HPAanalyze: An R Package that Facilitates the Retrieval and Analysis of The Human Protein Atlas Data

Anh Nhat Tran[1*], Anita B. Hjelmeland[1]

[1]Department of Cell, Developmental and Integrative Biology, University of Alabama at Birmingham, USA.

[*]Corresponding author.

ABSTRACT

Background: The Human Protein Atlas aims to map human proteins via multiple technologies including imaging, proteomics and transcriptomics. Access of the HPA data is mainly via web-based interface allowing view of individual proteins, which may not be optimal for data analysis of a gene set, or automatic retrieval of original images.

Results: HPAanalyze is an R package for retrieving and performing exploratory data analysis from HPA. It provides functionality for importing data tables and xml files from HPA, exporting and visualizing data, as well as download all staining images of interest. The package is free, open source, and available via GitHub.

Conclusions: HPAanalyze integrates into the R workflow via the tidyverse philosophy and data structures, and can be used in combination with Bioconductor packages for easy analysis of HPA data.

Keywords: Human Protein Atlas, Proteomics, Homo Sapiens, Visualization, Software

INTRODUCTION

The Human Protein Atlas (HPA) is a comprehensive resource for exploration of the human proteome which contains a vast amount of proteomics and transcriptomics data generated from antibody-based tissue micro-array profiling and RNA deep-sequencing (*1-7*).

The program has generated protein expression profiles in human non-malignant tissues, cancers, and cell lines with cell type-specific expression patterns via an innovative immunohistochemistry-based approach. These profiles are accompanied by a large collection of high quality histological staining images, annotated with clinical data and quantification. The database also includes classification of protein into both functional classes (such as transcription factors or kinases) and project-related classes (such as candidate genes for cancer). Starting from version 4.0, the HPA includes subcellular location profiles generated based on confocal images of immunofluorescent stained cells. Together, these data provide a detailed picture of protein expression in human cells and tissues, facilitating tissue-based diagnostic and research.

Data from the HPA are freely available via proteinatlas.org, allowing scientists to access and incorporate the data into their research. Previously, the R package hpar has been created for fast and easy programmatic access of HPA data (*8*). Here, we introduce HPAanalyze, an R package that aims to simplify exploratory data analysis from those data, as well as provide other complementary functionality to *hpar*.

IMPLEMENTATION

HPAanalyze is an R software package with a permissive license (MIT license) and is designed for easy retrievial and exploratory analysis of data from HPA. It provides functionality

for importing data tables and xml files from HPA, exporting and visualizing data, as well as downloading all staining images of interest.

The Different HPA Data Formats

The HPA project provides data via two main mechanisms: Full datasets in the form of downloadable compressed tab-separated files (.tsv) and individual entries in XML, RDF and TSV formats. The full downloadable datasets include normal tissue, pathology (cancer), subcellular location, RNA gene and RNA isoform data. For individual entries, the XML format is the most comprehensive: it provides information on the target protein, antibodies, summary for each tissue, and detailed data from each sample including clinical data, immunohistochemistry (IHC) scoring and image download links.

HPAanalyze Overview

HPAanalyze is designed to fulfill 3 main tasks: (i) import, subsetting and export of downloadable datasets; (ii) visualization of downloadable datasets for exploratory analysis; and (iii) working with the individual XML files (Figure 1). This package aims to serve researchers with little programming experience, while also allowing power users to utilize the imported data as desired.

Obtaining HPAanalyze

The development version of HPAanalyze is available on GitHub can be installed with the following code:

```
devtools::install_github("trannhatanh89/HPAanalyze")
```

Full Dataset Import, Subsetting and Export

The hpaDownload() function downloads full datasets from HPA and imports them into R as a list of tibbles, the standard object of tidyverse, which can subsequently be subset with hpaSubset() and exported into .xmlx files with hpaExport(). The standard object allows the

imported data to be further processed in a traditional R workflow. The ability to quickly subset and export data gives researchers the option to use other non-R downstream tools, such as GraphPad for creating publication-quality graphics, or share a subset of data containing only proteins of interest.

Visualization

The hpaVis function family takes the output of hpaDownload() (or hpaSubset()) and provides quick visualization of the data, with the intention of aiding exploratory analysis. Nevertheless, the standard ggplot object output of these functions give users the option to further customize the plots for publication. All hpaVis functions share the same syntax for arguments: subsetting, specifying colors, and opting to use custom themes.

The first release of the HPAanalyze package includes three functions: hpaVisTissue() for the normal tissue, hpaVisPatho() for the pathology/cancer, and hpaVisSubcell() for the subcellular location datasets.

Individual XML Import and Image Downloading

The hpaXml function family imports and extracts data from individual XML entries from HPA. The hpaXmlGet() function downloads and imports data as a “xml_document”/“xml_node” object, which can subsequently be processed by other hpaXml functions. The XML format from HPA contains a wealth of information that may not be covered by this package. However, users can extract any data of interest from the imported XML file using the xml2 package.

