can access the files in your **RESULTS** -tab (Image 1).

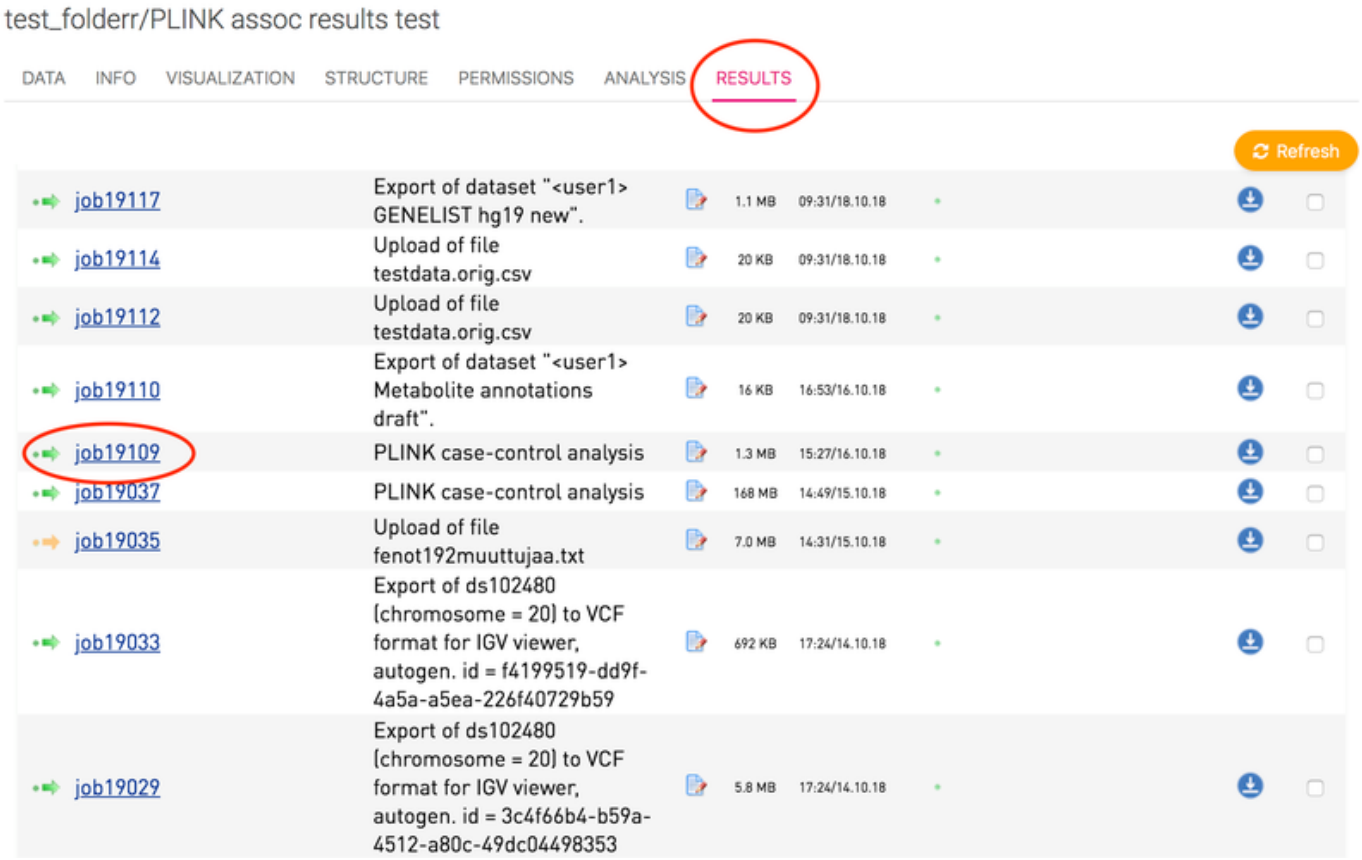


Image 1. In order to upload files from Results -archive to your dataset, select the RESULTS -tab in the dataset view, and find the result folder where your files are.

Once you have located your results, you can start the upload process by clicking the blue arrow in 'Upload' column of the result table. This will further allow you to choose a converter for your results, and possible converter-specific modifications. The upload work goes to the queue, as usual, and you will get a report in time it finishes.

BC|INSIGHT - 4.5 R script interface

User roles
Analyst
Developer

Child pages:

- BC|INSIGHT - 4.5.1 R Data input
  - BC|INSIGHT - 4.5.1.1 R Genotypes
  - BC|INSIGHT - 4.5.1.2 R Imputed data
  - BC|INSIGHT - 4.5.1.3 R Omics and multiQTL data
  - BC|INSIGHT - 4.5.1.4 R Phenotypes
- BC|INSIGHT - 4.5.2 R script Data output
- BC|INSIGHT - 4.5.3 Storing and sharing R scripts
  - BC|INSIGHT - 4.5.3.1 External R libraries
- BC|INSIGHT - 4.5.4 R script Examples

Table of contents:

- Introduction
- Running saved scripts

Introduction

This document describes various ways for advanced users to utilise R script in the BC|INSIGHT product. This manual describes steps for writing, storing and sharing scripts with other users of the system, and some practical recommendations and instructions for how to best access data, and manage the R script tasks.

The document also provides some examples of working R scripts which can be tested using the various data sets in the "Demo data" -folder of any BC|INSIGHT installation. It is assumed that the reader has expertise in writing scripts or is otherwise familiar with R. As such, this manual only offers practical notes related to running R scripts with the BC|INSIGHT system and is not a manual to the R script language.

Running saved scripts

1. Go to BC|INSIGHT Data navigator and open the dataset you wish to analyse
2. Go to Analysis tab and select the relevant R analysis option from R-script folder of analyses (Image 2)
3. Select input data for the script as for any other analysis program. Required fields in analysis GUI are marked by a star.
4. Select a script you want to use from the database (Image 3), or write your own script in the area reserved for it (useful for testing scripts).

R jobs are handled by the BC|INSIGHT queue system like any other analysis task, and when the calculation is finished, the results can be found in the result archive.

Image 2. Selecting the R -script analysis from the Analysis tab.

Image 3. Finding and selecting the pre-saved R script for the task. It is also possible to write R script from scratch to the text box.

BC|INSIGHT - 4.5.1 R Data input

User roles
Analyst
Developer

Child pages:

- BC|INSIGHT - 4.5.1.1 R Genotypes
- BC|INSIGHT - 4.5.1.2 R Imputed data
- BC|INSIGHT - 4.5.1.3 R Omics and multiQTL data
- BC|INSIGHT - 4.5.1.4 R Phenotypes

Table of contents:

- Data from SQL tables

Data from SQL tables

User can select the input data for R scripts from the database using the BC|INSIGHT user interface. Based on user's selections BC|INSIGHT generates the data frames described in the following chapters. Data frames are matrices with chosen variables (QTs or phenotypes) in columns, and subjects or samples in rows.

Data frames are slightly different for BC|INSIGHT Genotypes and Phenotypes data types. Genotype scripts process genotypic data alongside with relevant map, pedigree, affection status, and / or phenotype data. Phenotype scripts process in a generic way any data in the context data frame, and attached files. Phenotype scripts are default fallback for any datatypes in the system that might not have a dedicated R script template available. It is possible to combine multiple data frames from other datasets for a Phenotype R script.

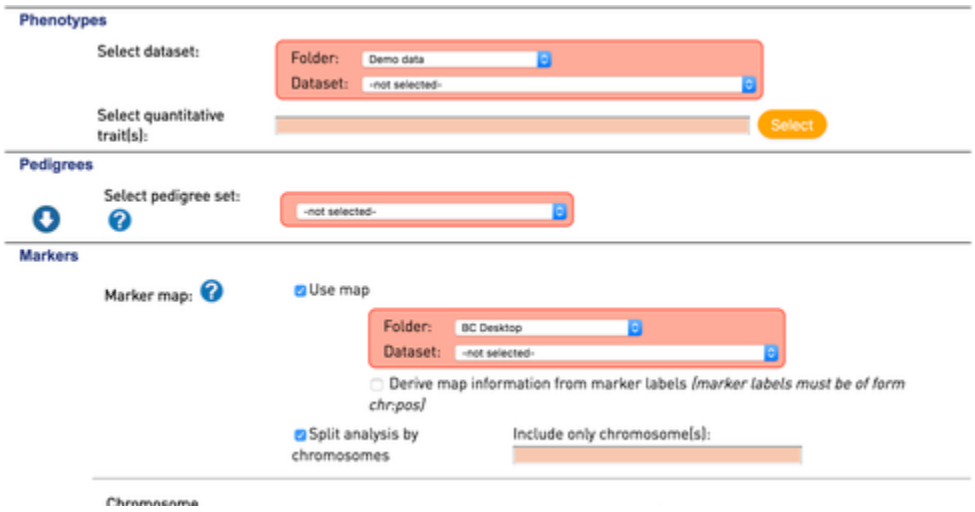


Image 1: Genotype R script analysis interface gives user the choice to attach relevant data to the script as data frames. Selections in this interface are then exported by the R engine and can be used from within the script. The available selection of extra data depends on the type of R script template that is chosen.

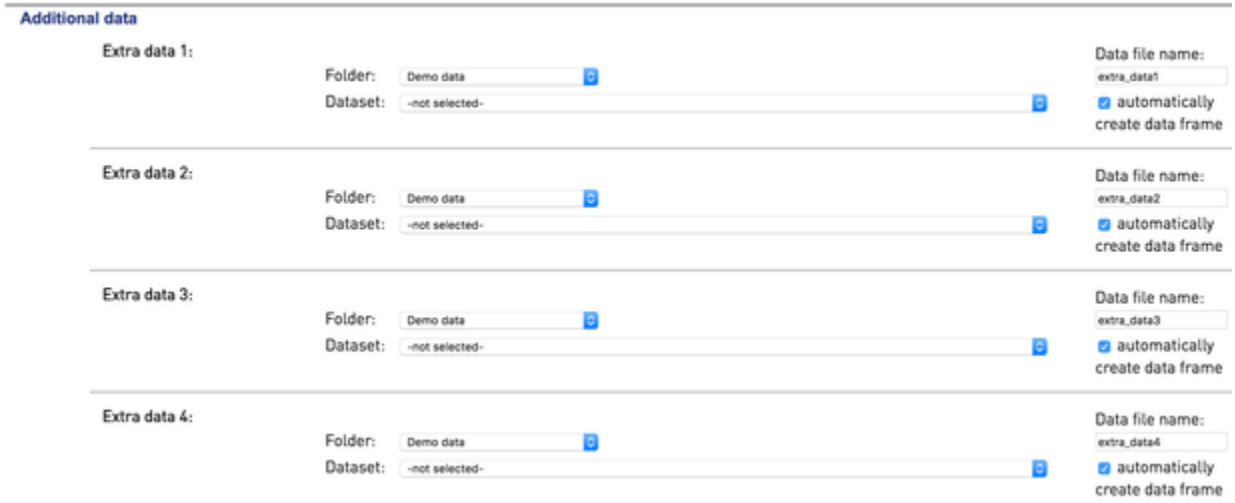


Image 2. Phenotype analysis GUI for R allows up to 4 external datasets to be imported into the script as data frames, making Phenotype R interface probably the most flexible of all.

MultiQTL data frames are formed based on user's choices in the R interface. User can select phenotype columns from the multiQTL dataset or other phenotype datasets. BC|INSIGHT generates the data frame for this data by using the subject identifier field (annotated BC_VARCLASS: patient or BC:subject) to join data.

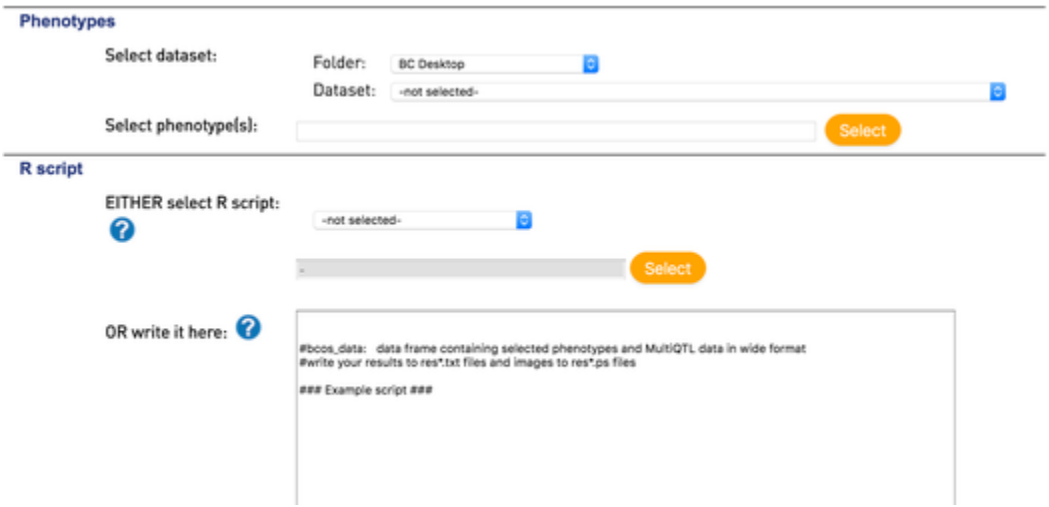


Image 3. MultiQTL analysis interface mimics genotype interface in the sense that it also allows user to select specific datasets and components from those datasets. However it is only limited to phenotypes.

BC|INSIGHT - 4.5.1.1 R Genotypes

User roles
Analyst
Developer

Table of contents:

- Data frames
- Map frame (bcos_map)
- Genotype frame (bcos_data)

Data frames

Genotype scripts analyze genotype data with associated map, pedigree, affection, and / or phenotype data. For genotype scripts two data frames are provided by the system:

bcos_map (marker map)

bcos_data (rest of the data)

Map frame (bcos_map)

Map frame bcos_map contains marker map data. Rows and columns are as in the BC|INSIGHT marker map dataset. For example:

Marker	Distance	Chrom	Order
RS0000	0	11	0
RS1111	100000	11	1
RS2222	200000	11	2
RS3333	300000	11	3

Genotype frame (bcos_data)

The basic order of the data columns within bcos_data frame is:

1. Family data (in case no pedigree data set selected, subject ID only)
2. Affection status / phenotype data (optional)
3. Allele data

An example of bcos_data with affection statuses defined:

SUBJECT	AFFSTAT	RS0000_chr11	RS1111_chr11
SAMPLE0	1	2	0
SAMPLE1	1	1	1

An example of bcos_data with pedigree data:

SUBJECT	PED	FATHER	MOTHER	SEX	AGE	RS1111_chr11
1000	1	0	0	2	38	0
1001	1	0	0	1	42	1
1002	1	1001	1000	1	12	1

NOTE: There is only one column per each genotype. The alleles are 0-1-2 coded; where 0 means alleles 11 in the original dataset, 1 mean alleles 12 and 2 means alleles 22. In the user interface, user can select whether the minor allele is coded as 1 (i.e. 0 means homozygous with the minor allele) or 2.

NOTE: A parameter bcos_first_snp defines the column where the genotype data begins. For example for the example data frames above the bcos_first_snp values would be 2 and 6.

User roles

Analyst

Developer

Table of contents:

- Data frames
- Genotype frame (bcos_prob)
- Phenotype frame (bcos_pheno)

Data frames

If the genotype datasets used in the R analysis contains imputed data (i.e. the dataset type is Compressed imputed SNPs), user can select whether to use most probable genotypes or probabilistic genotype data. If most probable genotypes are used, the data frames provided are the same as for normal genotype data. If probabilistic data is used, the following data frames are provided:

- bcos_prob (probabilistic genotypes)
- bcos_pheno (phenotypes / pedigrees / affection status data)
- bcos_map (marker map)

Genotype frame (bcos_prob)

Data frame bcos_prob contains the genotypes as probabilities.

An example of bcos_prob :

Marker	Allele_A	Allele_B	1000_AA	1000_AB	1000_BB
RS1000057	C	G	0.001	0.933	0.066
RS1000081	A	G	0.999	0.001	0.000
RS1000113	C	T	0.997	0.003	0.000

All probabilities are 0.000 for missing genotypes.

NOTE: Also all subjects that belong to selected pedigrees are included in the bcos_prob frame even if they do not have any genotype data.

Phenotype frame (bcos_pheno)

The basic order of the data columns within bcos_data frame is:

1. Family data (in case no pedigree data set selected, subject ID only)
2. Affectionstatus/phenotypedata(optional)

An example of bcos_pheno with only affection statuses defined:

SUBJECT	AFFSTAT
SAMPLE0	1
SAMPLE1	2

An example of bcos_pheno with pedigree data and one phenotype variable:

SUBJECT	PED	FATHER	MOTHER	SEX	AGE
1000	1	0	0	2	38
1001	1	0	0	1	42
1002	1	1001	1000	1	12

User roles

Analyst

Developer

Table of contents:

- Data frames

Data frames

Data in multiQTL and omics datasets can be analyzed as such or together with phenotype data. The data frame contains the multiQTL and omics data in matrix format (row=subject, col=variable).

