HSRproj writeup

Natalie R Smith, Katherine EM Miller, Bryan S Blette, and Leah Frerichs

Table of Contents

Last Updated 04/06/2020

Packages needed:

* tidyverse
* magrittr
* igraph
* ggnetwork
* ggsci
* graphlayouts
* gdata
* ggraph

# Initial data cleaning

library(tidyverse)  
library(magrittr)  
  
source("code/functions/classify-funders.R")  
source("code/functions/classify-type.R")  
source("code/functions/mesh-manual-classifications.R")  
source("code/functions/classify-funding-amount.R")

Import data

hsrProj <- readxl::read\_excel("data/update-data-download-03-31-20.xlsx")  
  
hsrCleaned <- hsrProj %>%  
   
 # data is imported with an extra row of space, get rid of that  
 filter(!is.na(ProjectID)) %>%  
   
 # these would all be great to have (# subjects, data source), but are all missing in the data download file  
 # others aren't missing but just aren't needed.   
 select(-DataSource, -NumberOfSubjects, -PopulationBase, -StudyDesign,   
 -StudyPopulation, -DataBank, -ArchivedYesNo, -GeneralNotes,   
 -DateRevised, -DatePublished, -POList) %>%  
   
 # make a project length variable  
 # second recode is because some grants begin and end in the same year, so they show up as 0,   
 # but considering these two be one year grants  
 mutate(lengthProj = as.numeric(FinalYear) - as.numeric(InitialYear),   
 lengthProj = if\_else(lengthProj==0, 1, lengthProj))

Classify the primary funder of each study - [see appendix of code for full specification](#funder).

hsrCleaned <- FUNDERS(hsrCleaned)

Classify whether a grant was research or not - [see appendix of code for full specification](#research).

# this function is dependent on who the primary funder is, so must be run after defining primary funders   
hsrCleaned <- TYPES(hsrCleaned)

Calculate the average yearly funding of grants - [see appendix of code for full specification](#fundingamt).

hsrCleaned <- FUNDINGAMT(hsrCleaned)

Subset to only research, and pull the 250 most recent research projects.

hsrCleaned <- hsrCleaned %>%   
   
 # only within those grants that are research  
 filter(isResearch==1) %>%  
   
 # sort so that most recent is first  
 # desc means descending = largest to smallest   
 arrange(desc(InitialYear), desc(InitialMonth)) %>%  
   
 # create running variable (rownames are reset after using arrange)  
 rownames\_to\_column(var="number") %>%  
   
 # convert variables to numeric so they are easier/useful to work with  
 mutate(number = as.numeric(number),   
 ProjectID = as.numeric(ProjectID))   
  
  
# and finally, keep numbers 1-250  
hsrRecent <- filter(hsrCleaned, number<=250)   
  
table(hsrRecent$InitialYear)

##   
## 2013 2014 2015 2016 2017 2018 2019   
## 7 59 51 49 44 30 10

# this piece is needed because some funding numbers weren't reported, but could have been  
# when making Figure 1 - the SEM/funding plot, NS output project ID's that had an SEM classification  
# but were missing funding information. then searched NIH reporter for those ID's to try and find numbers  
# this resulted in some additional data points   
# NOTE: VA funding does not appear to be publicly available so Figure 1 does not include those numbers.   
suppFunding <- readxl::read\_excel("data/manual-funding-numbers.xlsx") %>%  
   
 # just project ID and the averages NS calculated  
 select(ProjectID, avgYearlyFunding) %>%  
   
 # rename for some clearer merging/organization  
 rename(fundingExtra = avgYearlyFunding)  
  
# merge this supplementary funding information on the hsrRecent dataset  
hsrRecent <- hsrRecent %>% left\_join(suppFunding, by="ProjectID")   
  
# combine those two funding pieces  
hsrRecent <- hsrRecent %>%  
   
 # if the avgYearlyFunding is missing (returned from FUNDINGAMT() function above)  
 # then supplement with the fundingExtra variable, otherwise keep it as the original average  
 mutate(avgYearlyFunding = if\_else(is.na(avgYearlyFunding),   
 fundingExtra,   
 avgYearlyFunding))  
  
  
  
  
# not needed anymore, can uncomment if wanted.   
 # write.csv(hsrRecent, file="output/hsrRecent.csv", row.names = FALSE)  
 # write.csv(hsrRecent %>% select(ProjectID, ProjectTitle, Abstract), file="output/hsrRecentForSEM.csv", row.names = FALSE)  
 # write.csv(hsrRecent %>% select(ProjectID), file="output/hsrRecentIDs.csv", row.names = FALSE)  
  
# remove unneeded datasets for a cleaner workspace  
rm(hsrProj, # original data  
 hsrCleaned, # full data  
 suppFunding # supplementary funding info  
 )

Identifying the most common funders of diabetes research, within the 250 most recent projects.

tableone::CreateCatTable(vars=c("primaryFunder"), data=hsrRecent)

##   
## Overall   
## n 250   
## primaryFunder (%)   
## AHRQ 27 (10.8)   
## CDC 10 ( 4.0)   
## HRSA 1 ( 0.4)   
## NIDILRR 3 ( 1.2)   
## NIH 152 (60.8)   
## PCORI 21 ( 8.4)   
## PFIZER 1 ( 0.4)   
## VA 35 (14.0)

tableone::CreateCatTable(vars=c("NIHspecific"), data=hsrRecent)

##   
## Overall   
## n 250   
## NIHspecific (%)   
## NCCIH 1 ( 0.4)   
## NCI 4 ( 1.6)   
## NEI 1 ( 0.4)   
## NHGRI 2 ( 0.8)   
## NHLBI 15 ( 6.0)   
## NIA 14 ( 5.6)   
## NICHD 1 ( 0.4)   
## NIDCR 1 ( 0.4)   
## NIDDK 75 (30.0)   
## NIH-OD 1 ( 0.4)   
## NIMH 3 ( 1.2)   
## NIMHD 11 ( 4.4)   
## NINDS 1 ( 0.4)   
## NINR 13 ( 5.2)   
## NLM 9 ( 3.6)   
## NotSpecifiedNotNIH 98 (39.2)

# Social-ecological classification

## Import and data management

# some code is deleted here for readability - this old code output the combination of new projects from the 900 dataset  
# and old projects that already had classifications. the SEM for update 900 csv was output and reclassified as needed.   
  