In the first release, HPAanalyze includes four functions for data extraction from HPA XML files: hpaXmlProtClass() for protein class information, hpaTissueExprSum() for summary of protein expression in tissue, hpaXmlAntibody() for a list of antibodies used to stain for the protein of interest, and hpaTissueExpr() for complete and detailed data from each sample including clinical data and IHC scoring.

hpaTissueExprSum() and hpaTissueExpr() provide download links to obtain relevant staining images, with the former function also having the option to automate the downloading process

Compatibility with hpar Bioconductor Package

HPAanalyze was designed to be compatible and complementary to other existing software packages. Table 1 shows the different functionality between HPAanalyze and hpar, a Bioconductor package optimized for fast acquisition of HPA subsets of data.

Table 1: Complementary Functionality Between hpar and HPAanalyze.

Functionality	**Hpar**	**HPAanalyze**
Datasets	Included in package	Download from server or Import from hpar
Query	Ensembl id	HGNC symbol for datasets, Ensembl id for XML
Data version	One stable version	Latest by default, option to download older
Release info	Access via functions	Not applicable
View relevant browser page	Via getHPA function	Not applicable
Visualization	Not applicable	Exploratory via hpaVis functions
XML	Not applicable	Download and import via hpaXml functions
Histology image	View by loading browser page	Extract links via hpaXml functions

TYPICAL WORKFLOWS AND SAMPLE CODES

The HPAanalyze package can be loaded with the following code:

```
library(HPAanalyze)
```

Working with HPA Downloadable Datasets

A typical workflow with HPA downloadable datasets using HPAanalyze consists of the following steps: (1) Download and import data into R with hpaDownload(); (2) View available parameters for subsetting with hpaListParam(); (3) Subset data with hpaSubset(); and (4) (optional) Export data with hpaExport (**Error! Reference source not found.**Figure 1*)*.

The following code can be used to download the histology datasets (normal tissue, pathology, and subcellular location).

```
data <- hpaDownload(downloadList = 'histology',
          version = 'example')
```

The output of the code showing that data can be subset by normal tissue types, normal cell types, cancer types and subcellular location. The normal_tissue dataset contains information about protein expression profiles in human tissues based on IHC staining. The datasets contain six columns: ensembl (Ensembl gene identifier); gene (HGNC symbol), tissue (tissue name); cell_type (annotated cell type); level (expression value); and reliability (the gene reliability of the expression value).

The pathology dataset contains information about protein expression profiles in human tumor tissue based on IHC staining. The datasets contain eleven columns: ensembl (Ensembl gene identifier); gene (HGNC symbol); cancer (cancer type); high, medium, low, not_detected (number of patients annotated for different staining levels); prognostic_favorable, unprognostic_favorable, prognostic_unfavorable, and unprognostic_unfavorable (log-rank p values for patient survival and mRNA correlation).

The subcellular_location dataset contains information about subcellular localization of proteins based on IF staining of normal cells. The datasets contain eleven columns: ensembl

(Ensembl gene identifier); gene (HGNC symbol); reliability (gene reliability score); enhanced (enhanced locations); supported (supported locations); approved (approved locations); uncertain (uncertain locations); single_cell_var_intensity (locations with single-cell variation in intensity); single_cell_var_spatial (locations with spatial single-cell variation); cell_cycle_dependency (locations with observed cell cycle dependency); and go_id (Gene Ontology Cellular Component term identifier).

hpaListParam() function prints a list of available parameters that can be used to subset the downloaded datasets. Below are the first three items in each group:

```
hpaListParam(data)

#> $normal_tissue
#> [1] "adrenal gland" "appendix"   "bone marrow"
#>
#> $normal_cell
#> [1] "glandular cells"  "lymphoid tissue"  "hematopoietic cells"
#>
#> $cancer
#> [1] "breast cancer" "carcinoid"   "cervical cancer"
#>
#> $subcellular_location
#> [1] "Cytosol"   "Mitochondria" "Aggresome"
```

Based on the information, we can subsequently subset the downloaded data sets based on our genes, tissues, cells and subcellular locations of interest. The following code filters the datasets for MKI67 (Ki67), breast tissue and breast cancer.

```
hpaSubset(data,
     targetGene = 'MKI67',
     targetTissue = 'breast',
     targetCancer = 'breast cancer') -> ki67_data
```