An example of bcos_data multiQTL data frame without phenotype data:

SUBJECT	A2A1	BCL2L1	BCL2L10	BCL3
NA06985	169.197	301.341	143.002	895.921
NA06991	638.602	338.372	41.711	129.343
NA06993	161.296	659.965	385.962	780.335

An example of bcos_data multiQTL data frame with phenotype data:

SUBJECT	SEX	A2A1	BCL2L1	BCL2L10	BCL3
NA06985	1	169.197	301.341	143.002	895.921
NA06991	2	638.602	338.372	41.711	129.343
NA06993	2	161.296	659.965	385.962	780.335

User roles

Analyst

Developer

Table of contents:

- Data frames
  - Phenotype data with two keys

Data frames

Phenotype scripts can be used to analyze only phenotype data. If you need to combine genotypic or omics data into the analysis, design the script to be launched from the genotype or omics dataset analysis tool. By default, phenotype scripts have input data in a data frame called bcos_data, containing data in rows and columns as in the BC|INSIGHT phenotype dataset.

An example of bcos_data phenotype data frame:

SUBJECT	CHOL	GLUC	HDL	DBD	TRIG
NA06991	193	105	38	1	431
NA06993	155	101	30	1	177
NA06994	209	106	63	1	91

In phenotype scripts it is also possible to bypass the default frame by defining one's own in the script like in the example below:

my_data = read.delim("^$INFILE$^", row.names=0)

Here the token "^$INFILE$^" is replaced by the actual data file name. All custom data frame definitions must contain the token "^$INFILE$^".

Phenotype data with two keys

For example in case of a follow-up study, user will have to deal with two-index data where there's both a subject ID and a date corresponding to each particular set of measurements.

An example data frame for two-index data:

SUBJECT	VISIT	AGE	SEX	SCR
1000	2005-12-30	50	1	10.0
1000	2006-11-10	51	1	9.2
1001	2003-9-21	30	2	7.6
1001	2004-5-21	31	2	9.0

BC|INSIGHT - 4.5.2 R script Data output

User roles

Analyst

Developer

Table of contents:

- Result files

Result files

Output files from the R scripts are delivered to the user's result archive, when the calculation is finished. System automatically sends all data printed to stdout and stderr to user. By default all files created by the script are cleaned upon exit, except output files beginning with the string "res ". Therefore temporary files can be used without worrying about cleaning, as long as the file names do not start with "res". The possible output file types are listed and explained in Table 1.

Table 1. Output file types and their content.

FILE	EXPLANATION
res*.txt	Text files are returned to the user, and if the run is segmented, the files produced in each segment are concatenated into a single file upon delivery.
res*.ps	Postscript files are converted to PDFs and returned to the user. If the run is segmented, the PDFs from each segment are delivered together in one ZIP file.
res*.dat	DAT files behave the same way as res*.txt files, but they provide a way to create a separate file association in the browser e.g. notepad for small reports in *.txt files and MySuperStatSoftware for large *.dat files.
res*.bcos	BCOS files are assumed to be tab delimited files with a header row. In segmented runs they are concatenated so that the first row (header) is stripped from all except the first segment, thus producing a single clean tab delimited file with a header row.

BC|INSIGHT - 4.5.3 Storing and sharing R scripts

User roles

Developer

Child pages:

- BC|INSIGHT - 4.5.3.1 External R libraries

Table of contents:

- Storing scripts
- Sharing scripts

Storing scripts

It is strongly recommended that all scripts are thoroughly tested either from the shell, or from the R analysis GUI, before scripts are stored in R script datasets and shared with other users.

1. Create a new dataset using the form R scripts or Phenotype R scripts (Image 1)
2. Go to the 'Data' tab of the newly created dataset and choose Add -tool
3. Give the script a unique ID, a clear description of what it does and what data fields it requires
4. Fill in author name
5. Copy paste the script to the CODE -field
6. Save the entry

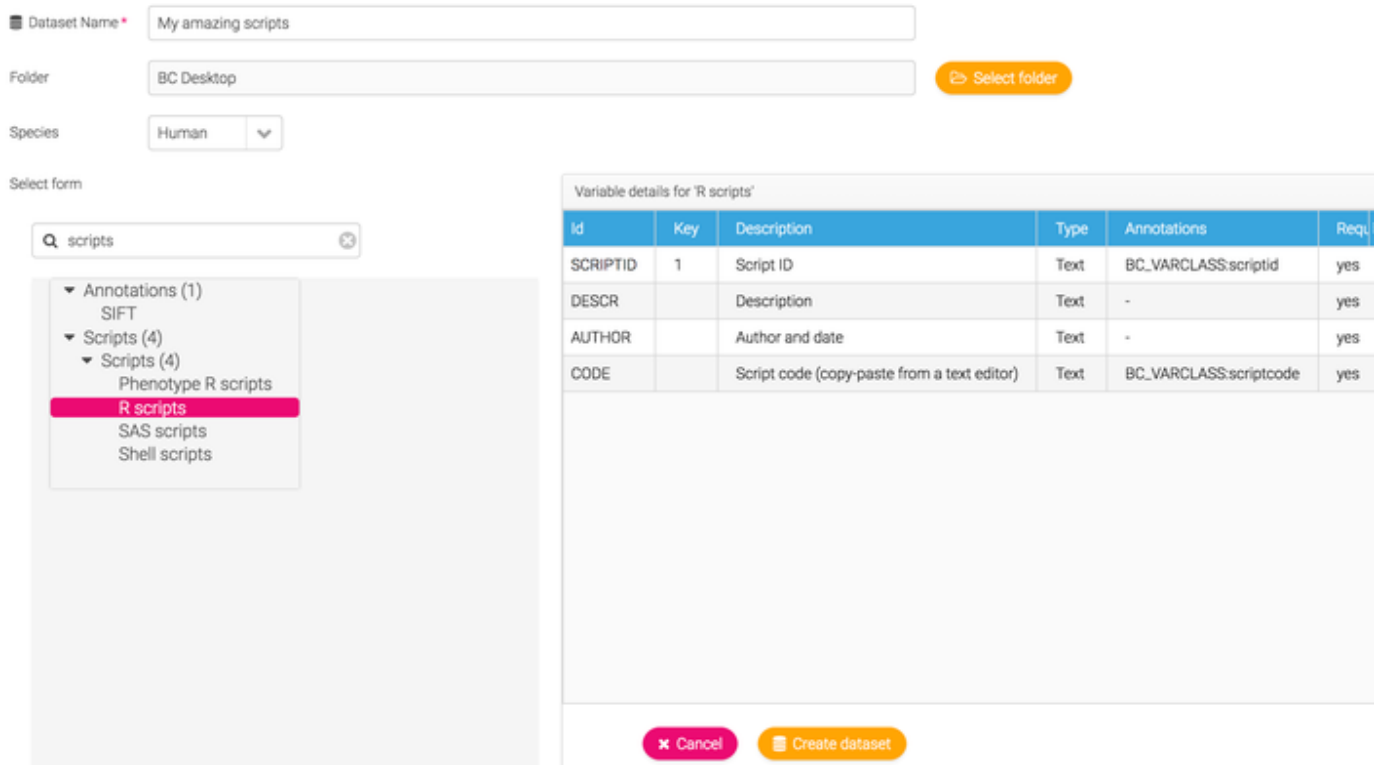


Image 1. Creating a script dataset to store and share R scripts.

Sharing scripts

1. In the scripts dataset open the Permissions -tab
2. Grant permissions to other users or user groups, as you would normally grant them to datasets
3. Permissions affect the use of scripts and script storage in following ways:
    a. All permissions: Users are able to run all scripts in dataset, modify them, and add new ones
    b. Read only: Users are able to run the scripts in the dataset
    c. Write only: Users are able to store new scripts in the dataset

Some user roles are restricted in ways that prevent them from adding or modifying R scripts, independent of the permissions they have to script storage.

Note that it is possible to create subsets of the script storage as with any other dataset, and grant permissions in those subsets.

BC|INSIGHT - 4.5.3.1 External R libraries

User roles

None, this page if information only

Table of contents:

- Using R libraries

Using R libraries

It is often necessary to use R libraries and packages that are not part of the normal R distribution. In BC|INSIGHT the system uses the default R package, which is accessible to all BC system-level users, including those taking care of calculation tasks. Therefore it is recommended to strictly follow the instructions for you own environment and OS in installing new packages. The most commonly used and probably the least error-prone is to install new libraries through the R shell, as is described for example here https://www.r-bloggers.com/installing-r-packages/.

This approach, however, requires that the installation is done using root privileges, otherwise the new binaries will not be visible to the system-level user accounts that run the analyses. If you cannot use root privileges, please contact BC support for more help for configuring your calculation environment to have access to these libraries.

BC|INSIGHT - 4.5.4 R script Examples

Child pages:

User roles

Developer

Table of contents:

- Example scripts for Demo datasets
  - Genotypes
- Other examples
  - How to access FILE variables

Genotypes

Can be stored in R script dataset that uses the basic format "R scripts".

The following script executes QTL analysis on ACGT coded SNP dataset, taking the marker map and phenotypic data from the GUI selections the user makes.

R script for quantitative traits

```
#Resultfile titles
resdata="results.txt"
resfig="results.ps"

#Result titles
write(c("TRAIT","MARKER","BINTC","BSLOPE","BINTC_SD","BSLOPE_SD","
BINTC_T","BSLOPE_T","P_INTC","P_SLOPE","ADJ_R"),file=resdata,
append=FALSE,sep="\t",ncolumns=11)


#Image settings
postscript(file=resfig,horizontal=FALSE,pointsize=5)
par(mfrow=c(4,3),omi=c(1,0.5,0.5,1))

#Loop traits
for(trait in 1:(bcos_first_snp-1))
{
  #Loop SNPs
  for(snp in bcos_first_snp:length(bcos_data))
   {
     #Calculate regression
     fm=lm(bcos_data[[trait]]~bcos_data[[snp]])
     res=summary(fm)

     trait_name=attr(bcos_data,"names")[trait]
     marker=attr(bcos_data,"names")[snp]
     results=c(trait_name,marker,res[[4]],res[[9]])

     #Draw image if F>5.
     if(res[[10]][1]>5)
      {
        boxplot(bcos_data[[trait]]~bcos_data[[snp]],ylab=trait_name,
xlab=marker)
      }
```

```
      #Write results
      write(results,file=resdata,append=TRUE,ncolumns=11,sep="\t")
    }
}
dev.off()
```

Other examples

How to access FILE variables

IMPORTANT: The BC|INSIGHT database file storage or 'blob storage' must be configured to use so called BCFS (BC virtual file system), in order for the R interface to have legitimate access to the files. If your system already uses external file storage or cloud storages for archiving and accessing files, your BC|INSIGHT instance is configured in this way. If this is not the case, please contact your system administrator for more information about the possibility of making this configuration.

If you create scripts that need to access FILE type variables (like list of BAMs, omics data files, etc) within the dataset, you need to save the R script as 'Phenotype R script', or run the script from Phenotype R script interface. At the moment BC systems do not support FILE access for R scripts in genotype script templates. This will be amended in future versions.

R script accessing FILEs

```
# bcos_data:   data frame containing the phenotype data in wide format
# write your results to res*.txt or res.data files and images to res*.
ps files
#
# Make a matrix out of dataframe
# and get the dimensions of the dataset

bcos_data_tab = as.matrix(bcos_data)
dim (bcos_data_tab)


# Get the index of the FILE variable you want to open
fidx = grep("^SOMEFILE$", colnames(bcos_data_tab))
fidx

# Read the content of the file on row 1, and print it
cont1 = read.delim(bcos_data_tab[1, fidx], quote="", header=FALSE)
cont1



# In case you have specified other datasets to be used in this script,
# those can be accessed from "data1", "data2", "data3", and "data4"
variables
#
#data1_tab=as.matrix (read.delim("data1", quote="", sep="\t",
header=TRUE))
#dim (data1_tab)
#
```

BC|INSIGHT - 4.6 Genome browsers

User roles
BC|INSIGHT user Analyst

Child pages:

- BC|INSIGHT - 4.6.1 LocusZoom
- BC|INSIGHT - 4.6.2 Manhattan and QQ plots
- BC|INSIGHT - 4.6.3 UCSC genome browser
- BC|INSIGHT - 4.6.4 Embedded IGV
- BC|INSIGHT - 4.6.5 Data service for IGV and LocusZoom

Table of contents:

- Embedded and external browsers
- Genomic association results

Embedded and external browsers

BC|INSIGHT has context-dependent implementations of various browsers for genome-based tracks and statistical results. Embedded browsers are available on the BC|INSIGHT data navigator (Image 1) and result browser pages (Image 2) as part of Charts -tool, and display the data in a file or a dataset, whilst providing some filtering and navigation options that would not exist in the original browser. These browsers are LocusZoom for statistical results at gene-range level window, and IGV for genomic data files like BAM and VCF data.

For genomic association results the Charts -tool also gives interfaces for creation of Manhattan plot, and QQ plot.

BC|INSIGHT provides also a link for uploading track files to UCSC browser, which is located in an external URL, and requires that the BC|INSIGHT server is able to access the URL. Some organisations have their own internal UCSC browser instances to circumvent possible network restrictions, and BC|INSIGHT can be configured to use those instances.

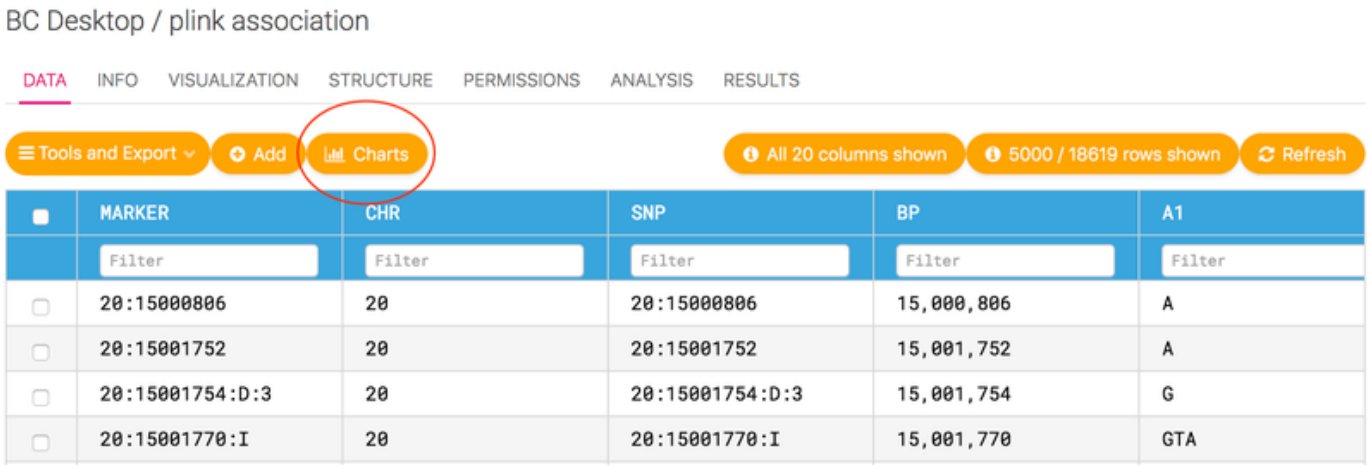


Image 1. Finding the Charts -tool in dataset context.

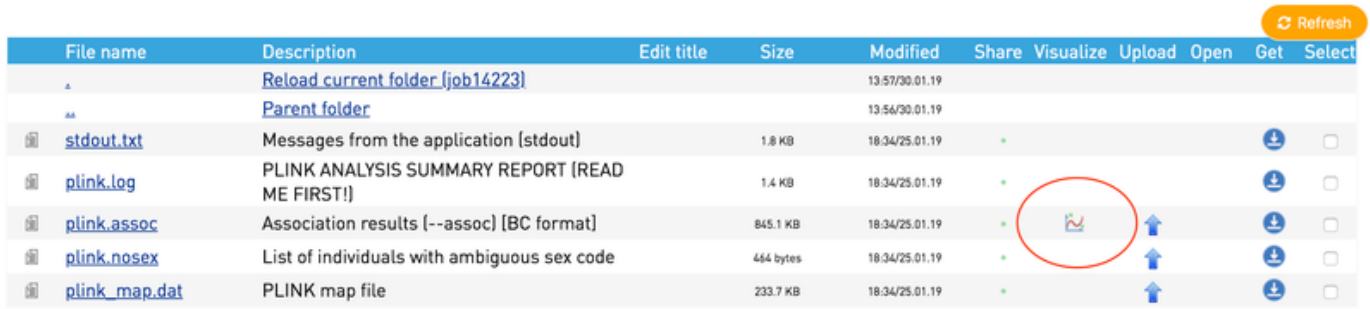


Image 2. Finding the Charts -tool in analysis results context.

> **Note**
> Analyst -role is required to run analyses and browse results for visualisation.

Genomic association results

The LocusZoom, Manhattan and QQ plot require a certain structure in the genomics association results, in order to be able to display the result data. The system comes with an inbuilt form for PLINK results specifically (PLINKRES, "PLINK association results"), but the same form can be used for uploading results from other algorithms, provided that necessary data conversion has been done by the file owner. The table below lists the mandatory and optional fields that the visualisation tools recognise, and are able to utilise. If a BC annotation is provided for the column, it means that in a dataset, any column name can be used for that field, for as long as the BC annotation is added to the column. For example you may have in your dataset a field 'SNP' and annotate that column with 'BC:marker', which will allow the visualisation tools to recognise the field correctly. Annotations for columns do not apply, when visualising files directly.