  
SEMOrig <- read.csv("output/SEMforUPDATE900-classified.csv")  
# there are 323 records here, but only 250 should be included  
 table(SEMOrig$fromUpdate900)

##   
## 1   
## 250

# 227 should be included   
 table(SEMOrig$fromUpdate900, SEMOrig$Not.Applicable, useNA = "always")

##   
## 1 <NA>  
## 1 23 227  
## <NA> 1 72

SEM <- SEMOrig %>%  
   
 # get rid of the three that should not have been included  
 # want to do programatically rather than deleting from excel file   
 # after the update, this gets rid of OLD IDs that aren't needed anymore  
 filter(ProjectID %in% hsrRecent$ProjectID) %>%  
   
 # redundant, but just in case  
 filter(fromUpdate900==1) %>%  
   
   
 # keep records that are NOT not applicable  
 filter(is.na(Not.Applicable)) %>%  
   
   
 # combine more granular levels of the SEM classification  
 # during coding, it was very difficult to distinguish and for both display purposes and data confidence we are combining them  
 mutate(people = case\_when(Individual==1 | Interpersonal==1 ~ "1people", TRUE ~ NA\_character\_),   
 places = case\_when(Organizational==1 | Community==1 ~ "2places", TRUE ~ NA\_character\_),   
 policy = case\_when(Policy==1 ~ "3policy", TRUE ~ NA\_character\_)) %>%  
   
 # remove variables that are not needed for analysis  
 select(-Description, -Not.Applicable, -Flag, -haveSEM, -fromUpdate900,   
 -ProjectTitle, -Abstract)  
  
  
# convert from 'wide' dataset to 'long' dataset because some projects targeted more than 1 of the 3 levels  
SEMlong <- SEM %>%  
   
 # converting from wide to long  
 gather(From, Level, people, places, policy) %>%  
   
 # only want the projectID and the 'level', which contains either people/places/policy  
 select(ProjectID, Level) %>%  
   
 # keep only those rows where the people/places/policy variable exists  
 # some are NA because of the wide-long conversion  
 filter(!is.na(Level)) %>%  
   
 # create a factor variable that will be the facet for future dotplot  
 # add information in the label that will show up in the facet description in the plot - see  
 # SEMavgs dataset printed later for these numbers  
 mutate(Level = factor(Level, levels=c("1people",   
 "2places",   
 "3policy"),  
   
 labels = c("1people" = "Individual & Interpersonal \n avg= $413,879 n=158",   
 "2places" = "Organizational & Community \n avg= $312,060 n=45",  
 "3policy" = "Policy \n avg= $332,784 n=14"))) %>%  
   
 # make the level variable name more informative  
 rename(SEM = Level)   
  
# identify how many of the 3 levels a project targeted  
# this will be used to split the funding up later, so that we are not double counting any dollars  
tmp <- SEMlong %>%   
   
 # for each projectID  
 group\_by(ProjectID) %>%   
   
 # identify how many rows there are  
 summarise(num = n())   
  
# add that number variable to the SEMlong dataset  
SEMlong <- left\_join(SEMlong, tmp, by="ProjectID")  
  
  
# now merge on funding from the hsrRecent dataset  
SEMlong <- left\_join(SEMlong,   
 hsrRecent %>% select(ProjectID, avgYearlyFunding),   
 by="ProjectID")  
   
# now create the final dataset for graphs  
SEMgraph <- SEMlong %>%  
  
 # this project is an very big outlier - the GRADE study  
 # budget for 2012 is listed as 23,000,000 (23 million)  
 filter(ProjectID != 20153570) %>%  
   
 # in NS manual classification -1 denotes missing funding after searching NIH reporter,   
 # so make sure those are not included  
 filter(avgYearlyFunding > 0) %>%  
   
 # divide avgYearlyFunding equally among levels targeted  
 # this probably underestimates the difference between each of the 3 levels, because  
 # individual/interpersonal aims (probably) require more funding that comm/org or policy  
 mutate(avgYearlyFunding = avgYearlyFunding/num)  
  
  
  
SEMavgs = SEMgraph %>%  
   
 # for each level  
 group\_by(SEM) %>%  
   
 # calculate the mean (primary), median (sensitivity), and total n  
 summarise(mean = mean(avgYearlyFunding),   
 total = n(),   
 median=median(avgYearlyFunding))  
  
print(SEMavgs)

## # A tibble: 3 x 4  
## SEM mean total median  
## <fct> <dbl> <int> <dbl>  
## 1 "Individual & Interpersonal \n avg= $413,879 n=15~ 413879. 158 3.41e5  
## 2 "Organizational & Community \n avg= $312,060 n=45" 312060. 45 2.84e5  
## 3 "Policy \n avg= $332,784 n=14" 332784. 14 4.48e5

# of studies where the SEM was applicable, these are missing funding information  
SEMmissing = SEMlong %>%  
   
 # now, we just want to examine how many we are missing funding info for - for caption  
 filter(avgYearlyFunding==-1 | is.na(avgYearlyFunding)) %>%  
   
 # for each level  
 group\_by(SEM) %>%  
   
 # count number of missing  
 summarise(numMissing = n())  
  
print(SEMmissing)

## # A tibble: 3 x 2  
## SEM numMissing  
## <fct> <int>  
## 1 "Individual & Interpersonal \n avg= $413,879 n=158" 27  
## 2 "Organizational & Community \n avg= $312,060 n=45" 15  
## 3 "Policy \n avg= $332,784 n=14" 4

nrow(distinct(SEMgraph, ProjectID))