The results (below) showed that Ki67 is expressed at non-detectable-to-medium levels in normal breast tissue, but medium-to-high in breast cancer. It was also indicated, with high reliability, that Ki67 is expressed at high levels in nuclear bodies, nucleoli and nucleus. (For the

sake of clarity, non-relevant columns were omitted.) These data are congruent with the expected pattern of Ki67 expression based on the literature and validate our approach.

```
#> $normal_tissue
#> ensembl    gene tissue cell_type     level    reliability
#> <chr>     <chr> <chr> <chr>       <chr>    <chr>
#> 1 ENSG00000148773 MKI67 breast adipocytes     Not detected Enhanced
#> 2 ENSG00000148773 MKI67 breast glandular cells  Medium   Enhanced
#> 3 ENSG00000148773 MKI67 breast myoepithelial cells Not detected Enhanced
#>
#> $pathology
#> ensembl    gene cancer    high medium low
#> <chr>     <chr> <chr>    <int> <int> <int>
#> 1 ENSG00000148773 MKI67 breast cancer  2  10  0
#>
#> $subcellular_location
#> ensembl    gene reliability enhanced
#> <chr>     <chr> <chr>   <chr>
#> 1 ENSG00000148773 MKI67 Enhanced  Nuclear bodies;Nucleoli;Nucleus
```

Moreover, to facilitate the downstream analysis of data using a non-R software, as well as storage of data subset for reproducible research, the HPAanalyze package included the hpaExport() function, which exports data into Excel file format, with each sheet for a dataset. The code to export the above Ki67 data is as below, which generate an .xlsx file called ‘ki67.xlsx’.

```
hpaExport(data = ki67_data,
     fileName = 'ki67.xlsx')
```

Visualization with The hpaVis Function Family

HPAanalyze provides the ability to quickly visualize data from downloaded HPA datasets with the hpaVis function family (Figure 1). The goal of these functions is to aid exploratory analysis of a group of target genes, which maybe particularly useful for gaining insights into pathways or gene signatures of interest.

The hpaVis functions share a common syntax, where the input (data argument) is the output of hpaDownload() or hpaSubset() (although they do their own subsetting so it is not necessary to use hpaSubset() unless reduction of the size of the data object is desired).

Depending on the function, the target arguments allows the user to choose to visualize vectors of genes, tissue, cell types, etc. (See the help files for more details.) All of hpaVis functions generate standard ggplot2 plots, which allow further customization of colors and themes. Colors maybe changed via the color argument, while the default theme maybe overridden by setting the customTheme argument to FALSE.

Currently, the normal_tissue, pathology and subcellular_location data can be visualized, with more functions planned for future releases.

Visualize Tissue Data with hpaVisTissue()

hpaVisTissue() generates a “heatmap”, in which the expression of proteins of interest (quantified IHC staining) is plotted for each cell type of each tissue. The following code generates a heatmap of expression of various cancer associated genes in certain normal tissues (Figure 2A).

```
geneList <- c('TP53', 'EGFR', 'CD44', 'PTEN', 'IDH1', 'IDH2', 'CYCS')
tissueList <- c('breast', 'cerebellum', 'skin 1')

hpaVisTissue(data,
      targetGene=geneList,
      targetTissue=tissueList)
```

Visualize Expression in Cancer with hpaVisPatho()

hpaVisPatho() generates an array of column graphs showing the expression of proteins of interest in each cancer. This example also demonstrates how the colors of the graphs could be customized, which is a common functionality of the hpaVis family Figure 2B).

```
geneList <- c('TP53', 'EGFR', 'CD44', 'PTEN', 'IDH1', 'IDH2', 'CYCS')
cancerList <- c('breast cancer', 'glioma', 'lymphoma', 'prostate cancer')
colorGray <- c('slategray1', 'slategray2', 'slategray3', 'slategray4')

hpaVisPatho(downloadedData,
      targetGene=geneList,
      targetCancer=cancerList,
      color=colorGray)
```

Visualize Subcellular Location Data with hpaVisSubcell().

hpaVisSubcell() generates a tile chart showing the subcellular location (approved and supported) of proteins of interest. This example also demonstrates the customization of the output plot with ggplot2 functions, which is applicable to all hpaVis functions. Notice that the customTheme argument is set to TRUE (Figure 2C).

```
geneList <- c('TP53', 'EGFR', 'CD44', 'PTEN', 'IDH1', 'IDH2', 'CYCS')

hpaVisSubcell(downloadedData,
       targetGene=geneList,
       customTheme=TRUE) +
  ggplot2::theme_minimal() +
  ggplot2::ylab('Subcellular locations') +
  ggplot2::xlab('Protein') +
  ggplot2::theme(axis.text.x=element_text(angle=45, hjust=1)) +
  ggplot2::theme(legend.position="none") +
  ggplot2::coord_equal()
```

Working with Individual XML Files for Each Target Protein

hpaXml function family supports importing and extracting data from individual XML files provided by HPA for each protein.

A typical workflow for use of XML files includes the following steps: (1) Download and import XML file with hpaXmlGet(); (2) Extract the desired information with other hpaXml functions; and (3) Download histological staining pictures, which is currently supported by the hpaXmlTissurExpr() and hpaXmlTissueExprSum() functions (Figure **1**).