Column	Description	BC annotation	Mandatory
MARKER	The name of the genetic marker, can be RS code or chr:xxxxx format	BC:marker	Yes
CHROM	The chromosome	BC: chromosome	Yes
DIST	Absolute base pair start position of the marker within the chromosome	BC: bp_position	Yes
P	The p-value for the association of this marker		Yes
TEST	Alternate or full-model association test, applies to PLINK results, values can be ADD, ALLELIC, DOM, DOMDEV, GENO, GENO_2DF, REC, TREND		No
TRAIT	The quantitative trait used for the association, this is the ID of the phenotypic trait selected to the analysis, if user ran a QTL analysis		No

For PLINK -specific alternate model tests see the following site http://zzz.bwh.harvard.edu/plink/anal.shtml#model. The values for TEST and TRAIT are typically used to filter or select specific results for the plot.

BC|INSIGHT - 4.6.1 LocusZoom

User roles
BC|INSIGHT user

Table of contents:

- Visualising association results

Visualising association results

Genetic association results can be visualised in the context of the gene region using embedded LocusZoom tool. The Charts application provides more comprehensive filtering options for various result types, which may include specific analysed traits and tests. The result parsing has been build based on PLINK association results. If other association analysis algorithms have been used, a conversion of result format may be required in the analysis pipe in order for Charts application to work.

For more help about use of LocusZoom visit http://statgen.github.io/locuszoom/.

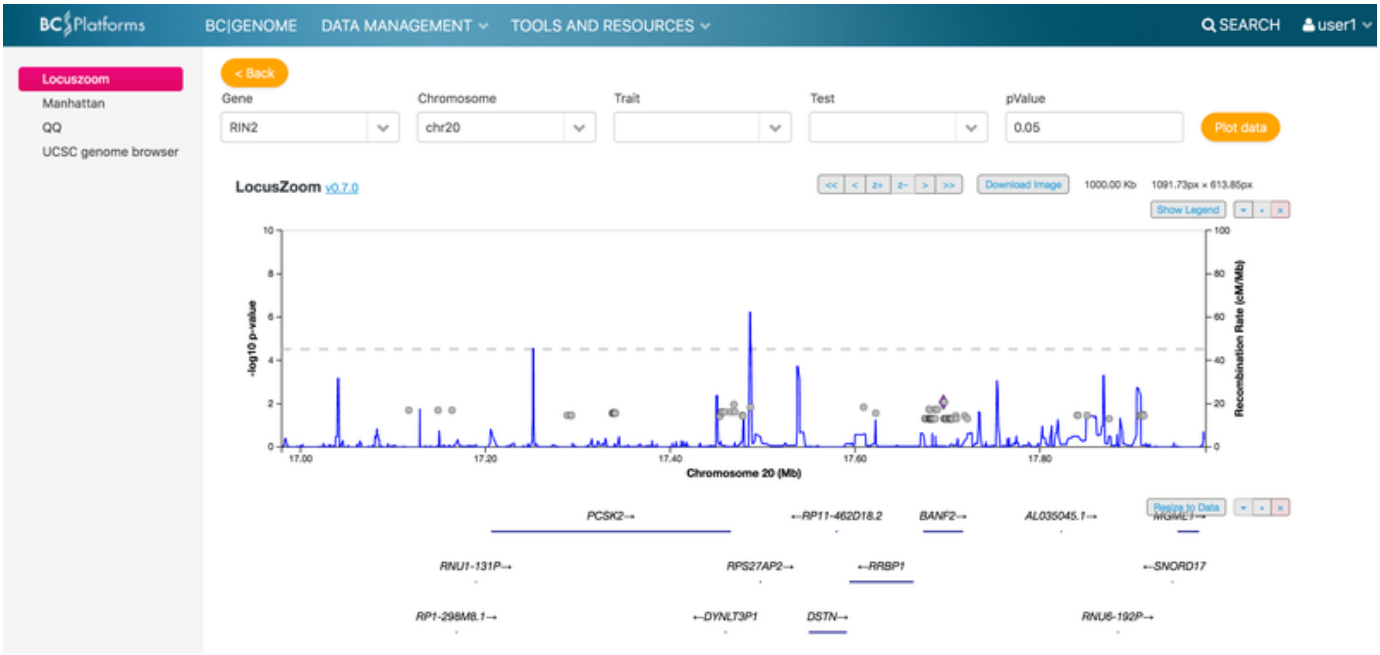


Image 1. LocusZoom provides filtering options for different types of association results (CC, QT, etc), and p-value filtering.

BC|INSIGHT - 4.6.2 Manhattan and QQ plots

User roles
BC|INSIGHT user

Table of contents:

- Genomic association result plots

Genomic association result plots

The same visualisation view offers two alternative plotters for the association data - Manhattan and QQ plot. Please refer to the data requirements defined in the parent chapter.

Manhattan plot displays as scatter plot the log-transformed p-values for each marker, as a function of chromosomal distance. This view is helpful for visualising peaks of high p-values, which may be indicators of significant association of the relevant genomic area with the phenotype of quantitative trait.

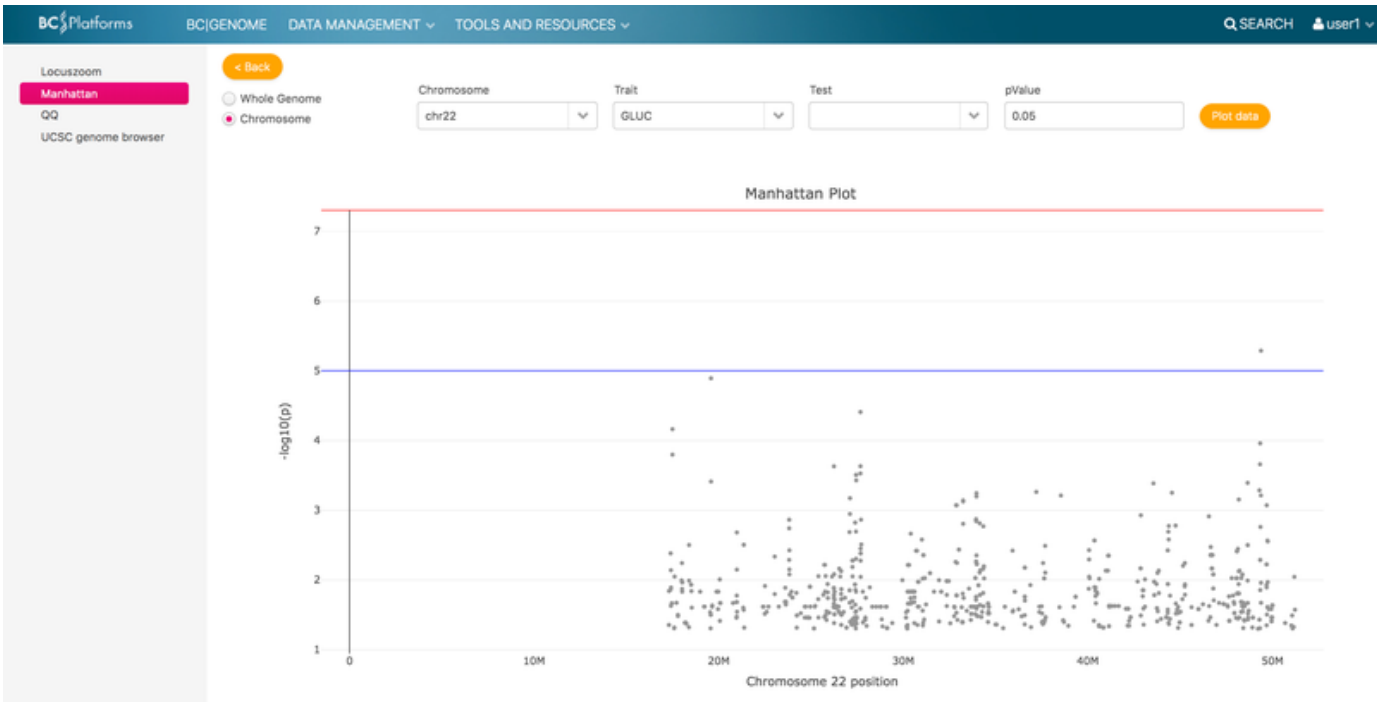


Image 1. Manhattan plot of quantitative trait analysis for fasting glucose levels, showing a close-up of chromosome 22.

QQ or Quantile-Quantile plot is used to assess, if the real-world values could be following a theoretical Normal distribution of values. In short, it tells the researcher, how well the chosen data actually described the question. If the plot form relatively uniform straight diagonal line, there are no major systematic deviations or biases between the compared groups. Typically there is a sharp uprising 'tail' of the markers that are significantly different between the groups, and indeed should then pop up in Manhattan as well. If there is clear deviation from straight line, the chances are that sampling has introduced a systematic bias to the analysis. This usually means that more care is needed in making sure sampling eliminates these biases.

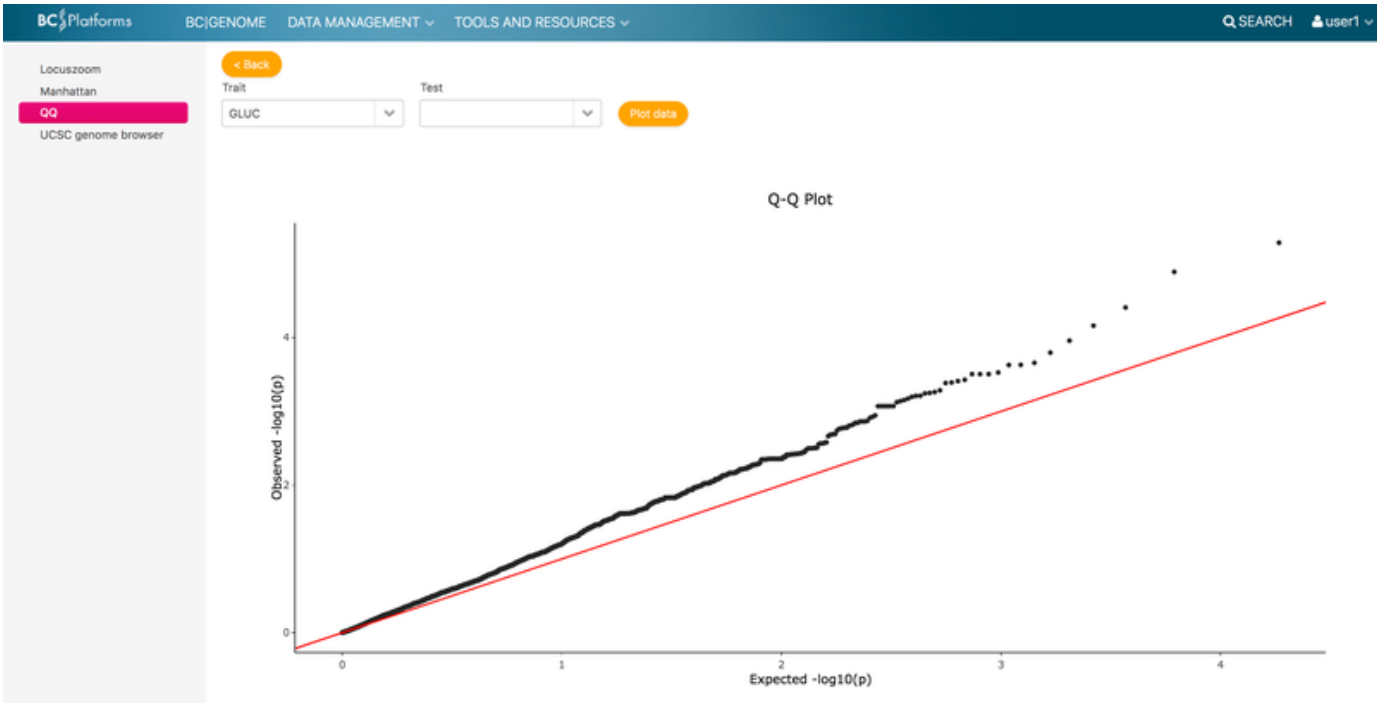


Image 2. QQ-plot showing light deviation in the association analysis, indicating the sampling process should be probably redefined. Also the end of the tail does not show any specific group of significantly differentiating markers.

BC|INSIGHT - 4.6.3 UCSC genome browser

User roles

BC|INSIGHT user

Table of contents:

- Viewing your results using UCSC Genome browser

Viewing your results using UCSC Genome browser

You can view your analysis results either in the result archive or result dataset using UCSC Genome browser. BC|INSIGHT directs users by default to a public UCSC web site, but upon request the UCSC Genome browser can be configured to use a local UCSC Genome browser, or GBiB (https://genome.ucsc.edu/goldenpath/help/gbib.html). Please contact BC support team for more information.

UCSC Genome browser can be found in the Charts -application in either the dataset's DATA and VISUALIZATION tabs, or in the result folder behind the visualisation -tool icon next to the result file. Image 1 shows the options available for UCSC track downloads. Depending on the type of the association result, you can filter the results by trait, test, and p-value. You can also choose to include all or only a selected chromosome in the track file. BC|INSIGHT will generate a suitable track file from the result data and the file can then be uploaded to the browser web application as custom track (Image 2). Download the track file and click 'Launch browser' to open the UCSC web application.

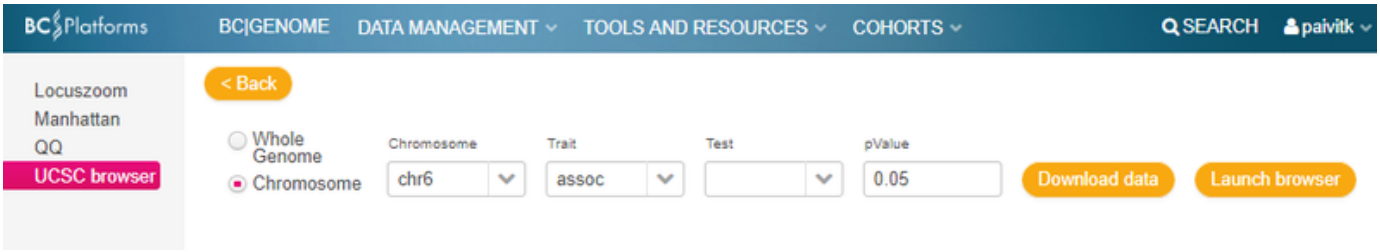


Image 1. UCSC browser interface provides filtering options to the result data, producing a downloadable track file, which can then be uploaded as custom track to the Genome Browser accessible through the 'Launch browser' button.

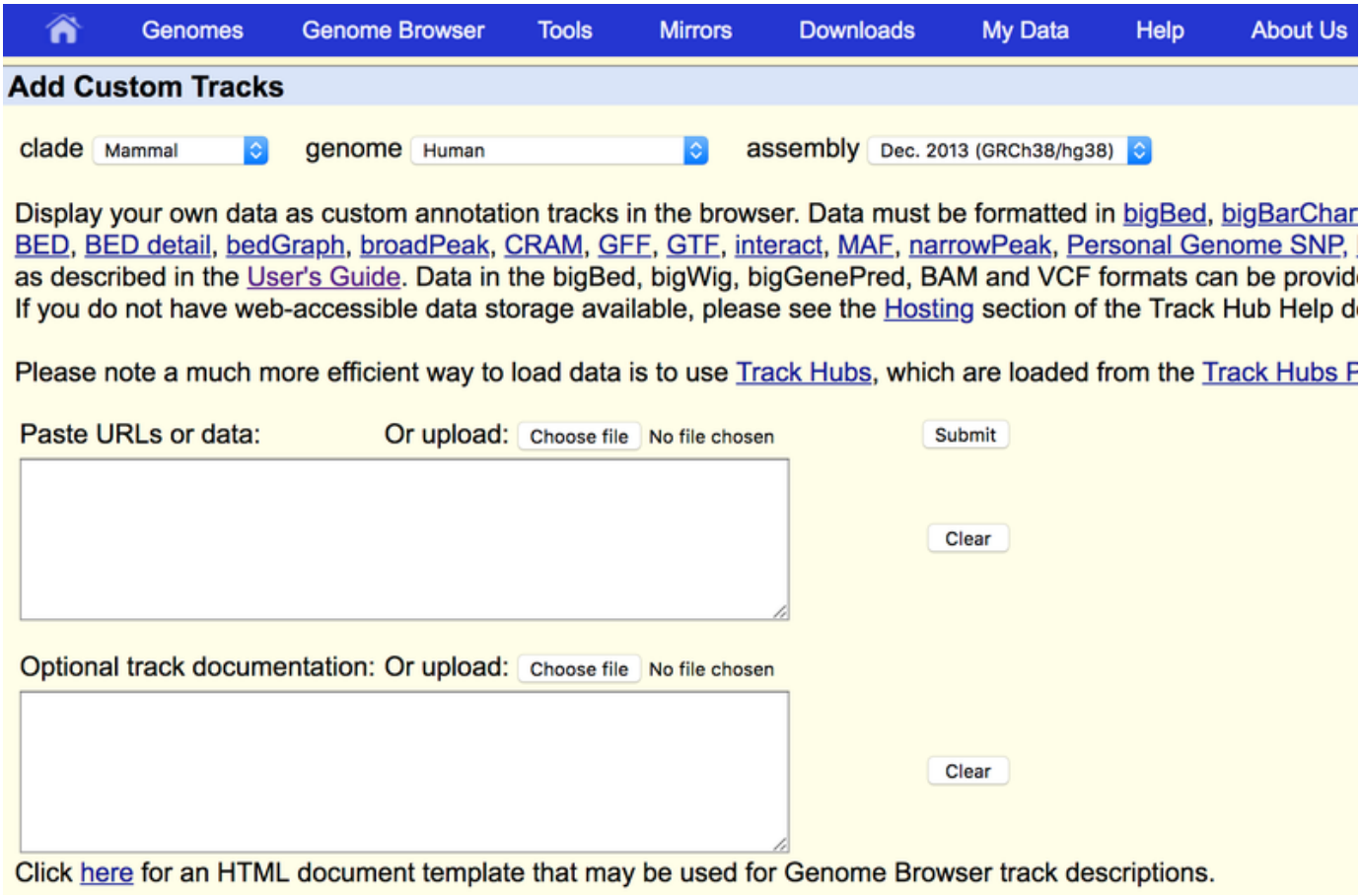


Image 2. UCSC web application tools for adding custom tracks, like association result files.