## [1] 192

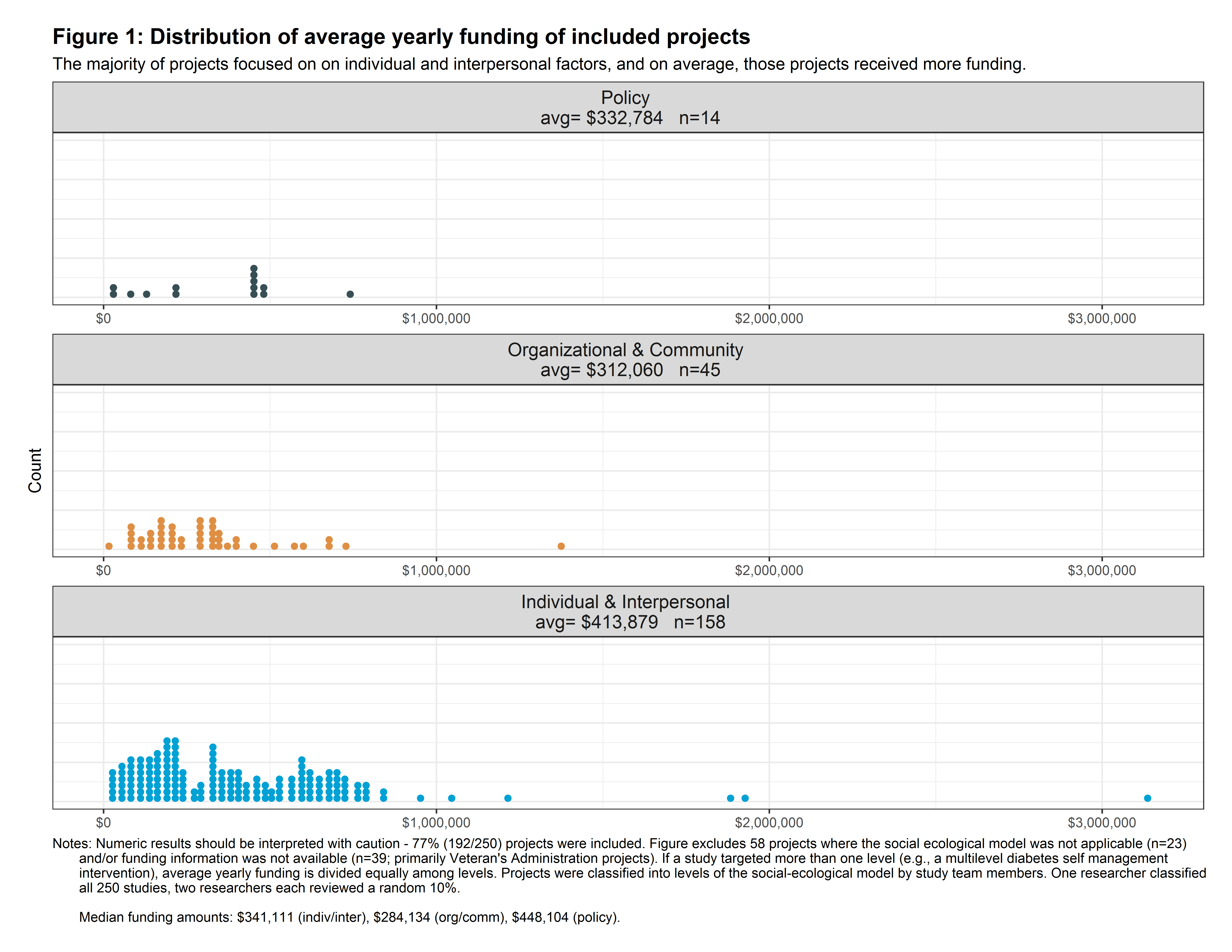
rm(SEMOrig, SEM, tmp, SEMlong, SEMavgs, SEMmissing)

## Visualization

ggplot(data=SEMgraph, aes(x=SEMgraph$avgYearlyFunding, # avg funding on the x-axis  
 colour=fct\_rev(SEMgraph$SEM), # color (OUTLINE) dots by the three SEM levels (reversing, so policy is on top)  
 fill=fct\_rev(SEMgraph$SEM))) + # fill (INSIDE) dots by the three SEM levels  
   
 # use JAMA color scheme for the outline and fill colors  
 ggsci::scale\_color\_jama() +   
 ggsci::scale\_fill\_jama() +   
   
 # display separate graph for each of the three SEM levels  
 lemon::facet\_rep\_wrap(fct\_rev(SEMgraph$SEM), # again, reversing so policy is on top  
 nrow=3, ncol=1, # stack on top of each other  
 scales = "fixed",   
 repeat.tick.labels=TRUE) + # I want an x-axis shown for each facet  
   
 # informative labels and captions, etc.  
 labs(y="Count",   
   
 title="Figure 1: Distribution of average yearly funding of included projects",   
   
 subtitle="The majority of projects focused on on individual and interpersonal factors, and on average, those projects received more funding.",  
   
 caption="Notes: Numeric results should be interpreted with caution - 77% (192/250) projects were included. Figure excludes 58 projects where the social ecological model was not applicable (n=23)   
 and/or funding information was not available (n=39; primarily Veteran's Administration projects). If a study targeted more than one level (e.g., a multilevel diabetes self management   
 intervention), average yearly funding is divided equally among levels. Projects were classified into levels of the social-ecological model by study team members. One researcher classified   
 all 250 studies, two researchers each reviewed a random 10%.  
   