*Import XML File*The hpaXmlGet() function takes a Ensembl gene id (start with *ENSG*) and import the perspective XML file into R. This function calls the xml2::read_xml() under the hood, hence the resulting object may be processed further with *xml2* functions if desired.

```
CCNB1xml <- hpaXmlGet('ENSG00000134057')
```

View Protein Classes

The protein class of a queried protein can be extracted from the imported XML with hpaXmlProtClass(). The output of this function is a tibble of 4 columns: id, name, parent_id and source.

```
hpaXmlProtClass(CCNB1xml)

#> # A tibble: 10 x 4
#>  id  name                 parent_id source
#>  <chr> <chr>                 <chr>  <chr>
#> 1 Ja  Transporters             <NA>   TCDB
#> 2 Jb  Transporter channels and pores    Ja    TCDB
#> 3 Mi  SPOCTOPUS predicted membrane proteins <NA>   SPOCTOPUS
#> 4 Za  Predicted intracellular proteins   <NA>   HPA
#> 5 Pp  Plasma proteins            <NA>   Plasma Proteome D~
#> 6 Ca  Cancer-related genes         <NA>   <NA>
#> 7 Cb  Candidate cancer biomarkers      Ca    Plasma Proteome I~
#> 8 Ua  UniProt - Evidence at protein level  <NA>   UniProt
#> 9 Ea  Protein evidence (Kim et al 2014)   <NA>   Kim et al 2014
#> 10 Eb  Protein evidence (Ezkurdia et al 2014) <NA>   Ezkurdia et al 20~
```

Get Summary and Images of Tissue Expression

The function hpaXmlTissueExprSum() extracts the summary of expression of a protein of interest in normal tissue. The output of this function is a (1) a string containing a one-sentence summary and (2) a dataframe of all tissues in which the protein was positively stained and histological stain images of those tissues.

```
hpaXmlTissueExprSum(CCNB1xml)

#> $summary
#> [1] "Cytoplasmic expression in proliferating cells."
#>
#> $img
#>      tissue
#> 1 cerebral cortex
#> 2   lymph node
#> 3     liver
#> 4     colon
#> 5     kidney
#> 6     testis
#> 7    placenta
#>                              imageUrl
#> 1 http://v18.proteinatlas.org/images/3804/10580_B_8_5_rna_selected.jpg
#> 2 http://v18.proteinatlas.org/images/3804/10580_A_8_8_rna_selected.jpg
#> 3 http://v18.proteinatlas.org/images/3804/10580_A_7_4_rna_selected.jpg
#> 4 http://v18.proteinatlas.org/images/3804/10580_A_9_3_rna_selected.jpg
#> 5 http://v18.proteinatlas.org/images/3804/10580_A_9_5_rna_selected.jpg
#> 6 http://v18.proteinatlas.org/images/3804/10580_A_4_6_rna_selected.jpg
```

```
#> 7 http://v18.proteinatlas.org/images/3804/10580_A_2_7_rna_selected.jpg
```

Those images can be downloaded automatically by setting the downloadImg argument to TRUE.

```
hpaXmlTissueExprSum(CCNB1xml, downloadImg=TRUE)
```

Get Details of Individual IHC Samples

More importantly, the XML files are the only format of HPA programmatically accessible data that contains information about each antibody and each tissue sample used in the project.

hpaXmlAntibody() extracts the antibody information and returns a tibble with one row for each antibody.

```
hpaXmlAntibody(CCNB1xml)

#> id    releaseDate releaseVersion RRID
#> <chr>  <chr>   <chr>     <chr>
#> 1 CAB000115 2006-03-13 1.2      <NA>
#> 2 CAB003804 2006-10-30 2       AB_562272
#> 3 HPA030741 2013-12-05 12      AB_2673586
#> 4 HPA061448 2016-12-04 16      AB_2684522
```

hpaXmlTissueExpr() extracts information about all samples for each antibody above and returns a list of tibbles. If an antibody has not been used for IHC staining, the returned tibble will be empty.

```
tissueExpression <- hpaXmlTissueExpr(CCNB1xml)
summary(tissueExpression)

#>   Length Class Mode
#> [1,] 18  tbl_df list
#> [2,] 18  tbl_df list
#> [3,] 0  tbl_df list
#> [4,] 16  tbl_df list
```

Each tibble contain clinical data (patientid, age, sex), tissue information (snomedCode, tissueDescription), staining results (staining, intensity, location) and one imageUrl for each