For more information about how to work with UCSCS Genome Browser, visit https://genome.ucsc.edu/cgi-bin/hgGateway.

BC|INSIGHT - 4.6.4 Embedded IGV

User roles
BC|INSIGHT user

Table of contents:

- IGV genomic feature browser

IGV genomic feature browser

BC|INSIGHT provides embedded IGV.js browser for VCF and BAM files stored as objects in database (Image 1). IGV can be launched from the Charts -application available in supported datasets' DATA and VISUALIZATION tabs. Datasets created using the following forms (or their derivatives) support IGV.

- BAM files

- BAM files with index
- BAM files with BAI and sampleId (multisample)
- VCF files
- VCF files with sampleIds (multisample)

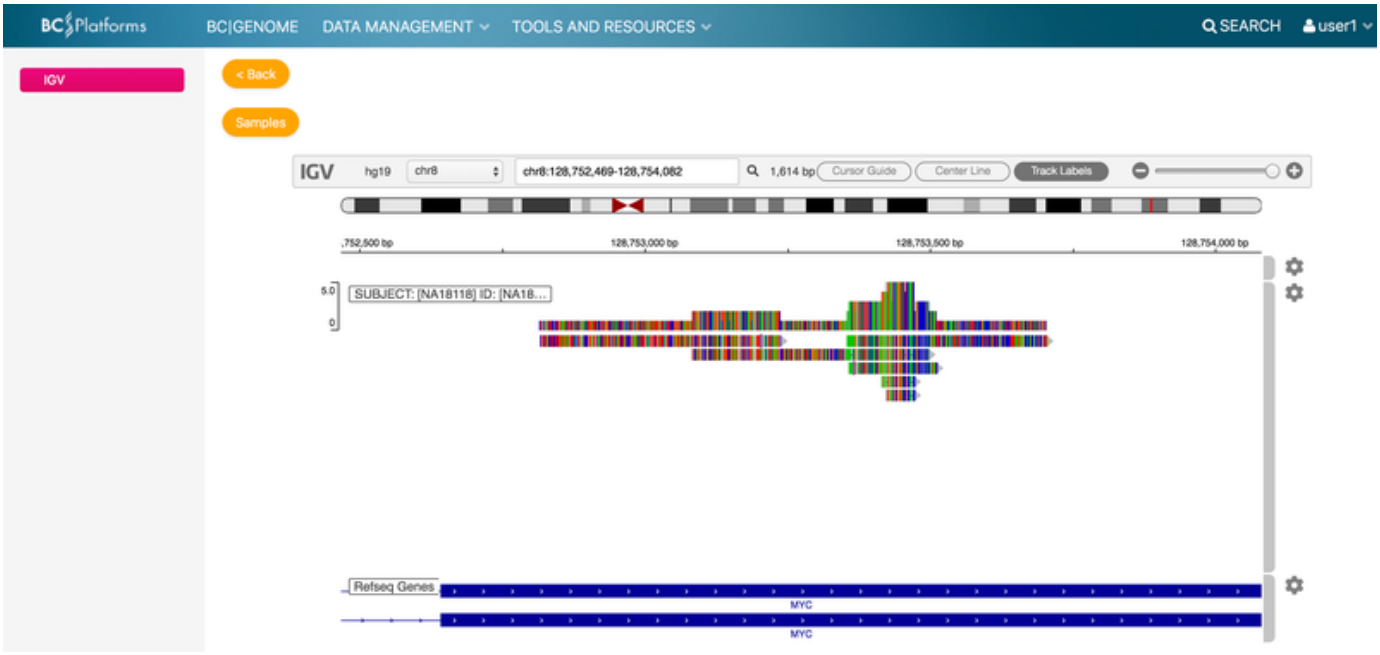


Image 1. Embedded IGV showing .BAM files from a dataset.

IGV relies on indexing of the BAM and VCF files content for faster browsing, and lighter memory load on the web browser. VCF files must always be stored together with the .TBI index file for IGV browser, but for BAM files IGV is able to generate the .BAI index files on-the-fly. However, if you store large BAM files, the browsing process will initiate much faster if the files are stored with their pregenerated .BAI index files (using "BAM files with index" or "BAM files with BAI and sampleId" forms).

IGV allows visualisation of max 10 files at a time, to protect browser performance. Image 2 shows the track selection interface for BC|INSIGHT Embedded IGV.

For more help and information about IGV.js visit https://github.com/igvteam/igv.js/wiki.

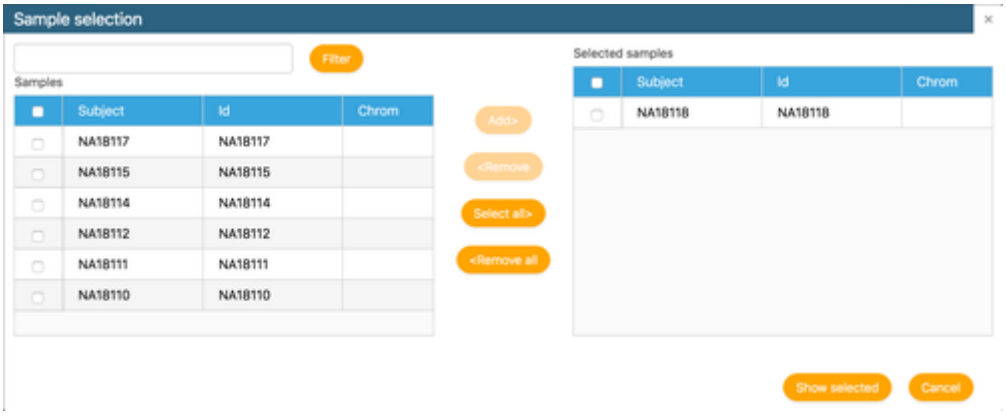


Image 2. Selecting BAM files for visualisation.

BC|INSIGHT - 4.6.5 Data service for IGV and LocusZoom

User roles
BC|INSIGHT user

Table of contents:

- BC secure data services

BC secure data services

LocusZoom and IGV are commonly used embedded genomic data browsers, and they require genetic annotations to view and locate correctly the features in data tracks. Many web sites and applications utilising LocusZoom and IGV resources rely on publicly available databases and data services specifically built for these applications. In BC|INSIGHT the protection of data and research activities on the platform require more secure provision of these data services.

BC Platforms hosts a bespoke encrypted cloud data service for those BC|INSIGHT instances, where local IT policy allows the use of external resources. A secure gateway is provided for bulk gene annotations and references to be accessed from the BC|INSIGHT web applications. This configuration is recommended to be used to save on local storage space (human genomic reference data can be huge), and ease of access configurations. By default BC|INSIGHT embedded LocusZoom and IGV applications rely on access to this service.

However, if local IT policy prohibits the use of external resources completely, there is an option of installing the data service locally. With this option there is an additional requirement for data storage that is locally accessible to BC|INSIGHT.

Location of the data service is configured in the BC Registry. Please contact BC support team for more details about how this configuration has been setup in your particular BC|INSIGHT instance.

BC|INSIGHT - 4.7 Embedded analysis tools

User roles
Analyst

Table of contents:

- Embedded analysis tools
- Recommended analysis tools supporting large datasets
  - Summary statistics
  - Population stratification
  - IBS/IBD estimation
  - Association
  - Haplotyping, LD, and haplotype association
  - Meta-analysis
  - Result annotation / visualisation
  - Epistasis
  - CNV
  - Family based
  - Mendelian Error Checks
  - Scripting / command line use
  - Data export formats

Embedded analysis tools

BC|INSIGHT supports the following list of embedded analysis packages. The list is not comprehensive but it gives an idea of what kind of analytical packages the analysis interface typically provides. Each tool is equipped with it's own web page for choosing data and parameters, and possibly multiple page templates for various workflows using the same tool.

Recommended analysis tools supporting large datasets

Summary statistics

- PLINK v1.07 and v1.9

Population stratification

- PLINK v1.07 and v1.9
- KING
- Eigenstrat and SmartPCA

IBS/IBD estimation

- PLINK v1.07 and v1.9

Association

- PLINK v1.07 and v1.9
- PLINK/SEQ
- GRANVIL (exome data only)
- EPACTS
- VAT

Haplotyping, LD, and haplotype association

- IMPUTE v2
- PLINK v1.07 and v1.9

Meta-analysis

- METAL

Result annotation / visualisation

- PLINK v1.07 and v1.9
- IGV
- SNPeff

Epistasis

- PLINK v1.07 and v1.9

CNV

- PLINK v1.07 and v1.9

Family based

- PLINK v1.07 and v1.9
- FBAT
- Single-point linkage (Merlin)

Mendelian Error Checks

- PLINK v1.07 and v1.9

Scripting / command line use

- R
- GenABEL

Data export formats

- VCF
- PLINK binary
- PLINK text

BC|INSIGHT - 5. Administration

User roles
Administrator
Database owner

Child pages:

- BC|INSIGHT - 5.1 User management
- BC|INSIGHT - 5.2 User role management
- BC|INSIGHT - 5.3 Managing dataset ownership
- BC|INSIGHT - 5.4 User group management
- BC|INSIGHT - 5.5 Browsing event log information

Table of contents:

- Administration application

Administration application

The Administration tools are located in their own application accessible from the user menu (see Image 1.). Depending on the user role, different functions are available for user and use role management, and dataset ownership control.

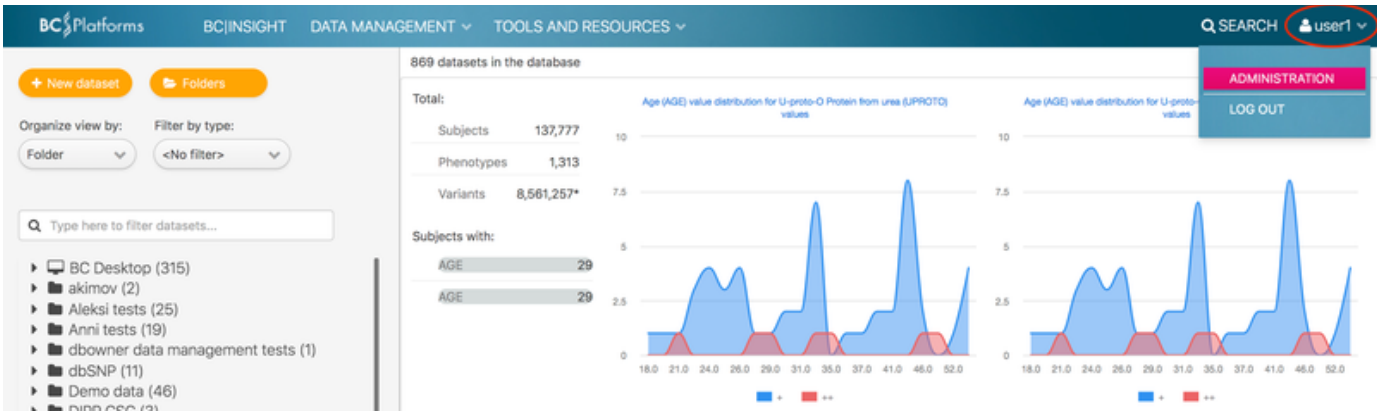


Image 1. Accessing Administration application.

BC|INSIGHT - 5.1 User management

User roles
Administrator
Database owner

Table of contents:

- User list and associated tools
- Adding a new user

User list and associated tools

> **Note**
> The tools for password and role management are only available if password and role management is local.

On the left hand panel Administrator or Database owner user is able to select the 'Users' section to list all users in the system. Selecting a single user entry and clicking 'Edit' button opens the user details dialog for editing. Mouse right click opens a context menu for manipulating following user attributes:

- **Change password** (Only available if user management is local) Administrator user is able to reset password for all users except other Administrator users and Database owner. Database owner can change password for any user.
- **Add to group** Database owner is able to add user to an existing User group (dataset permission group).
- **Remove from group** Database owner is able to remove a user from a User group (dataset permission group).
- **Copy permission from another user** Database owner is able to copy all dataset permissions from one user to another user.
- **Set user roles** (Only available if user management is local) Administrator can add and remove roles to user, except for Administrator and Database owner roles. Database owner is able to add/remove any role.

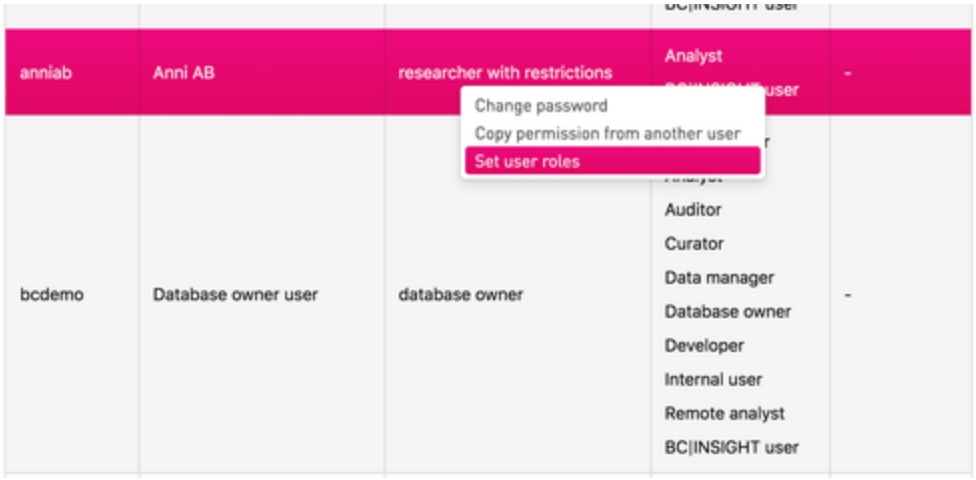


Figure 1. Tools in Administration / Users

Adding a new user

> **Note**
> This feature is only available if user password and role management is local.

Both Database owner and Administrator -roles can add new users. See the Figure 6 for information about adding a new user:

1. **User ID** can contain 2 – 16 alphanumeric characters and underscores (_).
2. **Full name** for descriptive information about the user (max. 65 characters).
3. **User's password** for typing the password

- A password has to be 7 (min) - 32 (max) characters long, contain both capital (A-Z) and lower case letters (a-z) and numbers (0-9) in combination. In addition, a good password is not easy to guess and it shouldn't be written on any paper.

1. **Verify user's password** for re-typing of the password

- if there is not match between the passwords a red exclamation mark (!) sign is shown

1. If a password violates specification a pop-up help will be shown

- Close the pop-up window by clicking mouse cursor on the window.

1. Pick one of the profiles matching to user's profile summarized in Table 1 and Table 2
2. Press Save to store the new password.

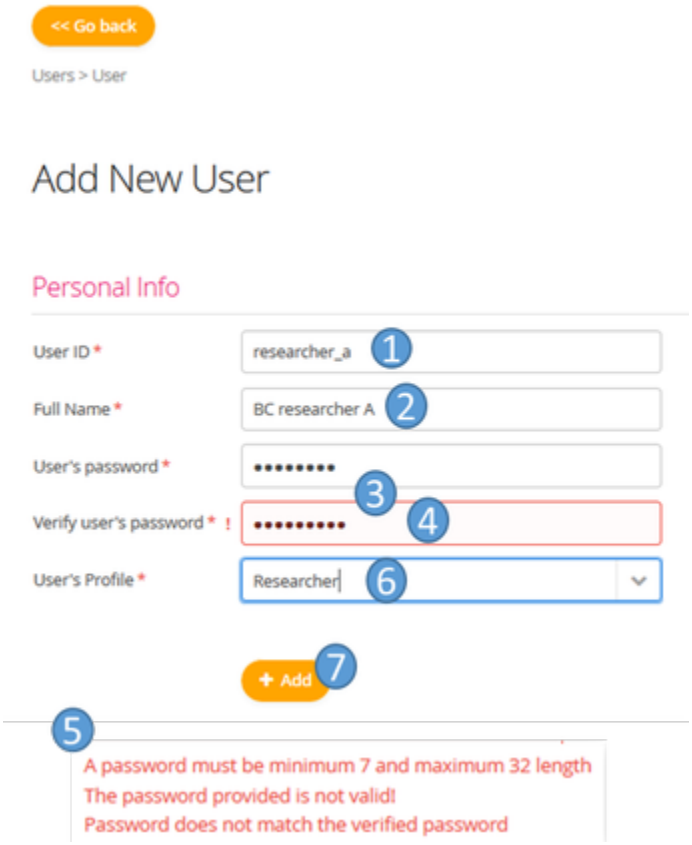


Figure 2. Specifying the user account information in Administration.