 Median funding amounts: $341,111 (indiv/inter), $284,134 (org/comm), $448,104 (policy).") +   
   
 # simple black and white theme  
 theme\_bw() +   
   
 # alter specific theme elements, like font size and alignment  
 theme(legend.position = "none",   
 axis.text.y = element\_blank(),   
 axis.ticks.y = element\_blank(),   
 strip.text.x = element\_text(size=12),   
 plot.title = element\_text(size=14, hjust = 0, face="bold"),   
 plot.subtitle = element\_text(size=11, hjust=0),   
 plot.caption = element\_text(hjust=0),   
 axis.title.x = element\_blank(),   
 plot.margin = margin(t = 0.25, r = 0.25, b = 0.25, l = 0.25, unit = "in")) +  
   
 # format the x-axis to look like dollars  
 scale\_x\_continuous(labels = scales::dollar\_format()) +  
   
 # and FINALLY, add the data via a dotplot. I have adjusted the sizes and binwidth iteratively until I like the sizing in the exported graphs  
 geom\_dotplot(binwidth = 24000, dotsize=0.8)

# save as both vector and raster formats  
ggsave(filename = "output/Fig-1-SEM-funding.pdf", plot=last\_plot(), device="pdf",   
 units="in", width=11, height=8.5)  
  
ggsave(filename = "output/Fig-1-SEM-funding.png", plot=last\_plot(), device="png",   
 units="in", width=11, height=8.5, dpi=500)  
  
  
  
rm(SEMgraph)

include\_graphics("output/Fig-1-SEM-funding.png")



# MeSH concept network

## MeSH cleaning and classification into concepts

Get a nice dataset of MeSH terms that are simplified.

mesh <- hsrRecent %>%   
   
 # focusing on the mesh headings for each project  
 select(ProjectID, MeshHeading) %>%  
   
 # remove trailing characters like "</li> <li>"  
 mutate(meshTrimmed = str\_sub(MeshHeading, start=10, end=-11),   
 listV2 = str\_split(MeshHeading, pattern="</li> <li>"),   
 numMeSH = map\_int(listV2, length))  
  
# distribution of mesh terms in each research project  
table(mesh$numMeSH)

##   
## 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 32   
## 1 1 2 2 7 10 19 25 28 22 32 23 23 18 17 9 3 5 1 1 1

mesh <- mesh %>%  
   
 # separate this trimmed mesh term into new columns named A-Z  
 # 32 is the max number of mesh terms, so alphabet + 6 combos works nicely   
 separate(meshTrimmed, into= c(LETTERS[seq(from=1, to=26)], "AA", "BB", "CC", "DD", "EE", "FF"),   
   
 # this separator was determined by looking at the data  
 sep="</li> <li>") %>%  
   
 # remove the mesh heading variable where all mesh terms are stored now that they are split  
 select(-MeshHeading)

## Warning: Expected 32 pieces. Missing pieces filled with `NA` in 249  
## rows [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  
## 20, ...].

meshLong <- mesh %>%  
   
 # convert from wide dataset to long dataset  
 gather(Letter, MeSH, A:Z) %>%  
   
 # remove this, we don't actually care about the numbers  
 select(-Letter) %>%  
   
 # drop the rows that are NA (induced by the separate function used above)  
 filter(!is.na(MeSH)) %>%  
   
 # these next couple steps work to clean up the mesh terms with qualifiers  
 # takes words like "Diabetes Mellitus /\* prevention & control </li> /\*management"  
 # and makes two words - Diabetes Mellitus: prevention and control  
 # and Diabetes Mellitus: management  
 mutate(flag1 = str\_detect(MeSH, pattern="\\</li>"),   
 flag2 = str\_detect(MeSH, pattern="\\/"),  
 count = str\_count(MeSH, pattern="\\</li>")) %>%   
   
 # separate those complex individual mesh words  
 # I know to use four because there are max three qualifiers, plus the first word  
 separate(MeSH, into = c("first", "second", "third", "fourth"), sep="\\</li>") %>%  
 separate(first, into=c("before", "after"), sep="/") %>%  
   
 # trim the asterisks and slashes  
 # paste function here just used to look for a couple different patterns at a time  
 mutate(before=str\_trim(str\_remove(before, pattern="\\\*")),   
 after=str\_trim(str\_remove(after, pattern="\\\*")),   
 second=str\_trim(str\_remove(second, pattern=paste(c("\\/\\\*", "\\/"), collapse = "|"))),   
 third=str\_trim(str\_remove(third, pattern=paste(c("\\/\\\*", "\\/"), collapse = "|"))),   
 fourth=str\_trim(str\_remove(fourth, pattern=paste(c("\\/\\\*", "\\/"), collapse = "|"))))

## Warning: Expected 4 pieces. Missing pieces filled with `NA` in 4441  
## rows [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  
## 20, ...].

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 3470 rows  
## [1, 3, 4, 5, 7, 10, 11, 12, 14, 15, 16, 19, 20, 23, 24, 25, 26, 27, 28,  
## 29, ...].

# now if a given mesh word (remember, this dataset is in long form) does not have qualifiers, it's good as is  
meshNoQualifiers <- meshLong %>%   
   
 # BOTH flags must be false   
 filter(flag1==FALSE & flag2==FALSE) %>%  
   
 # rename the word to be called MeSH  
 rename(MeSH=before) %>%  
   
 # keep just needed variables  
 select(ProjectID, MeSH)  
  
  
  
# now for the mesh terms that do have qualifiers  
meshQualifiers <- meshLong %>%  
   
 # either flag needs to be true  
 filter(flag1==TRUE | flag2==TRUE) %>%  
   
 # re-append the mesh qualifiers to the original word (called 'before')  
 # this creates the multiple terms we need based on qualifiers  
 # (This is unnecessary since we work with the simplified mesh terms anyways...)  
 mutate(mesh1 = str\_c(before, after, sep=": "),   
 mesh2 = str\_c(before, second, sep=": "),   
 mesh3 = str\_c(before, third, sep=": "),   
 mesh4 = str\_c(before, fourth, sep=": ")) %>%  
   
 # keep just the new mesh terms  
 select(ProjectID, mesh1, mesh2, mesh3, mesh4) %>%  
   
 # convert to long data format again (long x2!)  
 gather(v1, v2, mesh1:mesh4) %>%  
   
 # remove unnecessary vars and rename to be MeSH  
 select(-v1) %>%  
 rename(MeSH=v2)  
  
  
# append the qualifier and no qualifier datasets into one large dataset  
# this dataset has just projectID and cleaned up mesh terms  
meshCleaned <- rbind(meshNoQualifiers, meshQualifiers)  
  