sample. However, due to the large amount of data and the relatively large size of each image, hpaXmlTissueExpr does not provide an automated download option.

```
tissueExpression[[1]]

#> # A tibble: 331 x 18
#>  patientId age sex  staining intensity quantity location imageUrl
#>  <chr>  <chr> <chr> <chr>  <chr>  <chr>  <chr>  <chr>
#> 1 1653   53  Male <NA>  <NA>   <NA>  <NA>  http://v18.p~
#> 2 1721   60  Female <NA>  <NA>   <NA>  <NA>  http://v18.p~
#> 3 1725   57  Male <NA>  <NA>   <NA>  <NA>  http://v18.p~
#> 4 598   7  Male <NA>  <NA>   <NA>  <NA>  http://v18.p~
#> 5 666   82  Female <NA>  <NA>   <NA>  <NA>  http://v18.p~
#> 6 2637   36  Female <NA>  <NA>   <NA>  <NA>  http://v18.p~
#> 7 2667   75  Male <NA>  <NA>   <NA>  <NA>  http://v18.p~
#> 8 2671   72  Female <NA>  <NA>   <NA>  <NA>  http://v18.p~
#> 9 1447   45  Female <NA>  <NA>   <NA>  <NA>  http://v18.p~
#> 10 1452   44  Female <NA>  <NA>   <NA>  <NA>  http://v18.p~
#> # ... with 321 more rows, and 10 more variables: snomedCode1 <chr>,
#> # snomedCode2 <chr>, snomedCode3 <chr>, snomedCode4 <chr>,
#> # snomedCode5 <chr>, tissueDescription1 <chr>, tissueDescription2 <chr>,
#> # tissueDescription3 <chr>, tissueDescription4 <chr>,
#> # tissueDescription5 <chr>
```

CASE STUDIES

To demonstrate the functionality and use of HPAanalyze, we performed case studies as examples. Each case study was based on the literature to show how the resulting data might confirm or complement cancer research.

Case Study 1: Glioma Pathway Alteration at the Protein Level

Previous studies on the landscape of pathway alterations in glioblastoma (GBM) have identified frequent amplification and deletion of certain genes belonging to PI3K/MAPK, p53 and Rb pathways. The level of these proteins in normal brain (hippocampus and cerebral cortex) (Figure 3A) and glioma (Figure 3B) was visualized with HPAanalyze. It should be noted that these data were acquired by quantification of IHC stained specimens, hence point mutations may not be represented. Furthermore, the cancer dataset contains data for all glioma, and information about each specific cancer grade is only available via deep examination of individual antibodies, which is outside of the scope of this case study.

As a positive control, we first evaluated the expression of GFAP, the marker for astrocytes/glial cells. In the normal dataset, GFAP expression was found only on glial cells, suggesting that a false positive is unlikely. In the glioma datasets, GFAP was found to be expressed at mostly medium to high levels, which is also consistent with the literature.

According to the TCGA data, 90% of patients have alterations in the PI3K/MAPK pathway, in which EGFR, PDGFRA and PI3K genes are frequently amplified or mutated to gain function (57%, 10% and 25% respectively), while PTEN is deleted or mutated in 41% of patients (*9*). Data from HPAanalyze supports this pattern, although the proportions are not identical. This can be attributed to the differences between two analyses in term of target molecules (DNA/mRNA in TCGA versus protein in HPA) and the number of specimens. A clear example for this discrepancy can be seen with BRAF, which is only amplified/mutated in 2% of patients in TCGA but expressed at medium to high levels in all glioma specimens in HPA.

The p53 pathway is altered in 86% of patients, with amplification of MDM2 (7.6%) and MDM4 (7.2%) leading to the inhibition of p53, which is also highly mutated (*9*). The amplification of MDM2 and MDM4 is reflected at protein levels in HPA, in which the former expresses at high levels in all patients, and the later at medium in most patients. Similarly, the Rb pathway inhibitor CDK4 was found to be amplified in 14% of patient samples and confirmed by the stark contrast between normal and cancerous samples in HPA. The protein CDK4 is not detected in any normal brain cell, while it is present at some level in most glioma samples.

Case Study 2: PTEN's Novel Function through Chromatin-Associated Complexes

PTEN is known as a key tumor suppressor which is frequently mutated in GBM (*9*). Canonically, the protein functions as a phosphatase to dephosphorylate phosphatidylinositol (3,4,5)-trisphosphate (PIP_3), which leads to inhibition of the Akt signaling pathway (*10*). Akt is central to many hallmarks of cancer by promoting cell survival via inhibition of the apoptotic protein Bad, overcoming cell cycle arrest, facilitating glucose metabolism, inhibiting autophagy

via regulation of lysosomal biogenesis controller TFEB, and promoting tumor angiogenesis (*11*). Since PIP_3 is a phospholipid that resides on the plasma membrane (*11*), PTEN was once thought to act solely in the cytoplasm.

Nevertheless, in the HPA subcellular location dataset, PTEN was found to be presence in both the cytosol and the nucleus (Figure 3C), suggesting a non-canonical function for PTEN. In fact, a recently published study suggested that PTEN also forms complexes with the histone chaperone DAXX and the histone variant H3.3, modulating their association on the chromatin and thereby regulating oncogene expression. This effect is independent of PTEN enzymatic activity (*12*). The subcellular location of DAXX and H3.3, as well as PML which has been shown to interact with DAXX and regulate PTEN, corroborate the newly discovered model of PTEN-DAXX-H3.3 interaction.