BC|INSIGHT - 5.2 User role management

User roles
Administrator
Database owner

Table of contents:

- User roles and profiles
  - User roles for tasks
  - Default BC|INSIGHT User Profiles
- Managing roles

User roles and profiles

BC|INSIGHT comes with a granular user role management system. User role defines which functionalities or permissions a user with the role will have in the BC|INSIGHT system. These roles are mostly task-based, allowing access to specific tools and features, based on user role definitions. The table here lists the inbuilt roles and the tasks and features in the system these roles enable.

User roles stack. One user may possess multiple roles, allowing generation of user profiles with specific permissions and task privileges.

User roles for tasks

Role	Role description	User tasks enabled by role	Respective BC|INSIGHT features
BC|INSIGHT user	Base role for browsing the data shared with the user	• Browse data • Search data	• Data Navigator • Data Search
Analyst	User performs research tasks on the data, including filtering, saving views, combining data, and analysing it with scripts or tools	• Create subsets and joins • Cancel jobs	• Subset -tool • Cancel job -tool • R/SAS script -tool restricted to shared scripts • Analysis tools

		• Run R/SAS scripts from shared resource • Run enabled analysis tools	
Internal user	Trusted user, Internal user is organisation in-house trusted user, allowed to export data and files from the system.	• Export data • Export result files • Cancel jobs	• Export -tools • Result download • Cancel job -tool
Developer	User creates script-based algorithms for use within static workflows. These algorithms are shared via BC|INSIGHT with other users	• Create R/SAS script datasets, add scripts • Run R/SAS scripts inline • Cancel jobs	• Create script datasets • R/SAS script -tool full access • Cancel job -tool
Remote analyst	User has access to BC|INSIGHT data API for downloading data for remote tasks, effectively possessing same export privileges as 'internal user'. This role is also used to authorise external automatons.	• REST API requests for data	• REST API authentication
Data manager	Diverse data and access management -related organisatory tasks.	• Create/Edit datasets • Manage folders • Create/Edit forms • Create/Edit Dictionaries • Create patient questionnaires • Cancel jobs • Import data • Delete datasets • Delete data entries	• New dataset -tool • Folder -tool • Structure Editor • Dictionary Editor • Web form editor • Cancel job -tool • Import Wizard, Upload tool • Trash, Empty Trash • Delete selected rows
Curator	User performs data cleanup tasks and imports data to the system. Data is manipulated by BC|INSIGHT modules during harmonisation.	• Harmonise incoming data	• AutoCurator
Auditor	User accesses audit logs for data management and curation tasks. User accesses system logs for security events	• Read audit trail per dataset • Read security event audit trail	• Audit log table
Administrator	User performs user management tasks, and assigns roles to the users	• Create new user[1] • Change password /block user[1] • Change user roles[1]	• User administration[1] • Role administration[1]
Database owner	User has full privileges over data governance in the database	• Change data ownership • Give Administrator role to users[1] • Give Database owner role to users[1] • Create user groups • Manage group members • Change dataset permissions	• Dataset owner administration • Role administration[1] • User group administration • Dataset permissions

[1]If the BC|INSTANCE authentication and role management is connected to an external role management service, the features are not available on the BC|INSIGHT Administrator -tools.

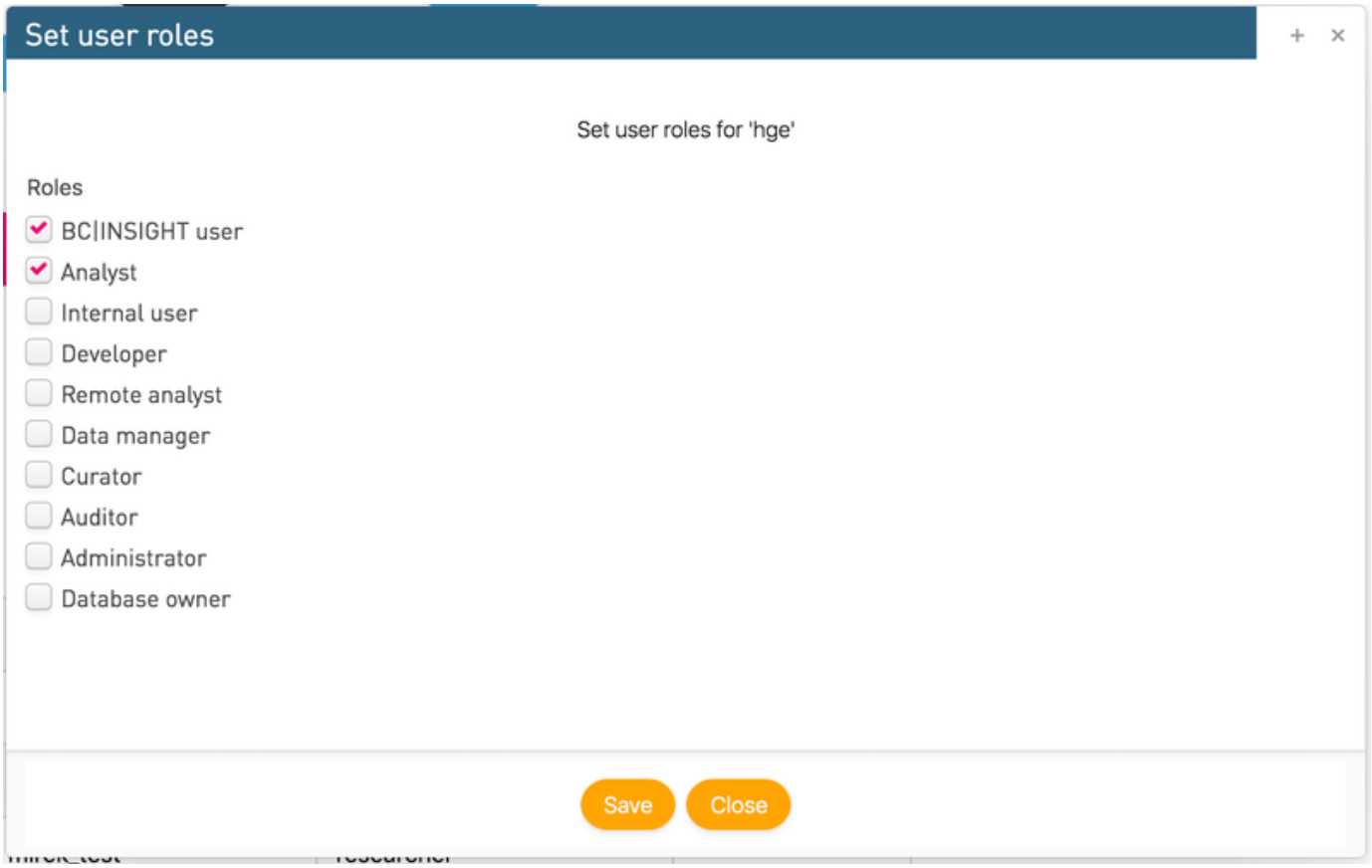
Default BC|INSIGHT User Profiles

It is possible to define different combinations of roles as new user profiles, using custom profile naming. System comes with the following defaults. User role and profile management can be externalised to a role management and authorisation service.

User Profile	Roles
All profiles	BC|INSIGHT user
Data viewer	• Auditor
Researcher with restrictions	• Analyst
Researcher	• Analyst • Internal user • Developer • Remote analyst • Data manager • Curator
Administrator	• Analyst • Internal user • Developer • Remote analyst • Data manager • Curator • Auditor • Administrator
Database owner	• Analyst • Internal user • Data manager • Curator • Auditor • Administrator • Database owner

Managing roles

As Administrator or Database owner, user can access in the ADMINISTRATION tool section called Users. This view lists all users in the system and their respective user profiles and associated roles. In this view the administrative user is able to right click on a user entry, and open the "Set user roles" -dialog.



BC|INSIGHT - 5.3 Managing dataset ownership

User roles
Database owner

Dataset ownership

Database owner is able to grant ownership of a dataset to another user in BC|GENOME. The datasets list tool lists all datasets in the database. See Figure 1 for information for the tools in the user interface.

1. Datasets List to view a list of datasets in BC|GENOME
2. Menu to specify up to 600 lines per page
3. Tools for browsing rows shown in the browser page
4. Select the dataset row to right-click your mouse button for the Change ownership tool.

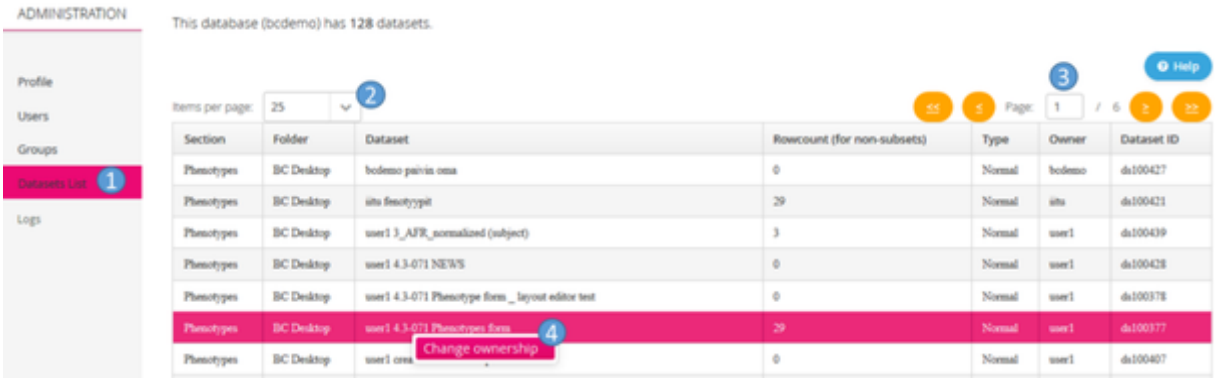


Figure 1. Administration / Datasets List.

BC|INSIGHT - 5.4 User group management

User roles
Database owner

Table of contents:

- Add new groups
- Edit groups
- Managing users in groups

In BC|INSIGHT dataset user permissions can be managed in groups by Database owner. Either read or read/write permissions for a dataset can be granted to a group of users. Technically a group is a database role to manage SQL-level access. One user can belong to several groups.

Add new groups

1. Group name can contain 1 – 128 alphanumeric characters and underscores (_), and the name cannot start with a number.
2. Description for descriptive information of a group.
3. Press Add to create a group.

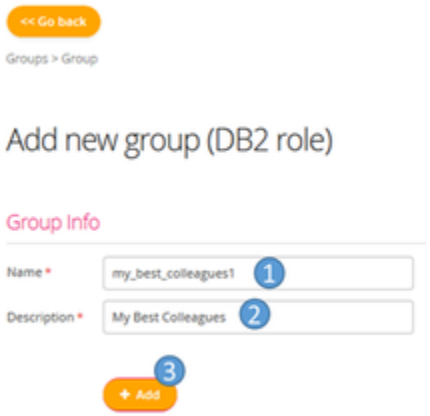


Figure 1. Adding new user group.

Edit groups

1. Use Administration / Groups to access the user group tools
2. User groups can be added using Add group tool
3. When a group has been specified the right-side mouse button will open the action menu for

- **Change description** for editing the group description
- **Edit users list** for either adding of removing users from groups

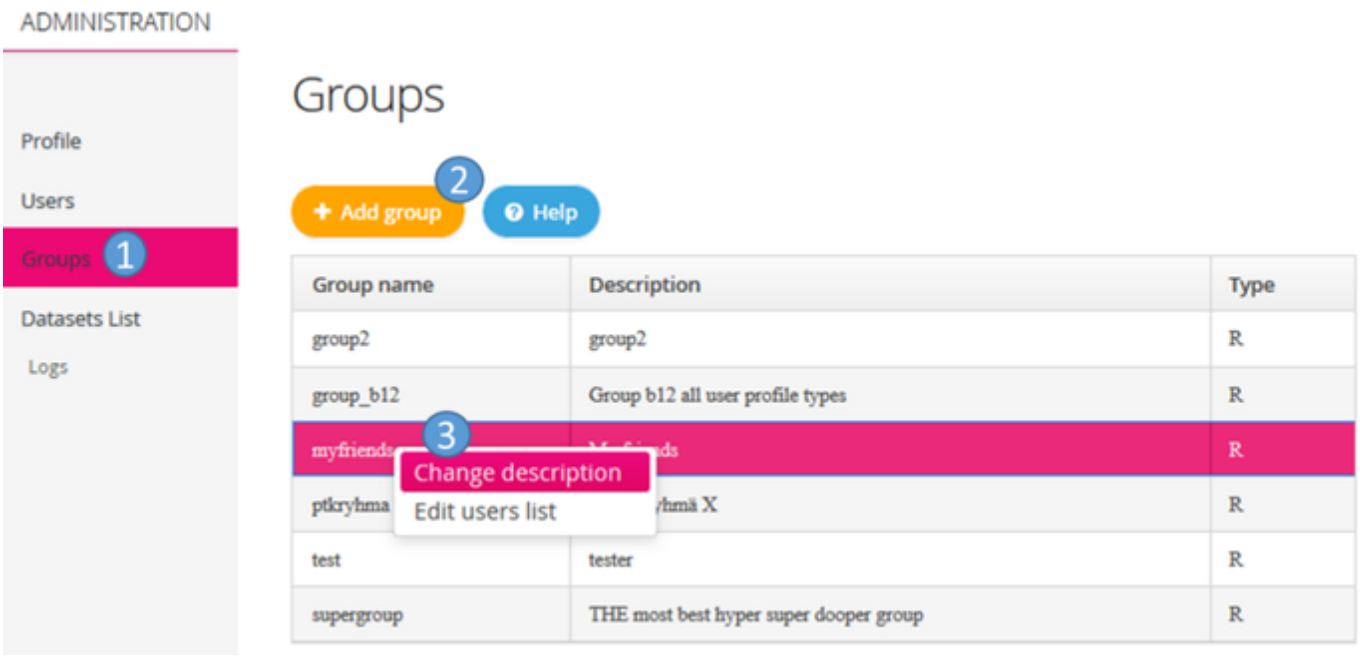


Figure 2. Group management tools.

Managing users in groups

1. Select one or more (*CTRL+click*) subjects to be added into a group
2. Use the right pointing arrow to add users to a group, and left pointing arrow to remove users from a group.
3. You can select one or more subjects to be removed from a group
4. Press OK to store changes

- If you want to remove unnecessary groups from graphical user interface please contact support@bcplatforms.com

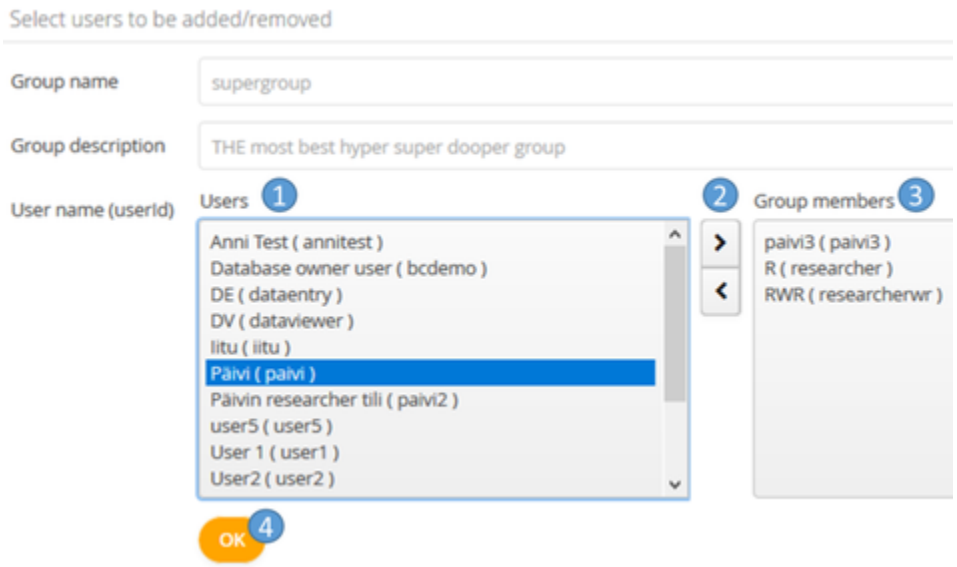


Figure 3. Administration / Groups: Edit user lists

BC|INSIGHT - 5.5 Browsing event log information

User roles
Auditor

Log view

The Logs tool is an administrative tool for viewing changes that are made in the BC|INSIGHT database. Auditor can view system logs using Administration / Logs tool by events. Logs shown in BC|INSIGHT are collected from system log files of the BC|INSIGHT application server in

- /var/log/bcos/messages
- /var/log/bcos/gened.log
- /var/log/httpd/ssl_access_log
- /var/log/httpd/ssl_error_log
- /var/log/bcos/vaadin-servlet.log
- /var/log/bcos/vaadin-events.log

Figure 1 shows the features of Administration / Logs:

1. System logs for viewing change log information
2. Refresh tool for updating log information
3. Clear filters in the log table
4. Click for menu to specify columns in Logs table:

- Event
- Timestamp
- Grantor
- Grantee
- Database
- Message
- Job id
- Client IP
- Type

1. Filter the Logs view by events, see the list of events in Table 1.
2. Press the column title to sort information in either ascending or descending order, or type field value to be searched.
3. Press the Event field to view the summary collected from column fields.