# now simplify those mesh terms  
meshCleaned <- meshCleaned %>%  
   
 # remove everything about the ":"  
 mutate(MeSHSimple = gsub(": .\*","", MeSH)) %>%  
   
 # get rid of NA's induced by the transposing  
 filter(!is.na(MeSH))  
  
  
# This is a key dataset - subsets to only MeSH terms that occur more than 2 times   
# this gets entered into the classification code   
meshSimpleReduced <- meshCleaned %>%  
   
 # for each simplified mesh term from above  
 group\_by(MeSHSimple) %>%   
   
 # count how many times it occurred  
 summarise(meSHoccurrences=n()) %>%  
   
 # keep only those that are not missing (redundant)  
 filter(!is.na(MeSHSimple)) %>%  
   
 # get rid of the Humans mesh term - all are tagged with it  
 filter(MeSHSimple!="Humans") %>%  
   
 # keep only simplified mesh terms that occur 3 or more times  
 filter(meSHoccurrences>2)  
  
  
rm(mesh, meshLong, meshNoQualifiers, meshQualifiers)

Classify MeSH terms into broader concepts. [See the appendix code for granular classifications](#concepts).

# and now finally we can aggregate these simple mesh terms into larger CONCEPTS   
# see the classification code for the vectors that define what words go in to specific concepts  
meshClassified <- meshSimpleReduced %>%   
   
 mutate(Classification = case\_when(  
   
 # Diabetes Types  
 # Almost too general, will likely be dropped later because MANY articles are tagged with it  
 MeSHSimple %in% "Diabetes Mellitus" ~ "Diabetes \n General",   
   
 # Sub-types  
 MeSHSimple %in% "Diabetes Mellitus, Type 1" ~ "Type 1 \n Diabetes",   
 MeSHSimple %in% "Diabetes Mellitus, Type 2" ~ "Type 2 \n Diabetes",   
 MeSHSimple %in% "Diabetes, Gestational" ~ "Gestational \n Diabetes",   
   
 # Related  
 MeSHSimple %in% "Prediabetic State" ~ "Prediabetes",   
 MeSHSimple %in% "Diabetes Complications" ~ "Diabetes \n Complications",   
 MeSHSimple %in% RESPIRATORY ~ "Respiratory \n Related",  
 MeSHSimple %in% KIDNEY ~ "Kidney \n Related",  
  
   
 # Comorbidities  
 MeSHSimple %in% CHRONCOMORBID ~ "Chronic Disease \n Comorbidity",   
 MeSHSimple %in% CARDIOV ~ "Cardiovascular",   
 MeSHSimple %in% BP ~ "Blood \n Pressure",  
 MeSHSimple %in% BLOODGLUCOSE ~ "BloodGlucose/Insulin",  
 MeSHSimple %in% BLOODLIPIDS ~ "Blood \n Lipids",   
  
 # Other potentially related diseases/issues  
 MeSHSimple %in% HIV ~ "HIV",   
 MeSHSimple %in% SLEEP ~ "Sleep",   
 MeSHSimple %in% OSTEO ~ "Osteoarthritis",   
 MeSHSimple %in% ALZ ~ "Alzheimers",  
  
 # Risk factors  
 MeSHSimple %in% WEIGHT ~ "Weight \n Related",  
 MeSHSimple %in% WEIGHTLOSS ~ "Weight \n Loss",  
 MeSHSimple %in% DIET ~ "Diet",   
 MeSHSimple %in% MENTALHEALTH ~ "Mental \n Health",   
 MeSHSimple %in% EXERCISE ~ "PA/Exercise",  
  
   
   
 # Populations  
 MeSHSimple %in% VETERANS ~ "Veterans",   
 MeSHSimple %in% FEMALE ~ "Women",   
 MeSHSimple %in% MALE ~ "Men",   
 MeSHSimple %in% YOUNG ~ "Children \n AdolescentYA",   
 MeSHSimple %in% AGED ~ "Older \n Adults",   
  
 MeSHSimple %in% CARE ~ "Caregivers",   
 MeSHSimple %in% DISABLE ~ "Disabled \n Persons",  
   
 MeSHSimple %in% AA ~ "African \n Americans",   
 MeSHSimple %in% HISPANIC ~ "Hispanic \n Americans",   
 MeSHSimple %in% AIAN ~ "Native \n Americans",   
 MeSHSimple %in% ASIAN ~ "Asian \n Americans",   
 MeSHSimple %in% MINORITY ~ "Minority \n Population",   
   
 MeSHSimple %in% URBAN ~ "Urban",  
 MeSHSimple %in% RURAL ~ "Rural \n Health",   
   
   
 MeSHSimple %in% HSR ~ "HSR",   
 MeSHSimple %in% ACCESS ~ "Access",   
 MeSHSimple %in% DELIVERY ~ "Delivery",  
 MeSHSimple %in% QUALITY ~ "Quality",  
 MeSHSimple %in% COST ~ "Cost",  
 MeSHSimple %in% PATIENT ~ "Patient \n Centered",   
 MeSHSimple %in% DISPARITIES ~ "Disparities",   
 MeSHSimple %in% GEOGRAPHY ~ "Geography",  
 MeSHSimple %in% DECISION ~ "Decision \n Support",  
 MeSHSimple %in% "Telemedicine" ~ "Telemedicine",  
 MeSHSimple %in% CONTINUITY ~ "Continuity \n of Care",   
 MeSHSimple %in% MEDADHERE ~ "Medication \n Adherence",  
   
   
 MeSHSimple %in% INDIVIDUAL ~ "Individual \n Focus",  
 MeSHSimple %in% INTERPERSONAL ~ "Interpersonal \n Focus",  
 MeSHSimple %in% COMMUNITY ~ "Community \n Focus",   
 MeSHSimple %in% POLICY ~ "PolicyFocus",  
   
   
   