It should be noted that HPA subcellular location information for individual proteins is acquired via immunofluorescent staining of human cell lines (*3*). The data therefore does not account for various physiological conditions that may relocate the protein nor does the data directly provide evidence of protein-protein interactions. A query of HPA should always be followed by a confirmation study to ensure the validity of the results in any cell type or cancer of interest. Nevertheless, HPAanalyze offers a powerful approach to quickly explore curated and validated antibody-based protein expression data.

CONCLUSION

With our R package HPAanalyze, we are able to retrieve, visualize and export data from the HPA program. Although it is a programmatic approach, which requires basic R programming skills, HPAanalyze was built with ease of use and reproducibility in mind, which makes the workflow and syntax very simple and straight-forward. With the case studies, we have also demonstrated how HPAanalyze can be easily integrated into different areas of research to

identify new targets or provide more evidence for a working hypothesis. This software package is highly supportive of our research, and we plan to update it with new features and ensure future compatibility with the HPA program.

AVAILABILITY AND REQUIREMENTS

- Project name: HPAanalysis
- Project home page: https://github.com/trannhatanh89/HPAanalyze
- Operating system(s): All platforms where R is available, including Windows, Linux, OS X
- Programming language: R
- Other requirements: R 3.5.0 or higher, and the R packages dplyr, XLConnect, ggplot2, readr, tibble, xml2, reshape2, tidyr, stats, utils, and hpar
- License: MIT
- Any restrictions to use by non-academics: Freely available to everyone

ABRREVIATIONS

GBM: Glioblastoma; HGNC: HUGO Gene Nomenclature Committee; HPA: Human Protein Atlas; IHC: immunohistochemistry; TCGA: The Human Genome Atlas;

DECLARATIONS

Acknowledgements

We appreciate the support of the National institutes of Health R01 NS104339 and R21 NS096531 funds from the Department of Cell, Developmental and Integrative Biology at the University of Alabama at Birmingham.

Authors' contributions

ANT created the R package and drafted the manuscript. ABH supervised the project and revised the manuscript. All authors read and approved the final manuscript.

