*Ontoscope:* Determining “identity-defining” transcription factors for various cell types

Vignette authors: Kartikay Chadha, Shivani Kamdar, Burton Mendonca, Anam Qudrat

University of Toronto

Department of Biochemistry

Compiled: April 27, 2016

**Contents**

**1 Introduction 2**

**2 Processing overview 2**

**3 Submodules and workflow**  **z**

3.1 Normalization z

3.2 FANTOM expression data import z

3.3 Background derivation from cell ontology z

3.4 Protein and transcription factor network analysis z

3.4.1 STRING

3.4.2 TRRUST

3.4.3 Regnet

3.5 Differential gene expression analysis z

3.6 Integration of network- and expression-based influence scores z

3.7 Selection of transcription factors needed for conversion z

**4 Post-processing and analysis z**

4.1 Visualization z

4.2 Literature-based confidence score z

4.3 Validation from published experimental data z

**5 Acknowledgments z**

**6 Session info z**

**7 References z**

**1 Introduction**

This document offers an overview of the R package *Ontoscope*, which uses publicly available expression and protein-protein interaction data to computationally determine a list of transcription factors able to facilitate conversion from one cell type to another. Following determination of this transcription factor list, it is also designed to assign confidence scores to each factor based on literature search and validation against published experimental data and/or other computational prediction software.

The *Ontoscope* package is designed to independently validate the findings of Owen Rackham and colleagues’ *Mogrify* ([www.mogrify.net](http://www.mogrify.net))1. Its workflow is based on the published *Mogrify* protocol, with modifications including the usage of updated transcription factor definitions and the inclusion of different regulatory networks (specifically, the TRRUST2 and Regnet3 protein-protein interaction datasets).

This vignette contains an overview of the package workflow, defining initialization parameters and walking through each submodule with examples and usage instructions provided.

**2 Processing overview**

*Ontoscope* determines transcription factors required for conversion based on calculation of network- and expression-based influence scores for differentially expressed factors in the target cell line as compared to the source. Target and source cells are identified by FANTOM consortium cell line IDs, and are restricted to those cell types for which FANTOM gene expression data from CAGE-seq is available.

Calculation involves six primary phases:

1. *Import of expression data:* The first step is to define source and target cell lines, reading in raw FANTOM expression count data for each cell line. Transcription factors are defined from published literature and databases, and all gene names are normalized to HGNC IDs in order to ensure accuracy of comparison between cell lines and datasets. Download of FANTOM count data is automatic upon input of cell line IDs, and does not require further input from the user.
2. *Calculation of background:* Something
3. *Differential expression-based influence score assignment:* Something
4. *Interaction-based influence score assignment:* Something
5. *Integration and binning:* Something
6. *Visualization and validation: Visualiztion goes here.* The characteristic set of transcription factors predicted by Ontoscope are validated against three sources of published transcription factor lists: published cell conversions [1], Mogrify predictions [1], and D’Alessio *et al.* predictions [6].

**3 Submodules and workflow**

This section outlines examples and functions underlying the usage of *Ontoscope* to generate lists of transcription factors for conversion.

Users should begin by defining cells of origin and desired target cell type as follows:

> sourcecell <- “eye”

> target <- “fibroblast”

Here, a retinal to fibroblast cell conversion has been used as an example. It is important to define both keywords and FANTOM IDs for desired cell types, as while FANTOM IDs will be used for the actual Ontoscope workflow, keyword definitions are important for visualization and literature validation.