 MeSHSimple %in% HEALTHCOMMPROM ~ "HealthComm \n Promotion",   
 MeSHSimple %in% PCAREPREVENTION ~ "PrimaryCare \n Prevention",   
 MeSHSimple %in% INSURANCE ~ "Insurance",   
 MeSHSimple %in% EHR ~ "Health Records \n Including Electronic",   
 MeSHSimple %in% ELECTRONIC ~ "eHealth \n mHealth",   
 MeSHSimple %in% OUTCOMES ~ "Outcomes \n Related",   
 MeSHSimple %in% HOSPITALS ~ "Hospital \n Inpatient",   
 MeSHSimple %in% SES ~ "SES related",   
 MeSHSimple %in% GIS ~ "Geography/GIS",   
 MeSHSimple %in% PRACTICIONERS ~ "Practicioners",   
 MeSHSimple %in% PREGNANCY ~ "Pregnancy \n Related",   
 MeSHSimple %in% AMBULATORY ~ "Ambulatory \n Outpatient",   
 MeSHSimple %in% MUA ~ "Medically \n Underserved Area",   
 MeSHSimple %in% EBM ~ "EBM or \n Guidelines",   
 MeSHSimple %in% PROGRAMS ~ "Program \n Development",   
 MeSHSimple %in% QI ~ "Quality \n Improvement",  
 MeSHSimple %in% SDOH ~ "Social Determinants \n of Health",  
   
 # Methods related  
 MeSHSimple %in% RCT ~ "RCT",   
 MeSHSimple %in% OBSERVATIONAL ~ "Observational",   
 MeSHSimple %in% ECONOMICSMETRICS ~ "Economics \n Econometrics",  
 MeSHSimple %in% MACHINELEARNINGRELATED ~ "MachineLearning \n Related",  
 MeSHSimple %in% QUALITATIVE ~ "Qualitative",   
 MeSHSimple %in% SURVEYPSYCHO ~ "Survey Design \n Psychometrics",   
 MeSHSimple %in% ANALYSIS ~ "Analysis Methods",  
  
 # Funders  
 MeSHSimple %in% NIDDK ~ "NIDDK",   
 MeSHSimple %in% AHRQ ~ "AHRQ",   
 MeSHSimple %in% CDC ~ "CDC",   
 MeSHSimple %in% CMS ~ "CMS",   
 MeSHSimple %in% WHO ~ "WHO",   
 MeSHSimple %in% VA ~ "VA",  
   
   
 TRUE ~ "zUnclassified")) %>%  
   
 select(MeSHSimple, Classification) %>%  
   
 arrange(Classification)  
  
# not necessary anymore, just output this to show the team what words were unclassified  
#write.csv(filter(meshClassified, Classification=="zUnclassified"), file="output/meshUnclassified.csv", row.names = FALSE)   
  
# get rid of all the mesh classifications  
gdata::keep(hsrRecent, meshCleaned, meshSimpleReduced, meshClassified, sure = TRUE)

## Create network data

Using classified concepts, create vertices and edges for MeSH concept network

# merge the classifications onto the full dataset of cleaned mesh terms  
# for each individual mesh term, merge based on the simple version  
# that is linked to the concept classification  
meshTotal <- left\_join(meshCleaned,   
 meshClassified,   
 by="MeSHSimple")  
  
  
meshClassOnly <- meshTotal %>%  
   
 # keep only rows that had a classification (which means they had >2 occurrences)  
 filter(!is.na(Classification)) %>%  
   
 # remove unnecessary columns  
 select(-MeSH, -MeSHSimple) %>%  
   
 # keep only distinct rows - so deletes duplicate classifications within the same projectID (e.g., two similar mesh terms that were grouped together)  
 distinct()  
  
  
# make the 'nodelist' for the network - list of all the vertices/dots  
meshNodes <- meshClassOnly %>%  
   
 # for each unique classification  
 group\_by(Classification) %>%  
   
 # count how many times it occurs   
 summarize(vsize = n())   
# we can do this now, it's alright  
# the edge list manipulations below don't induce any isolates, I checked   
# and even if there are isolates now, we will get rid of them in the graph steps  
# and/or igraph would be fussy and refuse to create a graph object for us  
  
  
  
  
# now we can start making the edgelist for the network  
meshWide <- meshClassOnly %>%  
   
 # within each project ID  
 group\_by(ProjectID) %>%  
   
 # create a running variable for the classifications to use when transposing  
 mutate(id = 1:n()) %>%  
   
 # go back to wide form (needed for linkages)  
 spread(key=id, value=Classification)  
  
  
# now join that wide dataset back to the dataset with unique mesh terms only  
# for one project with 3 mesh terms, now looks like this:   
 # A mesh1 mesh1 mesh2 mesh3  
 # A mesh2 mesh1 mesh2 mesh3  
 # A mesh3 mesh1 mesh2 mesh3  
tmp1 <- left\_join(meshClassOnly,   
 meshWide,   
 by="ProjectID")  
  
# this can now be used to make an 'edgelist'  
meshEdges <- tmp1 %>%  
   
 # transpose so that EACH linkage is its own row  
 # call the merged classification 'class2'  
 gather(colnum, class2, `1`:`19`) %>%  
   
 # and the original classification should be named class1  
 rename(class1 = Classification) %>%  
   
 # remove colnum  
 select(-colnum) %>%  
   
 # remove induced NAs from gather  
 filter(!is.na(class1) & !is.na(class2)) %>%  
   
 # remove self loops  
 filter(class1 != class2) %>%  
   
 # for each unique linkage  
 group\_by(class1, class2) %>%  
   
 # count how many times that linkage occurred  
 summarise(eweight=n()) %>%  
   
 # arrange for niceness  
 arrange(class1, class2) %>%  
   
 # keep only linkages/connections that occured in more than 3 projects  
 filter(eweight>3)  
# one issue that isn't fixed  
# a-b and b-a linkages are still in here -- that's ok, we will fix this in the next step   
# wouldn't even be a big deal if we didn't fix it, since for visualization purposes our edges would lay on top of each other  
# only thing it would affect is the community detection algorithm, and it would be consistent across all edges  
  
  
# just need the mesh edges and mesh nodes datasets now   
rm(meshClassOnly, meshWide, tmp1, meshClassified, meshSimpleReduced, meshCleaned)

## Detect network subgroups

library(igraph)  
library(ggnetwork)  
library(ggraph)  
library(graphlayouts)  
  
