SAS Connector for Biogen Documentation

# Revisions

|  |  |
| --- | --- |
| Revision | Description |
| 19th December 2013 | First complete draft |
| 6th January 2014 | Minor revisions |
| 4th March 2014 | Revised into documentation reflecting finished connector |

Table of Contents

[Revisions 1](#_Toc406053745)

[Overview 4](#_Toc406053746)

[Requirements 4](#_Toc406053747)

[Environment 4](#_Toc406053748)

[SAS Connection to the tranSMART Database 5](#_Toc406053749)

[Connectivity to an Oracle database 5](#_Toc406053750)

[Connectivity through OBDC 5](#_Toc406053751)

[SAS Connector Configuration 5](#_Toc406053752)

[Making the tranSMART macros available 5](#_Toc406053753)

[Database connectivity 5](#_Toc406053754)

[Database parameters 6](#_Toc406053755)

[SAS Connector Macros 6](#_Toc406053756)

[tm\_listStudies 6](#_Toc406053757)

[Arguments 6](#_Toc406053758)

[Output Data Set 7](#_Toc406053759)

[tm\_listSearchTerms 7](#_Toc406053760)

[Arguments 7](#_Toc406053761)

[Output Data Set 7](#_Toc406053762)

[tm\_getDistinctConcepts 7](#_Toc406053763)

[Arguments 7](#_Toc406053764)

[Output Data Set 7](#_Toc406053765)

[tm\_getClinicalData 7](#_Toc406053766)

[Arguments 8](#_Toc406053767)

[Output Data Set 8](#_Toc406053768)

[tm\_getPatientMapping 8](#_Toc406053769)

[Output Data Set 8](#_Toc406053770)

[tm\_getClinicalMutationData 8](#_Toc406053771)

[Arguments 9](#_Toc406053772)

[Output Data Set 9](#_Toc406053773)

[tm\_getGeneGoMembership 9](#_Toc406053774)

[Arguments 9](#_Toc406053775)

[Output Data Set 9](#_Toc406053776)

[tm\_getGEXData 9](#_Toc406053777)

[Arguments 9](#_Toc406053778)

[Output Data Set 10](#_Toc406053779)

[tm\_getProbeGeneMapping 10](#_Toc406053780)

[Arguments 10](#_Toc406053781)

[Output Data Set 10](#_Toc406053782)

[tm\_getProbeGeneSNPMapping 10](#_Toc406053783)

[Arguments 10](#_Toc406053784)

[Output Data Set 10](#_Toc406053785)

[transmart.getSNPData 10](#_Toc406053786)

[Arguments 11](#_Toc406053787)

[Output Data Set 11](#_Toc406053788)

[tm\_listHDDAttributes 11](#_Toc406053789)

[Arguments 11](#_Toc406053790)

[Output Data Set 11](#_Toc406053791)

# Overview

This document describes the design of the SAS Connector developed for Biogen Idec. It details the high-level approach for connecting to the tranSMART database and the macros that are available for interacting with its data.

# Requirements

Biogen requires a programmatic connector from SAS to tranSMART, enabling data to be extracted from the tranSMART database and used in SAS scripts. The design of the connector should be modelled after the existing R connector for tranSMART.

The connector needs to include:

* A method of configuring the connector to run against a specified Oracle database
* A suite of SAS macros practically identical to the functions provided in the existing R connector for querying and returning the tranSMART data.

# Environment

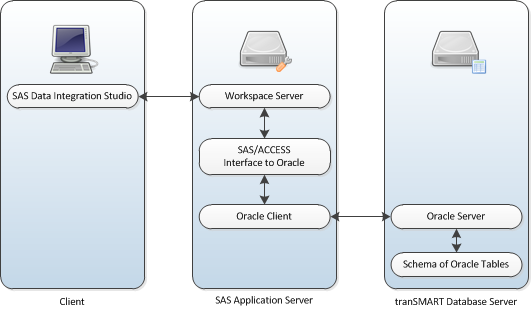
Biogen runs a SAS 9.3 environment on Hewlett-Packard UNIX. The SAS connector is coded to run against this standard.

tranSMART is running on an Oracle 10.2 server, and it is assumed that Biogen’s network configuration allows a connection between the SAS and database servers.

# SAS Connection to the tranSMART Database

SAS provides two options for connecting to Oracle databases. Both are made available through the SAS/ACCESS Interface to Relational Databases family of interfaces, which can be installed on top of an SAS instance.

## Connectivity to an Oracle database



The SAS/ACCESS Interface to Oracle is the most direct way of interacting with an Oracle database. The SAS connector uses the SQL pass-through facility to interact with tranSMART – this allows SQL statements to be constructed and passed to the database directly.