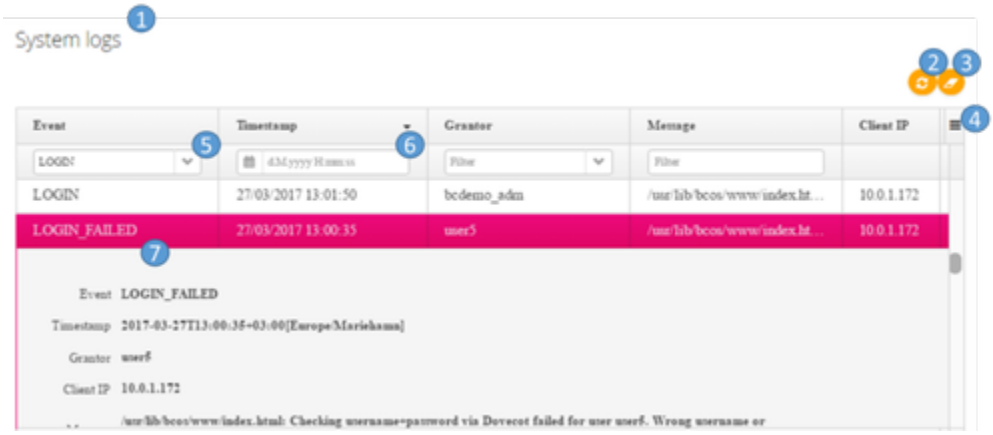


Figure 1. Administration / Logs for viewing system logs.

BC|INSIGHT - 6. Use-cases and HOWTOs

User Roles

Miscellaneous

Child pages:

- BC|INSIGHT - 6.1 Store BAM and FASTQ files
- BC|INSIGHT - 6.2 IMPUTE2
  - BC|INSIGHT - 6.2.1 Running IMPUTE2
  - BC|INSIGHT - 6.2.2 Uploading IMPUTE2 results
  - BC|INSIGHT - 6.2.3 Analysis of imputed genotypes
  - BC|INSIGHT - 6.2.4 Optional imputation features

- BC|INSIGHT - 6.2.5 X chromosome imputation
- BC|INSIGHT - 6.2.6 Troubleshooting IMPUTE2

- BC|INSIGHT - 6.3 PLINK analysis
  - BC|INSIGHT - 6.3.1 Example analysis with PLINK
  - BC|INSIGHT - 6.3.2 PLINK results
  - BC|INSIGHT - 6.3.3 Quantitative traits with PLINK
- BC|INSIGHT - 6.4 Upload PLINK genotype files
  - BC|INSIGHT - 6.4.1 Import genotypes from PLINK files
  - BC|INSIGHT - 6.4.2 Importing pedigree and affection status as PLINK files
  - BC|INSIGHT - 6.4.3 Import marker map from PLINK files

List of example use-cases and common tasks

From this chapter users are able to find specific use-cases and workflows for typical tasks performed in the system. Use these examples as tutorials as to how to perform your own related workflows and data management tasks.

BC|INSIGHT - 6.1 Store BAM and FASTQ files

User roles
Data manager

BAM and FASTQ files have their own storage and collection templates in BC|INSIGHT. Follow these steps to upload these types of files to the database.

FASTQ

The FASTQ format is a text-based format representing sequencing data in single-letter codes. The format contains also the quality scores that are used as input for various alignment and mapping algorithms (BWA and others). FASTQ files are stored in the BC|INSIGHT database in the FASTQ Files datasets. The variables in FASTQ dataset are listed and described in Table 1.

VARIABLE	DESCRIPTION	SOURCE	REQUIRED
SUBJECT	Subject ID		Yes
ID	File ID	Either instrument generated, or otherwise unique identifier file ID, can be the same as file name	Yes
FASTQ1_FNAME	FASTQ file name #1	Optional, a short name for FASTQ file	No
FASTQ1	FASTQ file #1	Folder path and file information of FASTQ files (folderID(s)/FASTQ file)	No
FASTQ2_FNAME	FASTQ filename #2	Optional, a short name for paired-end reads, otherwise leave empty	No
FASTQ2	FASTQ file #2	Folder and file information for paired-end reads in a FASTQ file	No
PLATFORM	Sequencer	Platform used for creating reads: 1 = 454, 2 = LS454, 3 = Illumina, 4 = Solid, 5 = ABI_Solid, 6 = CompleteGenomics (GC)	No

Table 1: Variables in the FASTQ File form.

Uploading FASTQ files

1. Create a dataset for storing the reference file for FASTQ files (Figure 1)
2. Create a FASTQ reference file
   a. For the FASTQ files form the combination of Subject ID and File ID needs to be unique.
   b. Specify FASTQ files #1 and #2 (the latter only when paired-end reads) with the folder path information, see Table 3 and 4
      i. In the case of external file system enquire the folder path information from your server IT team.
      ii. Normally, and In this example, the FASTQ files are transferred to user's upload folder defined in the reference file.
   c. Save the reference file either in the tab-delimited txt or csv format, and **copy** it to the root of your upload folder, or to a suitable subfolder
3. Upload reference information to a dataset using the FASTQ and BAM file converter
   a. Go to Data Upload files from server

b. Choose the converter e.g. "Upload BAM/FASTQ files to database using a tabulator separated reference file", depending on the column separator in the file
c. Check the name of the upload directory, where your **reference** file is
d. Type the search string for the FASTQ reference file
e. Check the Removal of original files option (if using external file system you need to skip this step)
f. Press Continue to check the upload summary
g. Press Upload to submit the job to the queue
h. When the job has completed check the upload report in Tools / result archive.

SUBJECT	ID	FASTQ1	PLATFORM
NA06986	SRR0000001_filt.fastq	fastq/NA06986_SRR0000001_1filt.fastq	3

Table 3: Reference file for single-end FASTQ files

SUBJECT	ID	FASTQ1	FASTQ2	PLATFORM
NA06986	SRR0000001_1filt.fastq	fastq/NA06986_SRR0000001_1filt.fastq	fastq/NA06986_SRR0000001_2filt.fastq	3

Table 4: Reference file for paired-end FASTQ files

BAM

The BAM format is a compressed binary version of the Sequence Alignment/Map (SAM) format that describes nucleotide sequence alignment. BAM files are often companied with BAI and BAS files: BAI file contains indexing for fast reading of the BAM file, and BAS file shows statistics about each alignment. For more information on BAM format, visit http://www.1000genomes.org/category/bam. BAM files are stored in the BC|INSIGHT server in BAM Files dataset. The BAM files dataset are created using the BAM Files form, and the variables are listed and described in Table 2.

VARIABLE	DESCRIPTION	SOURCE	REQUIRED
SUBJECT	Subject ID		Yes
ID	BAM ID	Either instrument generated, or otherwise unique identifier file ID, can be the same as file name	Yes
CHROM	Chromosome	Chromosome of sequence data. If a BAM file includes data from several chromosomes leave this field empty.	No
BAM_FNAME	BAM File name	Name of the BAM file. If left empty, the system creates it automatically.	No
BAM	BAM File	Folder path and file information of BAM files (folderID(s) /BAM file).	No

Table 2. Variables in the BAM files form.

Uploading BAM files

1. Create a dataset for your BAM files
2. Create a BAM reference file (Table 5)
    a. In the case of external file system enquire the folder path information from your server IT team
    b. Normally, and in this example, both BAM reference file and BAM files have been transferred into a subfolder in user's upload folder
    c. Save the reference file either in the tab-delimited txt or csv format
3. Use the BAM reference file to upload the BAM files information to the BAM files dataset
    a. Go to Data Upload files from server
    b. Choose the converter e.g. "Upload BAM/FASTQ files to database using a tabulator separated reference file", depending on the file column separator
    c. Check the name of the upload directory, where you have your **reference** file
    d. Type the search string for the BAM **reference** file
    e. Check the Removal of original files option (if using external file system you need to skip this step)
    f. Press Continue to check the upload summary
    g. Press Upload to submit the job to the queue
4. When the job has completed check the upload report in Reports

SUBJECT	ID	BAM
NA06986	NA06986	bam/NA06986.bam

Table 5. Reference file for BAM files, where the BAM files are stored in user's upload folder 'bam'.

BC|INSIGHT - 6.2 IMPUTE2

User roles
Analyst

Child pages:

- BC|INSIGHT - 6.2.1 Running IMPUTE2
- BC|INSIGHT - 6.2.2 Uploading IMPUTE2 results
- BC|INSIGHT - 6.2.3 Analysis of imputed genotypes
- BC|INSIGHT - 6.2.4 Optional imputation features
- BC|INSIGHT - 6.2.5 X chromosome imputation
- BC|INSIGHT - 6.2.6 Troubleshooting IMPUTE2

IMPUTE2 within BC|INSIGHT

IMPUTE2 (see https://mathgen.stats.ox.ac.uk/impute/impute_v2.html) is a program for statistically inferring unobserved genotypes, based on a set of known reference haplotypes. Running impute within BC|INSIGHT is automatically performed in segments of suitable size, such that many segments are imputed concurrently in the computation environment of the customer. After performing the imputations, resulting genotypes from the segments of each chromosome are combined in to a single result file, and files from multiple chromosomes are collected into a summary folder for easy upload to a BC|INSIGHT dataset (imputed genotyped need to be in a dataset to be later analyzed with BC|INSIGHT). Imputations can be started for a single chromosome or multiple chromosomes from a single GUI page. It is a good practice to first perform a small test run with a single chromosome (e.g. 22), or a segment of a chromosome, before starting imputations for the complete genome. In the following, we will provide detailed instructions on running Impute within BC|INSIGHT.

Prerequisites

- IMPUTE2 (version 2.2.2 or later) must be installed to your server (BC Platforms cannot automatically download and install it due to licensing reasons). If it is not installed, please download the Impute program from IMPUTE's web page: https://mathgen.stats.ox.ac.uk /impute/impute_v2.html#download_impute2,file "Linux (x86_64) Static Executable". Copy the file to your upload folder using BC file transfer tool. After the file is copied, please inform BC support (support@bcplatforms.com), and we will configure it to be run on the server.
- Genotype data has been uploaded to a BC dataset.
- Genotype data can contain SNP markers and indels. Indels must be coded as I/D.
- The genotypes must be strand-aligned with your reference haplotype panel (typically the forward strand of the genome build). As impute automatically performs strand alignment for SNPs for markers where this can be done unambiguously (ones with alleles A/C, A/G, C/T or G/T), this is not required if the data does not have A/T and C/G markers.
- If X chromosome is to be Imputed, a pedigree dataset or a phenotype dataset with form "gender list" must exist to provide gender information for the imputed subjects.

BC|INSIGHT - 6.2.1 Running IMPUTE2

User roles
Analyst

Child pages:

Table of contents:

- Basic parameters

Basic parameters

In the following, we describe the minimal steps that are needed to run Impute within BC|INSIGHT. Following these steps should be enough for running Impute in most cases. The section *Optional features* contains details on options in the Impute GUI that can be used by advanced users if needed.

1. **Open the genotype dataset** that you want to Impute in the dataset navigator (this must be done before opening the Impute GUI page).
2. Open Impute GUI
    a. Search for IMPUTE2 in the ANALYIS tab and click it open
3. **Give a "Run title",** to enable later recognizing different Impute runs in the result archive.
4. **Select a marker map** to specify coordinates for the markers in your genotype data.

a. The marker map must be based on the same genome build as the reference panel you're going to use in the imputation. Most current reference panels are based on NCBI genome build 37. The build is also visible for most marker maps, e.g. *NCBI dbSNP Build 135 (Nov 2011, hg 37.3*) and also in the reference panel name, e.g. *1000 Genomes Phase I integrated variant set v3 (Marc h 2012, NCBI build* 37).
b. Probably a marker map corresponding to you genotyping chip is found in the set of pre-installed marker maps in BC|INSIGHT, or you have uploaded a marker map corresponding to the your genotyping chip; in these cases it is best to use these maps.
c. If no specific marker map is available for you genotypes, it is generally OK to use the most recent version of dbSNP marker map that is based on the same genome build as the reference panel. Depending on the genotyping chip, some markers may not be found in dbSNP. Usually the number of such markers is rather small, and this should not hamper imputation accuracy much.
d. Warning: the marker map corresponding to the reference panel should NOT be used in imputation run; that map is only to be used for analysis of imputed data.

5. **Select chromosomes to be analyzed** in the "Include only chromosome(s)" text box:
   a. list of chromosomes is written as a comma-separated list of individual chromosomes, or chromosome ranges, e.g. "1,4-7". To impute the complete genome, write "1-22,X,XY".
   b. Warning: imputation of the X chromosome imposes some special requirements on the used marker map, and also requires a input data set file for specifying the gender of the subjects to be imputed. Usually X chromosome imputation needs to be performed as a separate run with a dedicated marker map. For details, see section "X chromosome imputation" at the end of this document.

6. **Select reference panel** under "phased reference"

1. Note that the NCBI genome build of the reference panel must match the build of the selected marker map
2. Although older reference panels are divided to separate population-specific panels, in more recent reference panel versions the subjects from all populations have been combined into a single reference panel. This is sensible because Impute can automatically choose a subset of the reference panel haplotypes that best match the genotypes of each individual.
3. Reference panels distributed on IMPUTE2's web page (https://mathgen.stats.ox.ac.uk/impute/impute_v2.html) are pre-installed in BC|INSIGHT. We aim to add new versions of reference panels as soon as possible when they are published.
4. If a certain reference panel is not shown in the GUI, please contact BC support.
5. User's own reference panels can currently only be installed as customization work. Please contact BC support to request installing custom reference panels. In an upcoming version of BC|INSIGHT, there will be a possibility to upload reference panels.
6. **Start the Imputation** by clicking "Run" at the bottom of the page
   a. If the panel containing the run button is shown in red, some parameters (also shown in red) will need to be modified before the run can be started.
7. Monitor the progress of the runs from the BC|SNP queue page
   a. link to the queue page is available from the page that opens after starting the run
   b. Alternatively, see DATA MANAGEMENT / QUEUE to view the progress of the run
   c. Number of MCMC iterations completed / remaining is shown in the queue page, once imputation has progressed far enough. Typically, imputing each segment takes several hours, at the least
8. **Verify imputation results** from the result archive (after all segments have been completed)
   a. Result archive is accessed from DATA MANAGEMENT / RESULT ARCHIVE
   b. The result files are as follows:
      i. *imputed_genotypes.bcos.txt.gz*: main result file containing imputed genotypes, converted to probabilistic BC format. This file can be directly uploaded to a BC|INSIGHT dataset (see next section).
      ii. *snpinfo:* info file produced by Impute, converted to BC format. This file can be uploaded to a BC marker info dataset (having form *IMPUTE SNPInfo*).
      iii. *MPUTE_summary*: summary file outputted by Impute. It is recommended to browse through this file to see that everything went as expected.
      iv. *markers_not_in_map.dat:* list of markers in the genotype dataset that were not found in the selected marker map. These are not used in imputation, and will not appear in the imputed data file (unless they are present in the reference panel, in which case naturally an imputed version of the genotype will be present in the result)
      v. *skipped_monomorphic_markers.txt:* list of monomorphic markers in the genotype data that were not used for the imputation. As Impute requires both allele alternatives to be specified in the input data even when the marker is monomorphic, such markers cannot be included in input. The missing allele is deduced from the reference panel where possible, but this is not always possible, e.g. due to stranding errors.
      vi. *marker_name_conversion.bcos.txt:* Mapping of marker names in the genotyped data to the marker names used in the reference panel. Unless option "use genotyped marker names in result" is chosen, output 4 will use marker names of the reference panel, instead of the original names.
      vii. Note that when imputation is performed in segments (as is usually the case), the above-mentioned files from individual segments will be automatically combined to chromosome-wide files.

BC|INSIGHT - 6.2.2 Uploading IMPUTE2 results

User roles
Analyst

Table of contents:

- Uploading imputation results
  - Uploading imputed genotypes from a single chromosome to a probabilistic dataset:

- Uploading imputed genotypes from multiple chromosomes to a probabilistic dataset:
- Uploading marker info

Uploading imputation results

In order to be used in analyses within BC|INSIGHT, the result files need to be uploaded from the result archive to a BC|INSIGHT dataset. (it is not possible to automatically upload imputation results to a dataset as part of the imputation job). Imputation results are typically stored in a probabilistic dataset, although they can also be stored to a deterministic dataset to save disk space, by just taking the most probable genotypes (naturally, information is lost in this conversion). It is also possible to upload the marker information produced by Impute, into a separate marker info dataset.

Uploading imputed genotypes from a single chromosome to a probabilistic dataset:

(Note, data in compressed dataset cannot be used for subsetting)

1. In the landing page of BC|INSIGHT use the New dataset button to create a new dataset using the form *Compressed imputed SNPs with 3-digit precision*.
2. Go to the result folder of the Impute job under the result archive.
3. Click on the blue upload arrow next to the file "imputed_genotypes.bcos.txt".
4. On the next page, select converter "Imputed SNPs with probability" and click "Continue".
5. On the next page, check the list of files to be uploaded and click "Upload" to schedule the upload job in the BC|INSIGHT queue.
6. Progress of the upload can be tracked in the BC|INSIGHT queue page; after the upload is complete, an upload report appears in the result archive.

Uploading imputed genotypes from multiple chromosomes to a probabilistic dataset:

1. In the landing page of BC|INSIGHT use the New dataset button to create a new dataset using the form *Compressed imputed SNPs with 3-digit precision*
2. Go to the result folder of the Impute job and go to the "summary" subfolder.
3. Click on the blue upload arrow next to one of the files having name like "chr22_imputed_genotypes.bcos.txt" (any chromosome can be chosen).
4. On the next page, select converter "Imputed SNPs with probability", replace the search string with "chr*_imputed_genotypes.bcos.txt" to match all chromosomes and click "Continue".
5. On the next page, check that files for all chromosomes are included in the list of files to be uploaded and click "Upload" to schedule the upload jobs in the BC|INSIGHT queue.
6. Progress of the upload can be tracked in the BC|INSIGHT queue page.
7. After the uploads are complete, an upload report appears in the result archive.