Next, users should select their desired FANTOM IDs for source and target cells by sourcing and running the **fantom\_import** submodule for both source and target cell types. Output for the source cell search is as follows:

> source(“./fantom\_import/fantom\_main.R”)

> fantomSearch(sourcecell)

V1 FANTOM.5.Ontology.ID FANTOM.5.Access.Number

1375 eye - muscle inferior rectus, donor1 FF:10272-104E2 1381

1376 eye - muscle lateral, donor2 FF:10298-104H1 1382

1377 eye - muscle medial, donor2 FF:10299-104H2 1383

1378 eye - muscle superior, donor2 FF:10297-104G9 1384

1379 eye - vitreous humor, donor1 FF:10268-104D7 1385

1380 eye, fetal, donor1 FF:10054-101G9 1386

From the list, the desired specific subtype (ie: lateral eye muscle, vitreous humor, fetal eye) of cell can be selected, and the FANTOM Ontology ID (FF:ID) noted. Here, we have chosen to convert inferior rectus eye muscle to cardiac fibroblast. FF:IDs may then be defined as follows:

> sourceFF <- “FF:10272-104E2”

> targetFF <- “FF:11268-116G8”

Following this, simply running runOntoscope.R will process the entire workflow for you, outputting a list of top transcription factors required for conversion with confidence rankings based on literature as well as heatmap and interaction network visualizations. The runOntoscope.R package also contains built-in functionality to cross-validate the transcription factor lists generated by Ontoscope against published conversion lists from either literature or the MOGRIFY package.

**3.1 Normalization**

Prior to beginning the workflow proper, it is important to first normalize all gene identifiers in order to allow comparison between gene lists and data from different sources. Ontoscope takes as input four main sources of data: gene expression data from FANTOM5, and protein-protein interaction and regulatory network data from STRING4, TRRUST, and REGNET. While FANTOM5 and TRRUST datasets contain HGNC identifiers for genes, STRING primarily identifies genes based on Ensembl IDs, and must be normalized to HGNC symbols for comparison purposes. Furthermore, while REGNET uses HGNC identifiers for its gene interactions, these IDs are four years old at the time of publication of this vignette, and may be outdated.

In order to normalize STRING interactions, the **normalizeWeave.R** submodule is sourced, using the R Bioconductor package biomaRt to create a new datafile from the base STRING data table with approximately 8.5 million protein-protein interactions. In order to avoid extremely lengthy processing times, STRING Ensembl protein IDs are placed in a new data frame as row names, following which a new vector of HGNC symbols is created with biomaRt. The updated file, **curatedOutput.Rdata**, is then available for usage in downstream applications.

Should Regnet files require updated HGNC symbol assignments, Entrez IDs can be retrieved from Regnet files and transcribed into HGNC symbols as well by altering the following lines in **normalizeRegnet.R** :

> IDmap <- data.frame(entrezgene=IDmap, HGNC=””, stringsAsFactors=FALSE)

> BMmap <- getBM(filters = “entrezgene”,

attributes = c(“entrezgene”, output),

values = IDmap$entrezgene,

mart = ensembl)

> colnames(BMmap) <- c(“entrezgene”, output)

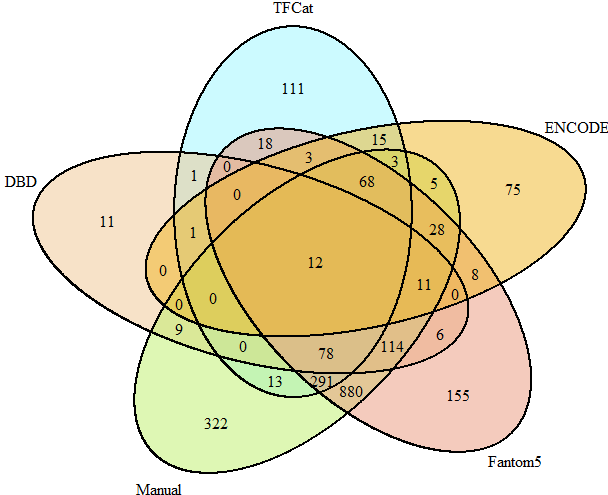
All input datafiles will now have been normalized to identify genes by HGNC symbols.