COPYRIGHT

© Anh Tran et al. 2018

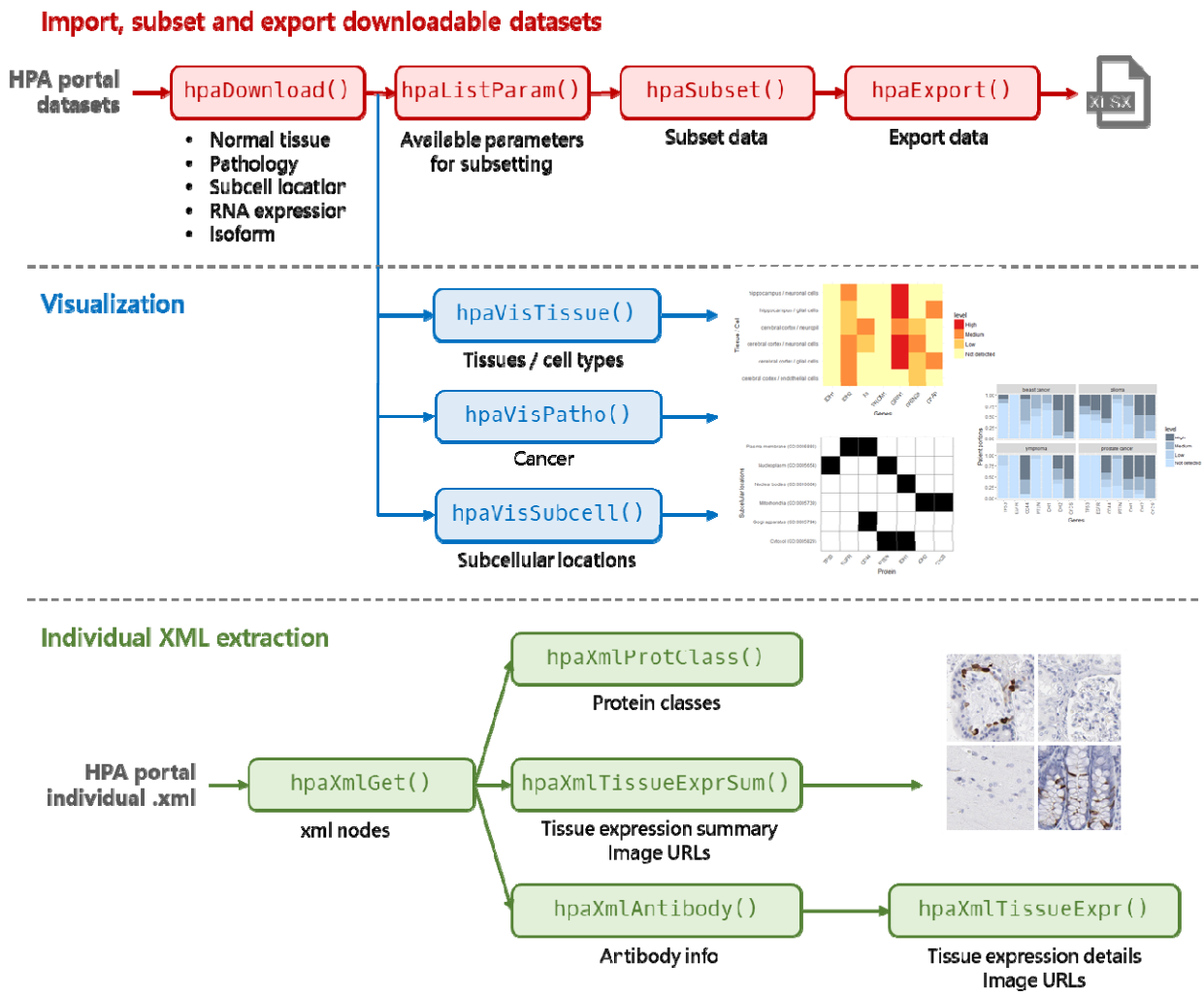

Figure 1: *HPA analyze workflow*.

HPAanalyze provides functions for downloading, extracting and visualizing data from HPA. The functions are divided into three different families: (1) hpaDownload for downloadable datasets, (2) hpaVis for quick and customizable visualization and (3) hpaXml for extracting information from individual XML files.

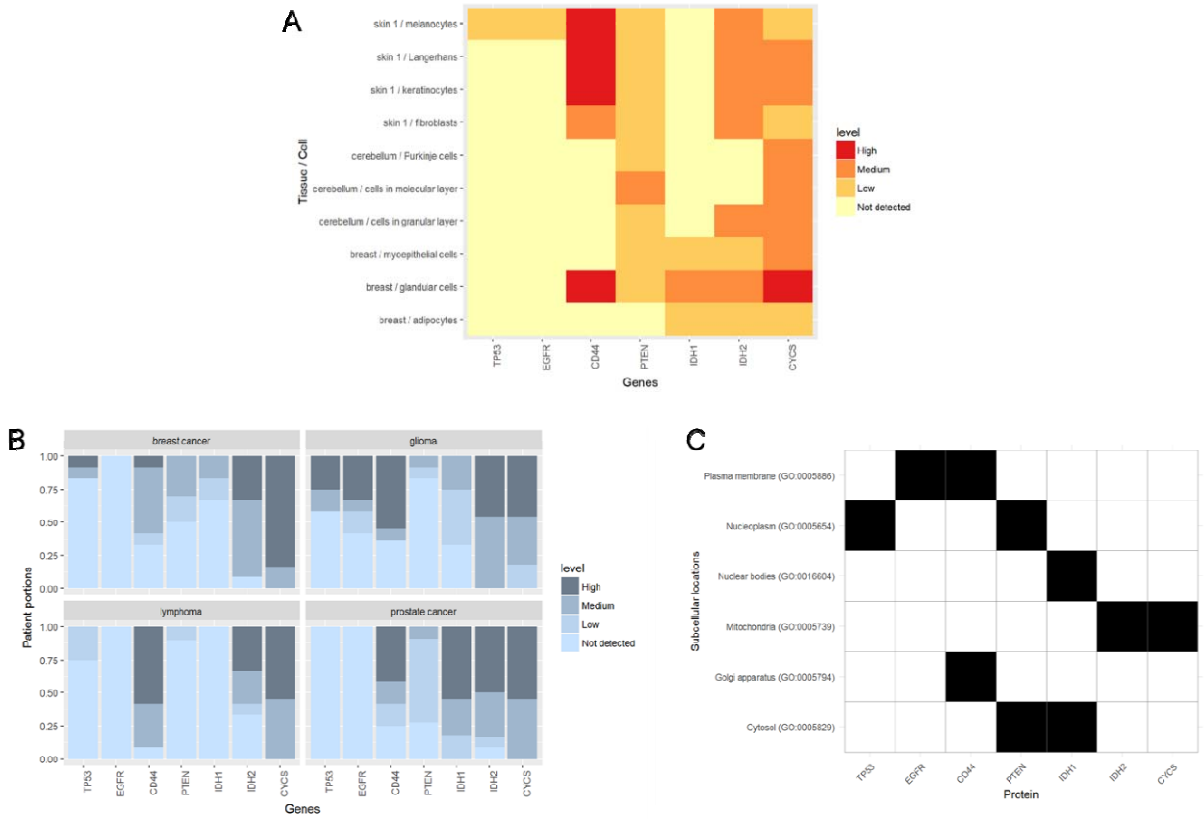

Figure 2: Example of hpaVis function family.

(A) hpaVisTissue() generates a heatmap of protein expression levels in tissues and cells of interest. (B) hpaVisPatho() creates multiple bar graphs of protein expression in individual target cancers. (C) hpaVisSubcell() visualizes subcellular location of proteins of interest in a tile chart.