Uploading imputed genotypes to a deterministic dataset:

Note, the subset functionalities have been supported with a deterministic dataset since genotypes have been stored into a database table
Similarly as for probabilistic data set above, but with following changes:

1. For the target dataset, use form "Compressed ACGT coded SNPs".
2. For the converter, select either "Imputed SNPs (uses most probable genotype)", or "Imputed SNPs (uses most probable genotype, min. quality 0.95)".
    a. The first one always gets the most probable genotype, while the latter one leaves a genotype as missing when the probability of the most probable genotype is less than 0.95.

Uploading marker info

Similarly as for genotypes above, but with following changes:

1. Create/select a marker dataset with form "IMPUTE SNPInfo"
2. Select "no conversion" as the converter

BC|INSIGHT - 6.2.3 Analysis of imputed genotypes

User roles
Analyst

After importing to a dataset, imputed genotypes can be analyzed with analysis programs integrated to BC|INSIGHT. Using analysis programs designed for probabilistic genotypes, such as Probabel and SNPtest, is recommended for imputed data. However, also analysis programs using deterministic genotypes, such as PLINK, can be used, as probabilistic genotypes are automatically converted to deterministic ones by BC|INSIGHT when deterministic genotypes are expected, by taking the most probable genotypes.
For analyses requiring a marker map, there are two options:

1. **Use a pre-installed map corresponding to the reference panel.** There are pre-installed marker map datasets in the *1000 Genomes* fol der corresponding to different versions of Impute reference panels. For instance, analysis of data imputed using reference panel "1000 Genomes Phase I integrated variant set v3 (March 2012)" should use map "1000 Genomes (Mar 2012) Impute map". Some older reference panels might not have a suitable map available; please contact BC support if a required map is not found.
2. **Use snpinfo dataset uploaded from the imputation results.** You can import the snpinfo files produced during the imputation into a BC format dataset with form "IMPUTE SNPInfo", and use this as the marker map dataset (see previous section for instructions on uploading the SNP info). Of these option 1) is preferred as less tedious. However, note that often the genotyped data contains at least some markers that have a different rs code (or other id) in the genotyped data and the reference panel. As Impute uses marker positions and alleles to map markers of genotyped data to markers of reference panel, such markers will always be used correctly in imputation. However, if option "Use genotyped marker names in result" (enabled by default in the GUI) is not used, these markers will have their original rs codes in the imputed data, and thus will not be found in the pre-installed map. In this case, using option 2) is required to provide the marker map where all markers are included.

BC|INSIGHT - 6.2.4 Optional imputation features

User roles
Analyst

Table of contents:

- Optional settings
  - Selecting regions to be imputed
  - Segmentation of imputation jobs
  - Imputation mode
  - Impute using pre-phased haplotypes.

Optional settings

There are a number of other options available in the BC|INSIGHT impute GUI that do not necessarily need to be adjusted in typical imputation runs, but can be used by more advanced users if needed. These are described below.

Selecting regions to be imputed

- parameters "restrict imputation to range" and "impute full chromosomes"
- by default, imputation is restricted to the region covered by the genotyped markers.
- user can restrict imputation to given range (in bp coordinates).
- option "impute full chromosomes" can be used to impute all markers in reference panel, instead of just the genotyped range.

Segmentation of imputation jobs

- parameters "max. imputation window size" and "use markers in flanking regions of "
- By default, imputation is done in windows of 5MB, with overlap of 250kb between neighboring windows. For very large datasets, it may be necessary to decrease the window size in order to prevent individual runs from using too much RAM memory / disk space. Even values as low as 1MB should not affect imputation accuracy.

Imputation mode

This option controls how impute is run. There are four modes of operation:

- **Impute using posterior probability distribution of haplotypes.** This is the "traditional mode of operation of Impute: consider a distribution of sampled haplotypes (using MCMC sampling) when doing the imputations, instead of using just the most probable haplotypes.
- **Impute using most probable haplotypes only.** In this mode, Impute is run twice: first phasing to estimate the most probable haplotypes and then imputation using these most probable haplotypes only. This mode is significantly faster than full MCMC estimation, but is slightly less accurate. This mode cannot be used for X chromosome imputation.

Impute using pre-phased haplotypes.

Impute into pre-phased haplotypes instead of doing both phasing and imputation. This mode is only visible in the GUI when the selected genotype dataset is already phased and the input genotypes are stored into a dataset with form *imputed haplotypes* (unfortunately, the name of

the form is a bit misleading, as no imputation has taken place yet). The program ShapeIT is recommended for pre-phasing by the authors of IMPUTE2. However, there is currently no support for pre-phasing haplotypes within BC|INSIGHT, thus please contact BC support for more information.

Phasing and imputation of sporadic missing data only. In this mode, only phasing and imputation of sporadic missing data is done. No reference panel is used. Note that phased haplotypes are outputted for each imputed segment separately, as there is no straightforward way of combining haplotypes from
separately phased segments into haplotypes for complete chromosomes (phase between markers of adjacent segments is not known). Also note that Haplotypes are outputted only in Impute format (and not in uploadable BC format), as haplotypes from individual segments cannot be stored into the same BC data set.

Imputation arguments

These options control Impute's MCMC algorithm and effective population size. The default values should be OK in most cases (for more documentation on these, see Impute's online manual: https://mathgen.stats.ox.ac.uk/impute/mcmc_options.html and https://mathgen.stats.ox.ac.uk/impute/output_file_options.html).

Output options

These options control the output from the software.

- **SNP types to be included in the output file (-os)** (default: output all SNP types). See IMPUTE's manual for details.
- **Store phasing results (-phase)** If checked, the phased genotypes are also outputted. Note that haplotypes are outputted for each segment individually, as they cannot be reliably be combined across segments.
- **Predict genotyped SNPs (-pgs)** If checked, impute also genotyped markers and replace their genotypes with the imputed ones in the result (if not checked, result will have the original genotypes for genotyped markers)
- **Use genotyped marker names in result (instead of names from reference panel)**. For genotyped markers, Impute normally outputs the original marker names into the output files.
    - However, by default the BC|INSIGHT impute driver converts the marker names already before imputation to match the ones in the reference panel, so that the marker names appearing in the imputed genotypes will correspond to the reference panel. This makes it possible to use a pre-installed marker map in subsequent analysis of Imputed data (see section "Analysis of Imputed genotypes" above). This option can be used to turn off this feature, so that the original marker names appear in the output file for genotyped markers.
    - **Output BC format results only (saves disk space)**. When this is checked, only uploadable BC format results are outputted (this is the default choice). If this is not checked, also the original result files of Impute are outputted. Note that the original impute format files also contain the markers on "flanking regions" of each imputed segment, and thus markers in the edges of segments will be present in the result files of both adjacent segments.

Reference panel

- See subsection 6: Select reference panel in section Running impute with basic arguments above.
- Some reference data sets contain population-wise minimum allele frequencies, which can be used to filter out low-frequency variants from the reference haplotypes, so that only ones meeting a MAF threshold are included in the imputation. Note that such filtering cannot be performed when using older reference panels, where these frequencies are not included.

Strand-alignment of genotyped data

These options control how strand of genotyped markers is aligned with the reference haplotypes. Ideally, the input genotypes should be already strand-aligned with the reference haplotype panel (typically the forward strand of the genome build). This is not required, if the the data only contains markers with alleles that can be unambiguously strand-adjusted (A/C,G/T,A/G and C/T). Impute automatically adjusts
strand in such markers (A/C <=> G/T and A/G<=>C/T). If option "Also align ambiguous alleles by MAF" is selected, Impute also adjusts strand in (ambiguous) A/T and C/G markers, based on allele frequencies. Note that this may not be accurate for markers with MAFs close to 0.5, and the authors of Impute do not recommend using this option. One can also upload a strand file to align the strand (see option -strand_g in IMPUTE2 documentation). The strand file need not contain all markers in the genotyped data; automatic strand correction (possibly including ambiguous markers, as described
above) is applied to any markers not contained in the strand file. Note that a strand file always applies to a single chromosome, so this option cannot be used when imputing multiple chromosomes as a single job. See "Strand alignment options" in IMPUTE's web page for more details.

Unphased reference

It is also possible to use unphased reference panels for imputation. Currently, these can only be installed as customization work. Please contact BC support to request installing custom reference panels.

Gender set

See next section "X chromosome imputation"

Additional command-line arguments

Here, you can write any command line options accepted by impute. Typically this should not be needed, as most commonly used options are already provided by the BC|INSIGHT impute driver based on the selections made by the user in the GUI. Note that for this reason, many options (such as ones governing input and output files, range to be imputed etc.) should not be given here, as they are already provided automatically by the driver.

BC|INSIGHT - 6.2.5 X chromosome imputation

User roles
Analyst

To perform X chromosome imputation, a separate dataset needs to be specified that contains sex information (for other chromosomes, sex information is not needed). The sex information is stored in a phenotype dataset, using the form "gender list". When importing the data to the gender list, the gender column must be coded as 1-male, 2-female.
The imputation of the X chromosome is done separately for the pseudo-autosomal (PAR) and non-pseudo-autosomal (nonPAR) regions of the X chromosome. In imputations done within BC|INSIGHT, these should be annotated as chromosome "X" and chromosome "XY", respectively in the chosen marker map. This poses some difficulties, as normally marker maps within BC do not make the distinction between the PAR and nonPAR regions, and markers on both are annotated as chr "X". If the genotype data has a dedicated marker map where PAR regions are annotated as "XY", those maps can be used as is. However, the generic dbSNP maps cannot be used for X chromosome imputation, as there also PAR markers are annotated as "X". Therefore, there are separate dbSNP maps for chromosome imputation within BC|INSIGHT. For dbSNP version 135, this map is called "dbSNP Build 135 chr X imputation map (NCBI build 37), available in the dbSNP folder. This map is installed with BC|INSIGHT versions 3.6-04 and later. Please note that as a different map is required, imputations for chromosome X cannot thus normally be started as part of the same run as the rest of the genome, but a separate run for is needed instead.

BC|INSIGHT - 6.2.6 Troubleshooting IMPUTE2

User roles
Analyst

Below, we list some of the most common problems encountered when running Impute within BC|INSIGHT.

- **Impute jobs are stuck in the job queue.** The most probable reason for this is that impute is not installed on your server. See section "prerequisites" at the start of this document.
- **None or only few genotyped markers map to the reference panel.** Probably the most common problem is using a marker map that is not based on the same genome build as the reference panel. Please check that the map and the panel are based on the same genome build.
- **Some of the genotyped markers are not found in the marker map.** Another common error is using a map corresponding to the reference panel (e.g. "1000 Genomes (Mar2012) Impute map"). Some markers, especially on the pseudo-autosomal regions of chromosome X and HLE regions of chromosome 6 might be named differently in the genotyped data and the reference panels, and will thus be discarded in imputations using the 1000 genomes maps. These maps should only be used for analysis of imputation results and not for the imputations; dbSNP — a genotyping chip-specific map should be used instead.
- **Problems in X chromosome imputation.** For X chromosome imputation, the marker map needs to contain markers annotated in a certain way. For problems regarding X chromosome, see section "X chromosome imputation".
- **Imputations fail to running out of memory / disk space, or they do not use all available computational resources, or** BC|INSIGHT has a certain default limit to the number of imputations run simultaneously. The optimal number is always dependent on the properties of the data (mainly number of genotyped subjects and number of markers in the chosen reference panel), which dictate e.g. how much RAM is needed. If too many imputations are run concurrently, there is a risk of running out of memory, and if too few, resources are not utilized optimally. For example, imputation of a single 5MB segment having 6000 subjects with the 1000 genomes march 2012 reference panels might take around 6GB of RAM and 17 hours of computation time on a relatively efficient processor. Naturally these numbers may vary a lot depending on marker density, segment size and other factors. If needed, please contact BC support to discuss the possibility of adjusting the number of concurrent imputations.

BC|INSIGHT - 6.3 PLINK analysis

User roles
Analyst

Child pages:

- BC|INSIGHT - 6.3.1 Example analysis with PLINK
- BC|INSIGHT - 6.3.2 PLINK results
- BC|INSIGHT - 6.3.3 Quantitative traits with PLINK

PLINK support

BC solutions support analysis with PLINK package, versions 1.07 and 1.9 (or PLINK2). See http://zzz.bwh.harvard.edu/plink/ for more information about the status of releases, and feature documentation.

PLINK is widely used to perform multiple different types of genetic statistical analyses. BC supports the following use cases with PLINK:

- Summary statistics
- Population stratification
- IBS/IBD estimation
- Association
- Haplotype association

- Annotation of results
- Epistasis
- CNV
- Mendelian error checks
- Family based association for disease and quantitative traits

BC|INSIGHT - 6.3.1 Example analysis with PLINK

User roles
Analyst

Table of contents:

- Case-control PLINK genomic association
  - Affection status
  - Covariates and Gender
  - Markers
    - Inclusion/Exclusion

Case-control PLINK genomic association

> **Note**
> BC|INSIGHT utilises PLINK software for genotype analysis, which is designed to perform a range of basic, large-scale analyses. It is a third party software, so for PLINKspecific information, go to http://zzz.bwh.harvard.edu/plink/ .

The following outlines the steps to run the ***PLINK case – control analysis***.

Select a genotype dataset/subset from the navigation tree. Go to the ***ANALYSIS*** tab of the selected dataset. Search and select ***GWAS /Association and LD / PLINK case-control analysis***. Fill in the General -section as has been described in chapter 2. For PLINK analyses you are also able to select which version of PLINK you want to use, in this section. Similarily, if needed for your analysis, Expand the **Subjects** -section from the arrow button and make you Subject filtering choices.

Affection status

In section **Affection status** you can make either a selection for case and control datasets/subsets, or select a Affection status dataset, which defines these two groups as has been described in chapter 2: cases (categorical value of 2) and controls (categorical value of 1) in variable *AFFS TAT*.

Covariates and Gender

For logistic model analysis you can select variables from a dataset in the **Covariates**-section. If you need sex information in the analysis, define it in **Gender**, as a phenotype or pedigree dataset: the dataset must use *SEX* variable to define males (=1) and females (=2).

Markers

If you need marker position information or you need to define segmentation in your analysis, you need to select a suitable map in **Marker maps**. The map must use the same marker naming as in your genotype dataset. Alternatively, If your marker identifiers are of format chr:position (implicit marker IDs), you can choose the ***Derive map information from marker labels*** option in **Marker maps**.

To split analysis by chromosome/loci to either get region -specific results, and to optimise the calculation capacity, choose ***Split analysis by chromosomes*** in ***Marker maps***. You can split the job by chromosome in the ***Split analysis by chromosomes*** -option. This will create one job run per chromosome, and allows the analysis system to distribute the jobs and run them in parallel, if parallelisation has been configured in the BC|INSIGHT system.

You can **include listed chromosomes** only by

- specifying lists of ranges using comma and dash (for example *1-22*, or *1-2,5,22*)
- specifying sex chromosomes as *X, Y* and *XY*
- specifying mitochondria as *MT*

If you want to exclude *indel* markers that have not been specified with I or D allele codes, use the ***Exclude indel markers***, which will determine this from the map information.

Chromosomal segmentation can be defined either as ***No segmentation*** – refers to a default case when one segment equals full chromosome, or as ***maximum window size*** – Chromosome segmentation can be set from 3 markers up to 10,000 markers with an overlap of 0.

Inclusion/Exclusion

You can include or exclude markers from the analysis as well. By default all markers that have sufficient data will be included in the analysis. You may have to expand the **Markers** -section using the blue arrow to see these options. In both options you can define your inclusion/exclusion lists either as text files, or as content from other datasets. For *inclusion* specifically it is also possible to simply type a comma -separated list of markers to be taken to analysis.

1. In section ANNOTATIONS – If you want to annotate your markers by gene, contact BC support at support@bcplatforms.com.
2. In section PLINK PARAMETERS – Specify PLINK parameters as outlined in the PLINK manual at http://zzz.bwh.harvard.edu/plink/.
    a. Expand the ***Filtering*** and ***Test types*** options from the arrow button and make the required selections.
3. In section PLINK PARAMETERS – EXTRA FILES/ADDITIONAL COMMAND LINE PARAMETERS - Type additional command line parameters and auxiliary files from your computer when needed.
    a. Only PLINK-compatible files can be used. For more information about file compatibility, see the PLINK manual at http://zzz.bwh.harvard.edu/plink/.
4. Select ***Run*** when all analysis datasets and parameters has been specified.
5. When a message displays, select the queue link to follow the progress of the job in the BC|INSIGHT queue system
6. Go to the ***RESULT***S page or **DATA MANAGEMENT > RESULT ARCHIVE**. For more information on results, see chapter 3.

BC|INSIGHT - 6.3.2 PLINK results

User roles
Analyst

Table of contents:

- Structure of PLINK results
- Saving PLINK .assoc files to database
    - Other filetypes for PLINK

Structure of PLINK results

Go to the ***RESULTS*** page of the dataset to which you would like to save the PLINK results. Select the PLINK analysis job you created earlier and where your results have been saved. Job folder has the name you defined at analysis time. By default the PLINK results are named "PLINK case-control analysis" or "PLINK quantitative trait analysis".