Finally, transcription factors must be defined prior to performing analysis. Sourcing the **normalizeTF.R** subscript will automatically compile a list of transcription factors common among at least two out of the five following datasets:

1. The TFCat database
2. The DBD (DNA-binding domain) database
3. A manually assembled list of transcription factors with annotations5
4. The FANTOM5 transcription factor dataset
5. ENCODE transcription factor data derived from their ChIP-seq antibody list

The choice of how many lists to use as a cutoff was based on variations in overlap between the lists **(Figure 1)**, and may be easily modified through subsetting of the MergedList table generated by this submodule, which lists all transcription factors and whether or not they are present in each dataset, via modification of the following line:

> TFList <- unique(MergedList[!MergedList$Count==”1”,])$Gene.Symbol



**Figure 1: Varying degrees of overlap between human transcription factor datasets.** Venn diagram depicting large overlap between the manually curated and Fantom5 lists, while other datasets possess less unique and overlapping transcription factors. Generated by the VennDiagrams package: draw.quintuple.venn()

Note that all transcription factors have been normalized to HGNC symbols where appropriate. The normalize submodule allows for easy replacement or update of lists through the replacement of their .csv datafiles with updated data.

**3.2 FANTOM expression data import**

Expression profiles for genes across available cell lines have been derived from cap analysis of gene expression (CAGE) sequencing by the FANTOM5 consortium. The FANTOM import submodule allows for the import and processing of raw read counts from this sequencing data, accepting several different possible search terms as inputs and allowing for import of normalized read counts as an alternative. For the purposes of the Ontoscope workflow, the **fantom\_main.R** submodule has already been sourced. Furthermore, as we perform our own normalization and background derivations further downstream, we must focus on importing raw read counts rather than normalized expression data.

The **runOntoscope.R** script first searches the FANTOM5 database for matching FF IDs and downloads the raw expression counts for each as follows:

> FFVect <- c(sourceFF, targetFF)

> fantomOntology(FFVect)

[1] "Sample\_DB Loaded!"

Returning RAW COUNTS

MATCHED: 2 of 2

2 Search Result(s) Were Found. Loading...

Loading Results from Fantom Access Number 1381 ( 1 / 2 ) ...

Results from Fantom Access Number 1381 Loaded!

Loading Results from Fantom Access Number 377 ( 2 / 2 ) ...

Results from Fantom Access Number 377 Loaded!

All results have been loaded into fantomResults

fantomResults is a large, subsettable list of dataframes containing genetic annotations, peak numbers, gene names, and HGNC, Uniprot, and Entrez Gene IDs for every sample, with one dataframe per sample. These results can be summarized in one dataframe, with normalized HGNC symbol gene names and raw expression counts for each gene, through the fantomSummarize command:

> fantomSummarize(5)

Preparing the Genes

Summarizing:eye - muscle inferior rectus, donor1.CNhs13444.10272-104E2

Summarizing:Fibroblast - Cardiac, donor1.CNhs12498.11268-116G8

Filtering Relevant Results. This step takes awhile ...

Preparing Normalized Gene Names ...

All Genes Normalized!

Fixing Duplicates ...

Applying Threshold ...

Your results have been summarized in: fantomCounts!

The bracketed number indicates the minimum threshold for read counts in each gene. Genes with read counts less than the threshold number (here, 5) will not be included in the summarized fantomCounts

file, which is now ready for downstream processing.

**3.3 Background derivation from cell ontology**

**3.4 Protein and transcription factor network analysis**

3.4.1 STRING

3.4.2 TRRUST

3.4.3 Regnet

**3.5 Differential gene expression analysis**

**3.6 Integration of network- and expression-based influence scores**

**3.7 Selection of transcription factors needed for conversion**

**4 Post-processing and analysis**

**4.1 Visualization**

**4.2 Literature-based confidence score**

**4.3 Validation from published data**

The characteristic set of transcription factors predicted by Ontoscope can be compared to published sets of transcription factors for cell conversions. The **OntoscopeValidate.R** script contains the functions to compare Ontoscope-predicted transcription factors with published transcription factors for a specific cell conversion.