## Connectivity through OBDC

SAS also supports connecting to an Oracle database through an ODBC middleman driver. This interface supports the same methods of accessing data as the Oracle interface does, but the syntax for accessing the database is slightly different.

# SAS Connector Configuration

## Making the tranSMART macros available

To be able to run the tranSMART macros, import the tranSMART library and define it as a macro source:

libname transmrt 'w:\biogen\SAStranSMART\';

options mstored sasmstore=transmrt;

(Replacing “w:\biogen\SAStranSMART\” with the location of the SAS library containing the macros)

## Database connectivity

If using the SAS/ACCESS connector for Oracle, no modifications are necessary for database connectivity.

If using ODBC on Windows, a valid ODBC connection must be set up through the Windows ODBC manager, and the macro for querying the database must be modified as follows:

%macro tm\_query(out=out, sql=, debug=false) / store;

%if debug=true %then %put TM running SQL: &sql;

proc sql;

CONNECT TO ODBC AS TSMART(database=&tmdsn user=&tmuser password=&tmpassword);

CREATE TABLE &out AS

SELECT \* FROM CONNECTION TO TSMART

&sql;

quit;

%mend;

## Database parameters

Three parameters must be set up in a SAS program in order to use the macros:

%let tmdsn=PRTRAN;

%let tmuser=biomart\_user;

%let tmpassword=biomart\_user;

“tmdsn” should be set up as the Oracle path to the data source (contact the Oracle administrator to determine what this should be), or if using ODBC, the TNS name.

tmuser and tmpassword are the username and password to use to connect to tranSMART.

# SAS Connector Macros

The layout of the SAS connector mirrors the layout and function of the R connector as far as it was possible, though the differences between the two environments necessitated a few changes to the detail of some of the functions.

Where a parameter for a macro is the name of a dataset, a single-column dataset is always expected. If the dataset contains multiple columns, the macro will use the first available column.

Transmart macros are prefaced with “tm\_”.

## tm\_listStudies

This macro will list studies that have a character match with the supplied strings.

This macro uses the passed in search string (with % as wildcards) to generate a list of studies and their concept codes. The created data set’s STUDYCODE column can be used to identify studies within the other SAS tranSMART macros. The second parameter is used to trim the concept path to collapse records.

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | studyList | The dataset to which the results of this macro will be written. |
| search | String |  | Text string to search for. |
| conceptSize | Numeric | 4 | The number of levels to keep in the concept path – the number actually reflects the number of "\" delimiters to keep. Default is 4. |
| gex | Boolean | False | If used, this flag will return a count from the microarray table of distinct probe IDs for the given study. This will indicate if GEX data is available in the study. |
| debug | Boolean | False | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

A data set is created with a STUDYCODE and CONCEPT\_PATH column. Both of these come from the table that is used to generate the tree in the Dataset Explorer. The concepts are grouped by the Study Code and the Concept Path. You can pass the STUDYCODE column to the other tranSMART SAS macros as a study filter.

## tm\_listSearchTerms

This macro finds terms in tranSMART’s search index (SEARCH\_KEYWORD tables).

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | searchtermlist | The dataset to which the results of this macro will be written. |
| search | String |  | String to match against the start of the search terms |
| category | String |  | Search term category (leave blank for all) |
| debug | Boolean | False | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

A data set is created with the search terms that begin with the given search string. If a category is given, such as GENE, then the list will be limited to search terms with that category.

## tm\_getDistinctConcepts

This macro finds clinical concepts in the tranSMART database that match the supplied strings.

Before clinical data can be retrieved from the tranSMART warehouse the user must find the concept codes associated with the data to be retrieved. This macro will perform a search of the concept paths within tranSMART, restricted to a list of studies.

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | distinctConcepts | The dataset to which the results of this macro will be written. |
| studyList | Dataset |  | Single-column data set containing the study accessions to search in for concept codes. |
| pathMatchList | Dataset |  | Single-column data set containing the list of paths to match (each term is surrounded by wildcard characters). |
| debug | Boolean | false | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

A data set is created with a column for the concept code (which will be used in the getClinicalData macro), the concept path and a count of the number of patients who have that concept. A column with the study ID is also supplied.

## tm\_getClinicalData

This macro creates a data set containing clinical data from the tranSMART database.

Clinical data (also known as low-dimensional data) represents clinical parameters that have been collected about patients and samples. This could include medical history, demographics, or low complexity assays. In order to retrieve this data the user must supply a list of codes which can be retrieved using the getDistinctConcepts macro.