# create our igraph object!  
meshGraph <- graph\_from\_data\_frame(meshEdges,  
 directed=FALSE,   
 vertices = meshNodes)  
  
rm(meshNodes, meshEdges)  
  
# simplify to remove the duplicate a-b and b-a linkages  
# specifying to take the average edgeweight of each simplified link pair, which should be the average of two identical numbers  
meshGraphSimple <- igraph::simplify(meshGraph,  
 remove.multiple = TRUE,  
 edge.attr.comb = "mean")  
rm(meshGraph)  
  
# remove nodes/concepts that are too broad and/or connected to be useful, or are unclassified   
meshGraphSimpleSubset <- induced\_subgraph(meshGraphSimple,   
 vids=!(V(meshGraphSimple)$name %in% c("zUnclassified",   
 "Diabetes \n General",   
 "Geography",   
 "Outcomes \n Related",   
 "Individual \n Focus",   
 "Program \n Development",   
 "HealthComm \n Promotion",   
 "AHRQ", "NIDDK", "CMS", "VA")))  
rm(meshGraphSimple)  
  
# and now remove the few concepts that become isolated concepts after this  
# e.g., osteoarthritis  
meshGraphSimpleSubset <- induced\_subgraph(meshGraphSimpleSubset,   
 vids=degree(meshGraphSimpleSubset)>0)  
  
  
  
  
# and now we can detect communities within this network!  
# only want to run the algorithm on NS desktop for reproducibility  
# this code is kept in here for transparency, but have commented out since we have final results now  
# if (Sys.info()[["nodename"]]=="DESKTOP-B5DVP0R" & file.exists("output/SPINGLASS\_RESULTS.Rdata")==FALSE){  
 # set.seed(28215)  
 # spgl <- cluster\_spinglass(meshGraphSimpleSubset, E(meshGraphSimpleSubset)$eweight, gamma=1.2)  
 # save(spgl, file="output/SPINGLASS\_RESULTS.Rdata")  
# } else {  
# load("output/SPINGLASS\_RESULTS.Rdata")  
# }  
  
  
# load the community detection results  
load("output/SPINGLASS\_RESULTS.Rdata")  
  
# there were originally 7 communities/themes  
sizes(spgl)

## Community sizes  
## 1 2 3 4 5 6 7 8   
## 8 7 9 3 13 2 12 10

# add these classifications to the network  
meshGraphSimpleSubset <- igraph::set\_vertex\_attr(meshGraphSimpleSubset,   
 name = "community",   
 value=membership(spgl))  
rm(spgl)  
  
  
# now to do a little data manipulation for final themes and visualization  
# get nodes and edges  
tmp1edges <- igraph::as\_data\_frame(meshGraphSimpleSubset, what="edges")  
  
# within nodes...  
tmp1nodes <- igraph::as\_data\_frame(meshGraphSimpleSubset, what="vertices") %>%   
   
 # just keep the community  
 select(name, community) %>%  
   
 # RECLASSIFY COMMUNITIES  
 # this combination was decided upon by NS, KM, and BB in a team meeting  
 # some communities seem to focus on pretty different subsets   
 # so are combined into the same theme for interpretation and visualization  
 mutate(commF = factor(community, levels=c("1", "2", "3", "4", "5", "6", "7", "8"),   
 labels=c("1" = "Diabetes management in veterans",   
 "2" = "Complications and comorbidity management",   
 "3" = "Decision Support",   
 "4" = "T1D management",   
 "5" = "T2D management 1",   
 "6" = "Lipids & pressure",   
 "7" = "T2D prevention",   
 "8" = "T2D management 2")),   
   
 hsrCommunity = case\_when(# Diabetes and associated complication management  
 community==1 ~ 1,   
 community==2 ~ 1,   
   
 # Decision Support  
 community==3 ~ 2,   
   
 #t1d management  
 community==4 ~ 3,   
   
 #t2d management  
 community==5 ~ 4,   
 community==6 ~ 4,   
 community==8 ~ 4,  
   
 # prevention  
 community==7 ~ 5   
 ),  
  
 # now create a factor variable with the name of the community  
 hsrCommF = factor(hsrCommunity, levels=c("1", "2", "3", "4", "5"),  
 labels = c("1" = "Diabetes and complication management",  
 "2" = "Decision support",  
 "3" = "T1D management",  
 "4" = "T2D management",  
 "5" = "T2D prevention")))  
   
  
# upweight edges if they are in the same community  
# join the nodelist to the edgelist - get community designation for the FROM concept  
tmp2edges <- left\_join(tmp1edges,   
 (tmp1nodes %>% select(name, community)),   
 by=c("from"="name")) %>%  
   
 # and now repeat, to get community designation for the TO concept  
 left\_join((tmp1nodes %>% select(name, community)),   
 by=c("to" = "name")) %>%  
   
 # and finally, upweight if the communities are the same  
 mutate(UPweight = if\_else(community.x==community.y, 60, 1))  
  
  
  
  
meshGraphNew <- graph\_from\_data\_frame(d=tmp2edges,   
 directed=FALSE,   
 vertices=tmp1nodes)  
  
rm(meshGraphSimpleSubset, tmp1edges, tmp2edges)