One of the inputs required to run the script is a **list** of characteristic transcription factors for a cell conversion. This can be obtained by subsetting the first column in the dataframe produced by the pruneTFDataFrame function in the **OntoscopeRank.R** script (bolded below).

gene finalRank gsxRank tisSTRINGRank tisMARARank

1 **gene 14** 8 2 1 5

4 **gene 4** 11 1 5 5

10 **gene 3** 15 5 5 5

Another important input for the script is cell conversion transcription factor lists obtained from published datasets. These are the lists used for validation of Ontoscope predictions. As part of initial build of Ontoscope, published cell conversion transcription factor dataframes have been produced from the following sources:

1. Published conversion transcription factors, referenced by Rackham *et al.* [1]
2. Cell conversion transcription factors predicted by D’Allessio *et al.* [6]
3. Cell conversion transcription factors predicted by Mogrify (Rackham *et al.*) [1]
4. Cell conversion transcription factors predicted by Mogrify (Rackham *et al.*) [1] using only MARA network influence scores
5. Cell conversion transcription factors predicted by Mogrify (Rackham *et al.*) [1] using only STRING network influence scores

These five dataframes have the following general format, with the CellFrom and CellTo columns identifying the cell conversion, and the TFList column containing the transcription factors for the conversion.

CellFrom CellTo TFList

1 Bcell Macrophage CEBPA

2 Bcell Macrophage SPI1

3 Fibroblast Macrophage CEBPA

4 Fibroblast Macrophage SPI1

5 Fibroblast Myoblast MYOD1

6 Fibroblast IPS SOX2

7 Fibroblast IPS OCT4

8 Fibroblast IPS KLF4

9 Fibroblast IPS CMYC

The getTFList function is used to retrieve the transcription factor list for a specific cell conversion based on the . For example, to get the transcription factor list for the cell conversion from “Bcell” to “Macrophage” in the example dataframe above, we must enter the following command:

myList <- getTFList(dataSet, "Bcell", "Macrophage")

Once transcription factor lists are generated, they can be compared with Ontoscope-predicted lists using three different functions in the **OntoscopeRank.R** script: fractionRecovery, averageTFRank and averageOverlap.

The fractionRecovery function calculated the fraction of of transcription factors in a list that are matched, or recovered, in another “reference” list. For example, the published transcription factors for a Bcell-to-macrophage conversion are SPI1, CEBPA (Xie et al, 2004). The Mogrify predicted TFs are MITF, SPI1, CEBPA, MAFB, DBP, ETS2, SNAI3, HMGA1. The fractional recovery of the published TFs by Mogrify for this conversion is 1, or 100%.

The function is called as follows:

fracRec <- fractionRecovery(myList, refList)

The output is a number between 0 and 1. An output of 0 means no identity between the two lists and an output of 1 means that all the items present in myList are present in refList.

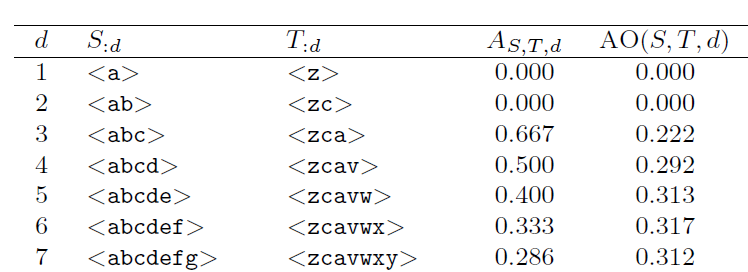
The averageTFRank function calculates the average rank of transcription factors in a “reference” list that have been matched in another list. For example, the published transcription factors for a B cell-to-macrophage conversion are SPI1, CEBPA (Xie et al, 2004). The Mogrify predicted TFs are MITF, SPI1, CEBPA, MAFB, DBP, ETS2, SNAI3, HMGA1. The average rank of recovered TFs is (2 + 3) / 2 = 2.5.The function is called as follows:

avRank <- averageTFRank(myList, refList)

The output is a number between 0 and the number of transcription factors in the reference list. If there is no overlap between myList and the refList, the output is 0.