Saving PLINK .assoc files to database

There is no need for conversion in the case of PLINK .assoc association results, if you use the "PLINK association results" -form for your target dataset. Leave the upload conversion selection empty, "upload file as it is". If you have not made any modifications to the names of the data files, you should be able to upload the .assoc file directly without changing any settings on this page (Image 1).

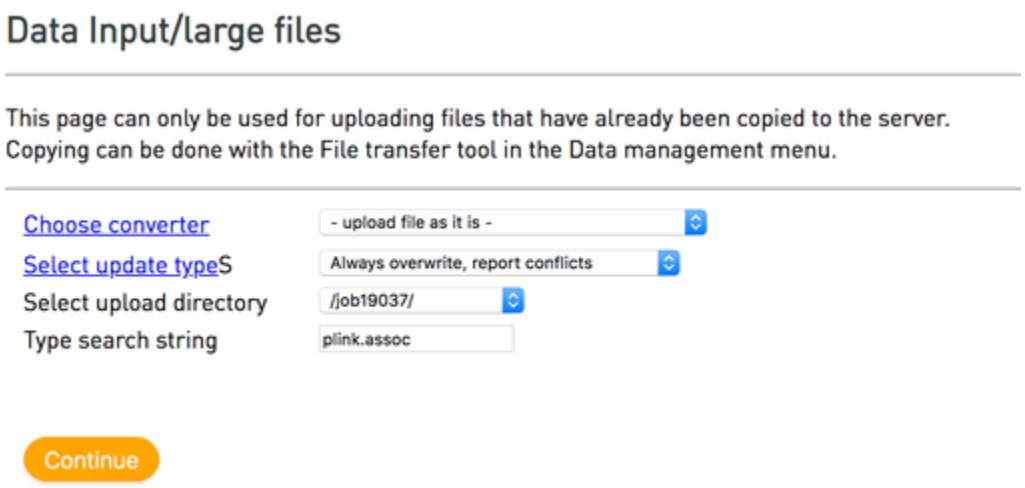


Image 1. The upload options for files in the job result folder. You typically do not need to change anything in here.

Other filetypes for PLINK

There are converters for PLINK allele frequencies (plink.frq), and PLINK bp map (.map, .bim, .tped). These file types are not generated by all PLINK analysis workflows, and some of them only with certain parameter settings. Please refer to PLINK manual to find out more about these options.

BC|INSIGHT - 6.3.3 Quantitative traits with PLINK

User roles

Analyst

Table of contents:

- Prerequisites
- Using phenotypes as QT
- Using multiQTL as QT

Prerequisites

All PLINK analyses are initiated from the genotype dataset. Make sure you have the corresponding marker map available, or that implicit marker identifiers are used in the data ('chrA:position'). Also check that the subjects to be analysed have matching subject identifiers between the genotype and the quantitative trait dataset. If you need to specify covariates in your analysis, make sure they exist, or are joined with, the QT phenotype dataset. All generic settings for genetic analyses apply in PLINK QT analysis.

Using phenotypes as QT

In order to analyse genetic association of quantitative traits stored in phenotype tables, you need to select the **PLINK quantitative trait association** from ***ANALYSIS*** tab. The phenotype dataset must define QTs as numeric values (integers or floats). In the analysis page (Image 1) you choose specific traits from the selected phenotype table to be analysed. It is also possible to select numerical covariates from the same dataset.

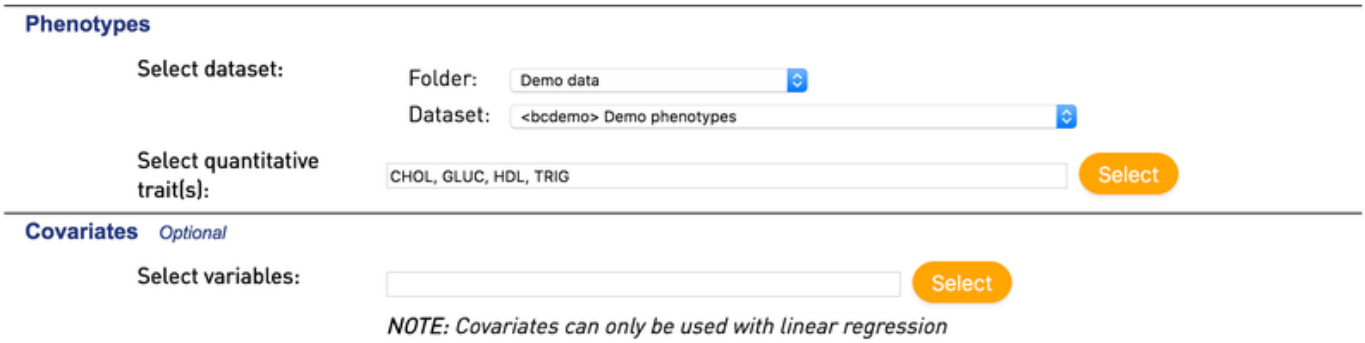


Image 1. Selection of quantitative traits from phenotype dataset.

Using multiQTL as QT

MultiQTL data can be used to provide quantitative traits for PLINK association analyses. The analysis uses all distinct traits defined by the VAR - column in the multiQT dataset (annotated BC:variable_name), and you can define the number of traits to be analysed in one analysis task. BC|INSIGHT will split the work based on total number of distinctive traits, and combines the results into one report. Note that covariates can be chosen from another phenotype dataset for this analysis, they do not need to be present in the multiQTL dataset.

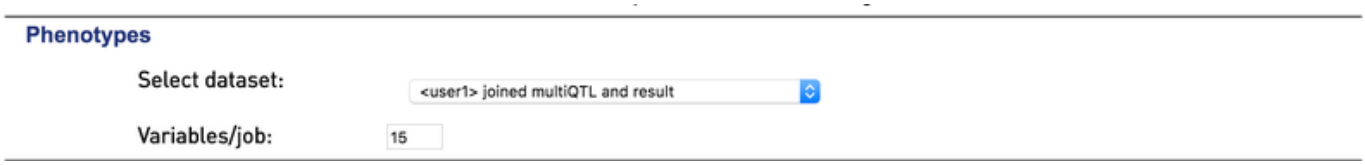


Image 2. Selecting MultiQTL dataset, and defining number of traits to be analysed in one job.

BC|INSIGHT - 6.4 Upload PLINK genotype files

User roles

Data manager

Child pages:

- BC|INSIGHT - 6.4.1 Import genotypes from PLINK files
- BC|INSIGHT - 6.4.2 Importing pedigree and affection status as PLINK files
- BC|INSIGHT - 6.4.3 Import marker map from PLINK files

Table of contents:

- PLINK data files

PLINK data files

PLINK files can be imported into the BC|INSIGHT database directly using the PLINK converter tools in the system. In addition to genotype data PLINK files can contain information about marker positions, pedigree structures, and affection status of individuals. These data are imported to BC|INSIGHT in data type -specific datasets. Table 1 describes the three different PLINK file sets. PLINK provides three differently formatted file sets for user convenience, and for compatibility with up- and downstream analysis workflows. BC|INSIGHT PLINK converter is able to any of these file sets, provided that all files are available within the set.

File name	Text or binary	Content	File set
file_name.ped	Text	pedigree structure, affection status, and the genotypes	Normal PLINK
file_name.map	Text	marker IDs, chromosomes and positions	Normal PLINK
file_name.tped	Text	marker names, chromosomes, positions, and genotypes	Transposed
file_name.tfam	Text	pedigree structure and affection status	Transposed
file_name.bed	Binary	genotype information	Binary PLINK
file_name.bim	Text	marker IDs, chromosomes and positions, allele names	Binary PLINK
file_name.fam	Text	pedigree structure and affection status	Binary PLINK

Table 1. PLINK file sets that can be used to upload data to BC|INSIGHT.

BC|INSIGHT - 6.4.1 Import genotypes from PLINK files

User roles

Data manager

Table of contents:

- Prerequisites
- Uploading data using PLINK options

Prerequisites

You need a whole file set (see Table 1 in parent page) to import the genotype data to BC|INSIGHT database. Note that the file names must be the same for all the files in the file set (e.g. myfile.ped, myfile.map).
The BC|INSIGHT system requires that all subject IDs are unique. However, in PLINK files the subject ID must be unique only within the family. If that is the case in your data, the PLINK family ID (FID) can be concatenated with the subject ID (IID) in order to make all the IDs unique. BC|INSIGHT provides tools for concatenating the IDs automatically as part of the data import.

The Genotype data is imported to the BC|INSIGHT genotype section using the Data input / large files function. Before the data can be imported to the database the PLINK files must be copied to the BC|INSIGHT server. Follow this checklist:

1. Make sure that all the file names in your PLINK file set are the same, except the postfix (e.g. myfile.ped and myfile.map).
2. Check, whether the PLINK individual IDs are unique across all the families.
3. Check that the genotypes are ACGT coded.

Uploading data using PLINK options

Copy the PLINK file set to your upload folder in the BC|INSIGHT server using the File transfer tool. Create a folder for your file set.

Next you need to create the dataset for the genotypes. Using the 'New dataset' tool in BC|INSIGHT, select the form “Compressed ACGT coded SNPs” and name your dataset. Open the newly created dataset and in the Data -tab navigate to *Tools and export -> Upload -> Files already copied to server*. As illustrated in Image 1, Select the PLINK converter suitable for your file set content, usually "PLINK (bed+fam+bim, ped+map or tped+tfam)". If the Individual IDs in the PLINK data are not unique, you need to choose the converter with Family ID concatenation option “PLINK with family-subject concatenation”.

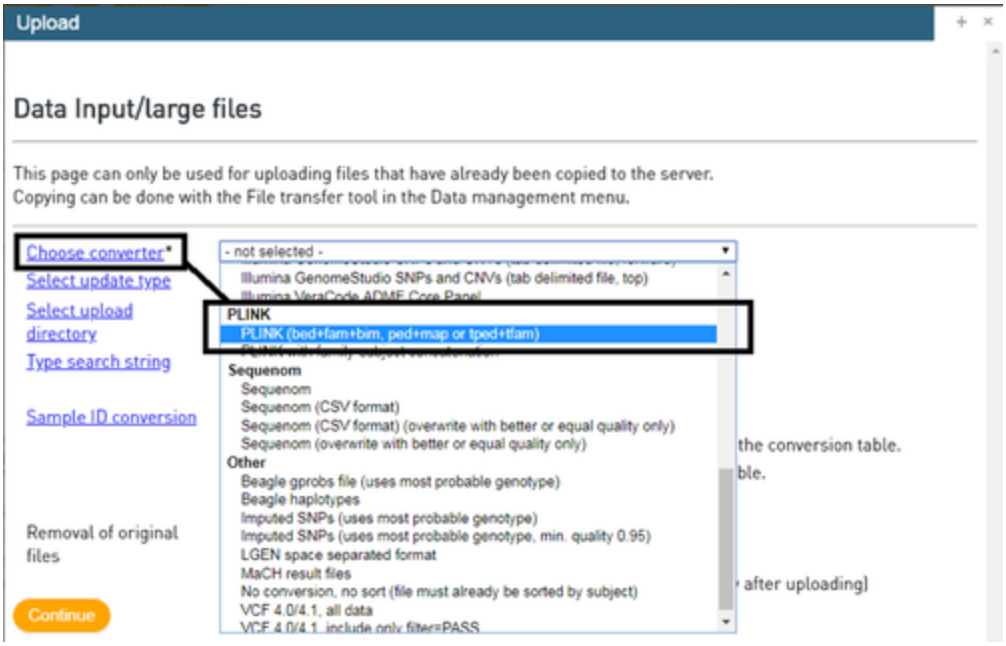


Image 1. Upload options for PLINK -formatted genotype data.

Select the upload folder where you saved your PLINK file set, and if the folder contains other files as well, type a filter text using * wildcard (e.g. myfile*) to include only the wanted upload files. Proceed to fill in the rest of the upload options that are relevant for you, and continue with the upload by clicking "Continue" button.

You can confirm your file selection in the next dialog page. This page also gives you the choice of adding PLINK data management and filtering options as shown in Image 2. The various options available are explained in Table 1. For more information, see the PLINK manual. Go to http://zz z.bwh.harvard.edu/plink/.

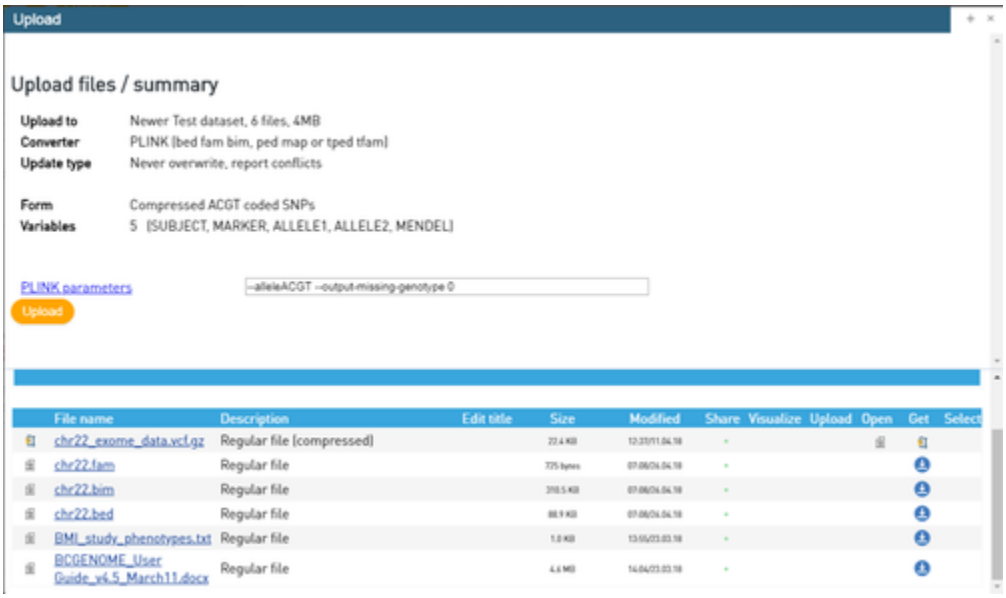


Image 2. You can specify PLINK options for the data upload. See Table 1 for the full list of supported flags.

Flag	Explanation
--no-sex	PED file does not contain column 5 (sex)
--no-parents	PED file does not contain columns 3,4 (parents)
--no-fid	PED file does not contain column 1 (family ID)
--no-pheno	PED file does not contain column 6 (phenotype)

--liability	PED file does contain liability (column 7)
--map3	Specify 3-column MAP file format
--dog	Set chromosome codes for dog
--mouse	Set chromosome codes for mouse
--horse	Set chromosome codes for horse
--cow	Set chromosome codes for cow
--sheep	Set chromosome codes for sheep

Table 2. Available options, flags and filters for the PLINK binary which is used to create the data conversions.

After you are satisfied with your choices you can proceed with "Upload". Your upload work will now go to the queue and you can follow the progress, and read the generated reports after the work is finished.

BC|INSIGHT - 6.4.2 Importing pedigree and affection status as PLINK files

User roles
Data manager

Table of contents:

- Importing pedigree data and affection status data

Importing pedigree data and affection status data

If your PLINK file set (i.e. .ped , .fam or .tfam file) contains pedigree structures (i.e. the paternal ID, maternal ID, sex and phenotype columns contain other values than 0 for all the rows), you can import that data to the BC|INSIGHT and use that for family based analysis or pedigree visualisation. Note that the subject IDs in the pedigree data must be the same as what they are in the genotype data. So, if you concatenated the PLINK family ID and individual ID when importing the genotype data, you must concatenate them again for pedigree data too.

In order to upload the data you need to choose or create a pedigree dataset. To create a new one, go to "New dataset" -tool and choose the form “Pedigrees with affection status”. In your selected or newly created dataset go to *Tools and export Upload Files from local computer*. Select your .ped, .fam or .tfam file from your local desktop, and choose a suitable PLINK converter using these criteria:

1. If you have phenotype column in the PLINK file and you do not need the ID concatenation, choose “PLINK / linkage format with affection status”
2. If you have phenotype column in the PLINK file and you need ID concatenation, choose “PLINK / linkage format and family ID - subject ID concatenation (fid_id) with affection status”,
3. If you do not have phenotype data in the PLINK file and you do not need ID concatenation, use “PLINK / linkage format”

Continue with the upload by clicking "Upload". The file import work will go to the job queue, where you can follow the progress, and read the report of upload events in the results archive.

BC|INSIGHT - 6.4.3 Import marker map from PLINK files

User roles
Data manager

Table of contents:

- Importing marker map data

Importing marker map data

The PLINK .map, .bim or .tped file contains the marker map data. You can store this data in BC|INSIGHT as marker information or annotation. Note that, if the markers are RS coded in your data, it is also possible to use the dbSNP marker maps from BC Platforms' Download service in your BC|INSIGHT.

To import the marker map, you need to choose or create a map dataset that contains marker ID, chromosome and position fields. When creating new datsets, choose either "Physical bp map" or “dbSNP chromosome position” form. In your selected or newly created dataset, go to *Tools and export Upload Files from local computer*. Select the converter “PLINK bp map (.map / .bim / .tped)” and choose your corresponding PLINK file for upload. Continue to click "Upload" and the file import goes to the job queue. You can follow the progress in the queue and see the upload report after the import job is finished in the results archive.