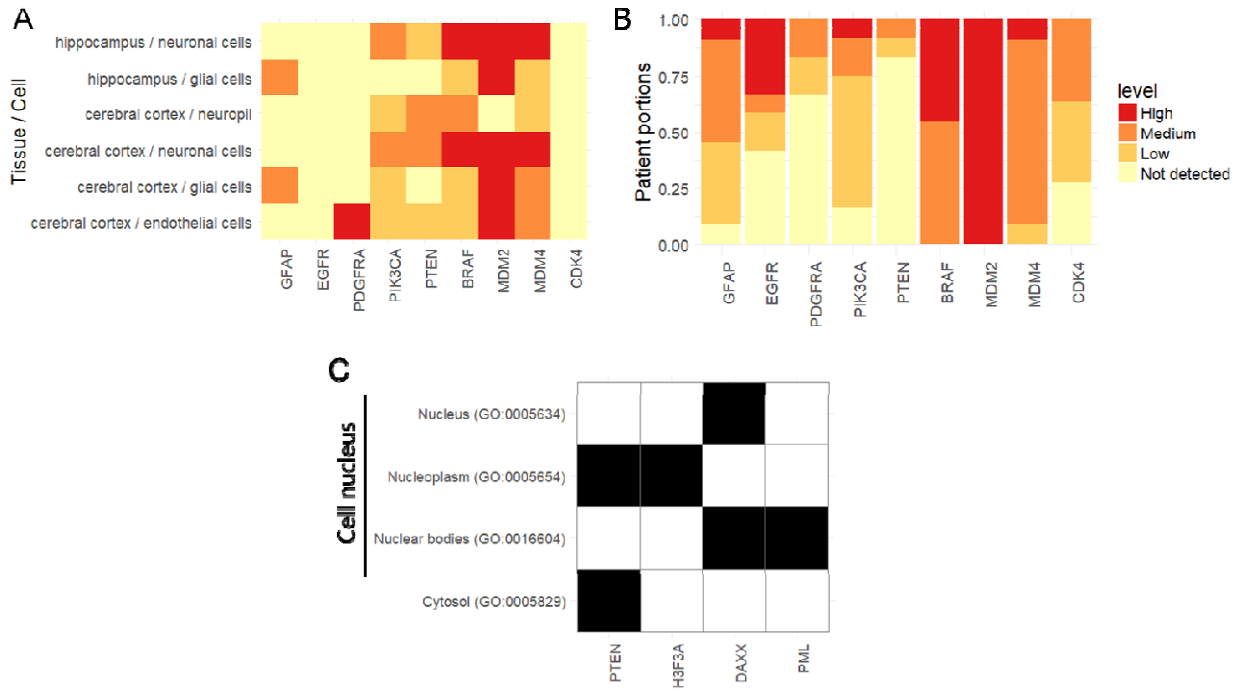

Figure 3: Case studies

(A-B) Expression of proteins frequently altered in GBM. (B) Subcellular localization of proteins associated with PTEN non-canonical functions.

REFERENCES

1. L. Berglund *et al.*, A genecentric Human Protein Atlas for expression profiles based on antibodies. *Mol Cell Proteomics* **7**, 2019-2027 (2008).

2. F. Pontén, K. Jirström, M. Uhlen, The Human Protein Atlas--a tool for pathology. *J Pathol* **216**, 387-393 (2008).

3. P. J. Thul *et al.*, A subcellular map of the human proteome. *Science* **356**, (2017).

4. M. Uhlén *et al.*, A human protein atlas for normal and cancer tissues based on antibody proteomics. *Mol Cell Proteomics* **4**, 1920-1932 (2005).

5. M. Uhlén *et al.*, Proteomics. Tissue-based map of the human proteome. *Science* **347**, 1260419 (2015).

6. M. Uhlen *et al.*, Towards a knowledge-based Human Protein Atlas. *Nat Biotechnol* **28**, 1248-1250 (2010).

7. M. Uhlen *et al.*, A pathology atlas of the human cancer transcriptome. *Science* **357**, (2017).

8. L. Gatto. (Bioconductor, 2018).

9. Cameron W. Brennan *et al.*, The Somatic Genomic Landscape of Glioblastoma. *Cell* **155**, 462-477 (2013).

10. D. Koul, PTEN signaling pathways in glioblastoma. *Cancer Biol Ther* **7**, 1321-1325 (2008).

11. K. R. Auger, L. A. Serunian, S. P. Soltoff, P. Libby, L. C. Cantley, PDGF-dependent tyrosine phosphorylation stimulates production of novel polyphosphoinositides in intact cells. *Cell* **57**, 167-175 (1989).

12. J. A. Benitez *et al.*, PTEN regulates glioblastoma oncogenesis through chromatin-associated complexes of DAXX and histone H3.3. *Nat Commun* **8**, 15223 (2017).