The averageOverlap function calculates a similarity score between two transcription factor lists, based on the “average overlap” method, outlined by Webber *et al* [7]*.* An example of the average overlap method is illustrated in the figure below for two lists S (<abcdefg>) and T (<zcavwxy>). The fractional overlap between the lists is calculated at increasing depth, d. Fractional overlaps are averaged from depth 1 to d. This average overlap gives a score measuring the similarity of the two lists.

Note that the order of the lists, not only their similarity of elements affects the average overlap score. Higher ranked items in the lists are included repeatedly in the calculation. This means that similarity between higher ranked transcription factors is given more importance in the similarity score. In other words, the similarity score is considered “top-weighted”.



The averageOverlap function is called as shown below. The myList and refList arguments have to be of the same length. If necessary, the longer list must be truncated to match the length of the other list.

avOverlap <- averageOverlap(myList, refList)

The output of the function is a number between 0 and 1, for completely non-identical and identical lists respectively.

**5 Acknowledgments**

This package was developed as part of the BCB420/JTB2020 bioinformatics class as part of the Department of Biochemistry at the University of Toronto, Canada, with the support of Dr. Boris Steipe. We wish to acknowledge all co-authors of *Ontoscope* package modules, who have equally contributed to the development of this project: Julian Mazzitelli, Dmitry Horodetsky, Christopher Li, Kartikay Chadha, Fupan Yao, Shivani Kamdar, Dan Litovitz, Howard Wu, Anam Qudrat, Jialun Tom Chen, Ryoga Li, Eugenia Barkova, Burton Mendonca, Phil Fradkin, and Boris Steipe.

**6 Session Info**

> sessionInfo(package=NULL)

(The output of the above command needs to be filled in by Dr. Steipe after integrating all the code and running it).

**7 References**

[1] Rackham OJ, Firas J, Fang H, Oates ME, Holmes ML, Knaupp AS, FANTOM Consortium, Suzuki H, Nefzeger CM, Daub CO, Shin JW, Petretto E, Forrest AR, Hayashizaki Y, Polo JM, Gough J. A predictive computational framework for direct reprogramming between human cell types. *Nat Genet,* 48(3): 331-335, 2016.

[2] Han H, Shim H, Shin D, Shim JE, Ko Y, Shin J, Kim H, Cho A, Kim E, Lee T, Kim H, Kim K, Yang S, Bae D, Yun A, Kim S, Kim CY, Cho HJ, Kang B, Shin S, Lee I. TRRUST: a reference database of human transcriptional regulatory interactions. *Sci Rep,* 5:11432, 2015.

[3] Chi SM, Seo YK, Park YK, Yoon S, Park CY, Kim YS, Kim SY, Nam D. REGNET: mining context-specific human transcription networks using composite genomic information. *BMC Genomics,* 15:450, 2014.

[4] Szklarczyk D, Franceschini A, Wyder S, Forslund K, Heller D, Huerta-Cepas J, Simonovic M, Roth A, Santos A, Tsafou KP, Kuhn M, Bork P, Jensen LJ, von Mering C. STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res*, 43:D447-52, 2015.

[5] Vaquerizas JM, Kummerfield SK, Teichmann SA, Luscombe NM. A census of human transcription factors: function, expression and evolution. *Nat Rev Genet*, 10(4):252-263, 2009.

[6] D'Alessio, A.C. et al. A systematic approach to identify candidate transcription factors that control cell identity. Stem Cell Reports doi:doi:10.1016/j.stemcr.2015.09.016 (23 October 2015).

[7] Webber, W., Moffat, A., & Zobel, J. (2010). A similarity measure for indefinite rankings. ACM Transactions on Information Systems TOIS ACM Trans. Inf. Syst., 28(4), 1-38. doi:10.1145/1852102.1852106