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | clinicalData | The dataset to which the results of this macro will be written. |
| conceptCodeList | Dataset |  | A single-column data set of concept codes. Can be passed in from the return of the getDistinctConcepts macro. |
| pivot | Boolean | true | A boolean indicating whether the data should be pivoted after retrieving. |
| prePivotTrim | Boolean | true | This boolean indicates if the concept paths should be trimmed before the pivot occurs. A concept path is a "\" separated string indicating a concept’s relation to other concepts. The path is laid out visually in the dataset explorer tree within tranSMART.  By trimming items off before pivoting, columns can be collapsed. An example is when multiple studies worth of data are retrieved but the path is trimmed so that the survival time node from all 4 studies has the same concept code. When the data is pivoted all the survival times will be put into one column. |
| trimLength | Numeric | 4 | This is the number of "\" separated elements to remove from the concept path when trimming, if prePivotTrim is true. A negative trim length will leave items from the end of the path - positive lengths remove from the beginning. |
| debug | Boolean | false | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

A data set is created with a patient num (internal identifier), Subject ID (ID that is taken from the raw input files when data is loaded into tranSMART), Trial Name (Name of the study) and the remaining columns are the clinical data items. If the data is not pivoted, it will be represented in a long format instead of a wide one.

## tm\_getPatientMapping

This macro gets a data set that contains a mapping between tranSMART unique IDs and the Subject IDs supplied in the curation process.

tranSMART assigns unique IDs to patient records in addition to the identifier that is carried over from the data loading process. This macro will get the mapping for a list of studies.

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | patientmapping | The dataset to which the results of this macro will be written. |
| studyList | Dataset |  | A list of studies to get the patient IDs for. Patient IDs are unique across studies where the ID provided in the data loading process may not be. |
| Debug | Boolean | False | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

The macro creates a data set mapping tranSMART unique IDs to Subject IDs for patients in the given studies.

## tm\_getClinicalMutationData

This macro performs a very specific search for a Mutation Type concept within the ontology tree.

In order to pull a list of the mutations that a patient has a search is done on the ontology to find the concept that represents the mutations on the genes specified. A data set is constructed with a column per supplied gene and a row per patient in the studies supplied. The intersection lists the concept (if one exists).

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | clinicalMutationData | The dataset to which the results of this macro will be written. |
| studyList | Dataset |  | A list of studies to limit the text search to. This lookup is case insensitive. |
| geneList | Dataset |  | A list of genes to look for mutations for. |
| trimLength | Numeric | 4 | This is the length of the resulting concept code at a patient/gene intersection. |
| Debug | Boolean | False | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

A data set is created with PATIENT\_ID (Internal Identifier), SUBJECT\_ID (Identifier from source data), STUDY ID (Internal ID) and a column for each gene specified. In each column is the result of searching to see if that patient was associated with a mutation in that gene. If nulls are present, that means the concept was not associated with this patient (i.e. they do not have the mutation).

## tm\_getGeneGoMembership

Retrieve GeneGo pathway information from the tranSMART database.

This macro will take a pathway name and return all the genes in that pathway. Optionally the user can download all the pathway information by leaving the parameter blank.

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | geneGoMembership | The dataset to which the results of this macro will be written. |
| geneGoName | String |  | The name of the GeneGo pathway to look up genes for. |
| Debug | Boolean | False | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

The macro creates a data set listing the gene names for the specified GeneGo pathway.

## tm\_getGEXData

This macro gets Gene Expression data from the tranSMART database based on filters. Options are available to pivot and aggregate the data.

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | gexData | The dataset to which the results of this macro will be written. |
| studyList | Dataset |  | A list of studies to limit the text search to. This lookup is case insensitive. |
| geneList | Dataset |  | A list of genes to filter the GEX by. |
| pathway | Dataset |  | A list of pathways (Exact text match, case insensitive.) |
| signature | Dataset |  | A list of names of gene signatures that were previously loaded into the tranSMART database. |
| patientList | Dataset |  | A list of patient IDs to filter the data by. This is the patient identifier generated within tranSMART. |
| sampleTypeList | Dataset |  | A list of sample types to filter the data by. The available sample types can be retrieved using the transmart.listHDDAttributes macro. |
| tissueTypeList | Dataset |  | A list of tissue types to filter the data by. The available tissue types can be retrieved using the transmart.listHDDAttributes macro. |
| timepointList | Dataset |  | A list of timepoints to filter the data by. The available timepoints can be retrieved using the transmart.listHDDAttributes macro. |
| platformList | Dataset |  | This is a list of platforms to filter on. |
| probeList | Dataset |  | A list of probe ids to filter the results by. |
| removeOnOverlap | Dataset |  | This is a list of platforms that you want to be overlapped if there are probes in multiple platforms. The script will check for probes in multiple platforms and remove the records that have a platform in the provided list. |
| showGenes | Boolean | False | If this is set to true a gene column will be added to data output. |
| pivot | Boolean | True | Flag indicating whether the GEX data should be pivoted or not. |
| pivotAggregate | Macro |  | The name of a macro to use for aggregating and pivoting data (if pivot is true). If this is not provided, a pivot will be performed without aggregation. |
| pivotPatientId | Boolean | false | This will use the tranSMART internal identifier which is guaranteed to be unique per study (barring any data issues). |
| debug | Boolean | False | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

A data set with the filtered GEX data, pivoted depending on the given parameters.

## tm\_getProbeGeneMapping

This macro will get a lookup table of probes/genes from the tranSMART database through the MRNA annotation table.

tranSMART has annotation information stored that can be queried by either the probe or gene ID to get the appropriate mapping information between the two.

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | pgmapping | The dataset to which the results of this macro will be written. |
| probeIds | Dataset |  | A list of probe identifiers to find the gene symbols for. |
| geneIds | Dataset |  | A list of gene symbols to look up the probe ids for. |
| debug | Boolean | False | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

A data set mapping the given probe IDs to gene symbols, or vice versa.

## tm\_getProbeGeneSNPMapping

This macro will get a lookup table of SNP Names/genes from the tranSMART database through the SNP/gene mapping table.

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | pgsnpmapping | The dataset to which the results of this macro will be written. |
| probeIds | Dataset |  | A list of SNP identifiers to find the gene symbols for. |
| geneIds | Dataset |  | A list of gene symbols to look up the SNP names for. |
| debug | Boolean | False | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

|  |  |
| --- | --- |
| Argument | Description |
| probeIds | A list of SNP identifiers to find the gene symbols for. |
| geneIds | A list of gene symbols to look up the SNP names for. |

### Output Data Set

A data set mapping the given probe IDs to gene symbols, or vice versa.

## transmart.getSNPData

This macro will retrieve SNP data from the tranSMART database based on filters. Options are available to pivot and aggregate the data.

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| out | Dataset | snpData | The dataset to which the results of this macro will be written. |
| studyList | Dataset |  | A list of studies to limit the text search to. This lookup is case insensitive. |
| geneList | Dataset |  | A list of genes to filter the SNP data by. |
| pathway | Dataset |  | A list of pathways (Exact text match, case insensitive.) |
| signature | Dataset |  | A list of names of gene signatures that were previously loaded into the tranSMART database. |
| patientList | Dataset |  | A list of patient IDs to filter the data by. This is the patient identifier generated within tranSMART. |
| sampleTypeList | Dataset |  | A list of sample types to filter the data by. The available sample types can be retrieved using the transmart.listHDDAttributes macro. |
| tissueTypeList | Dataset |  | A list of tissue types to filter the data by. The available tissue types can be retrieved using the transmart.listHDDAttributes macro. |
| timepointList | Dataset |  | A list of timepoints to filter the data by. The available timepoints can be retrieved using the transmart.listHDDAttributes macro. |
| platformList | Dataset |  | This is a list of platforms to filter on. |
| probeList | Dataset |  | A list of probe ids to filter the results by. |
| showGenes | Boolean | False | If this is set to true a gene column will be added to data output. |
| pivot | Boolean | True | Flag indicating whether the SNP data should be pivoted or not. |
| pivotAggregateCN | Macro |  | The name of a macro to use for aggregating and pivoting copy number data (if pivot is true). If this is not provided, a pivot will be performed without aggregation. |
| pivotAggregateGT | Macro |  | The name of a macro to use for aggregating and pivoting genotype data (if pivot is true). If this is not provided, a pivot will be performed without aggregation. |
| pivotPatientId | Boolean | False | This will use the tranSMART internal identifier which is guaranteed to be unique per study (barring any data issues). |
| debug | Boolean | False | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

If unpivoted, a single data set is created. If pivoted, two data sets are created, one for Copy Number and one for Genotype.

|  |  |
| --- | --- |
| [data set name]CN | Copy Number Data |
| [data set name]GT | Genotype Data |

## tm\_listHDDAttributes

List high-dimensional data attributes within tranSMART.

For a given list of studies, this macro will query the table that maps patients to samples. The macro creates 3 data sets, one for each of the data types and their distinct values. These values can be used to filter in other macros, like retrieving gene expression data by a sample type.

### Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| Arguments | Type | Default | Description |
| sampleTypes | Dataset | sampleTypes | The dataset to which the sample type results of this macro will be written. |
| tissueTypes | Dataset | tissueTypes | The dataset to which the tissue type results of this macro will be written. |
| timepoints | Dataset | Timepoints | The dataset to which the timepoint results of this macro will be written. |
| studyList | Dataset |  | A list of studies to retrieve the HDD attributes for. |
| debug | Boolean | false | Common to all macros – if set to true, the macro will output the SQL it constructs before running it. |

### Output Data Set

A list of 3 data sets, each frame is a distinct list of a sample attribute type.