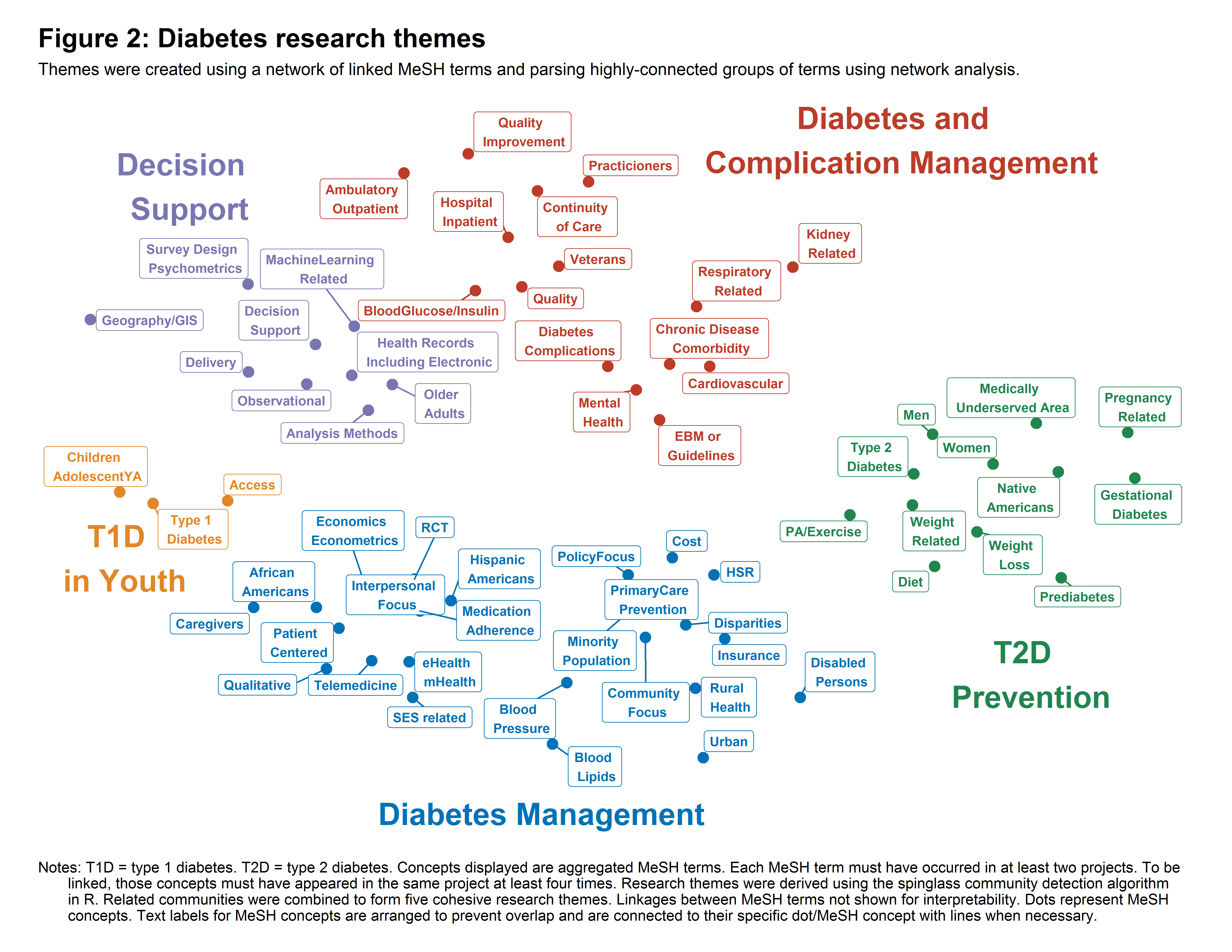
## Visualization

#set.seed(142378)  
 #layoutGG = layout\_with\_fr(meshGraphNew, weights=E(meshGraphNew)$UPweight)  
 #save(layoutGG, file="output/GGNETWORK\_LAYOUT.Rdata")  
  
# load the pre-specified layout  
 load("output/GGNETWORK\_LAYOUT.Rdata")  
   
 rownames(layoutGG) <- tmp1nodes$name  
 layoutGG["Disabled \n Persons", 1] <- 0.3  
 layoutGG["Disabled \n Persons", 2] <- -0  
   
 layoutGG["Urban", 1] <- 0  
   
 layoutGG["Type 1 \n Diabetes", 1] <- -1.7  
   
 layoutGG["Blood \n Lipids", 2] <- -0.2  
   
 layoutGG["SES related", 2] <- -0  
   
  
# and now we can visualize this network, using the layout we just pulled in  
set.seed(1234)  
base <- ggraph(graph=meshGraphNew,   
 layout=layoutGG) +  
   
 # remove all axes and the legend  
 theme\_blank() + theme(legend.position = "none")+   
   
 # modify theme elemnts, font sizes etc.  
 theme(plot.title = element\_text(size=17, face="bold", hjust=0),   
 plot.subtitle = element\_text(size=11, hjust=0),   
 plot.caption = element\_text(hjust = 0),   
 plot.margin = margin(t = 0.25, r = 0.4, b = 0.25, l = 0.3, unit = "in")) +   
   
  
   
 # add lots of informative labels and titles  
 labs(title = "Figure 2: Diabetes research themes",   
 subtitle = "Themes were created using a network of linked MeSH terms and parsing highly-connected groups of terms using network analysis.   
 ",   
 caption = "Notes: T1D = type 1 diabetes. T2D = type 2 diabetes. Concepts displayed are aggregated MeSH terms. Each MeSH term must have occurred in at least two projects. To be   
 linked, those concepts must have appeared in the same project at least four times. Research themes were derived using the spinglass community detection algorithm   
 in R. Related communities were combined to form five cohesive research themes. Linkages between MeSH terms not shown for interpretability. Dots represent MeSH   
 concepts. Text labels for MeSH concepts are arranged to prevent overlap and are connected to their specific dot/MeSH concept with lines when necessary.")  
  
  
  
  
  
# 5 communities -----------------------------------------------------------  
  
nejm\_not\_LAZY = c("Diabetes and complication management" = "#BC3C29FF",  
 "Decision support" = "#7876B1FF",  
 "T2D management" = "#0072B5FF",  
 "T1D management" = "#E18727FF",  
 "T2D prevention" = "#20854EFF")  
  
base +   
   
 # and now add the network points  
 geom\_node\_point(aes(colour=hsrCommF), size=3) +   
   
 # label those points with the concept name  
 geom\_node\_label(aes(label=name,   
 colour=hsrCommF),   
 size=3,   
 repel=TRUE,   
 fontface="bold") +   
  
 # specify that we want to use the reordered colors specified above  
 scale\_color\_manual(values=nejm\_not\_LAZY) +  
 scale\_fill\_manual(values=nejm\_not\_LAZY) +   
  
  
 annotate("text", x=0.6, y=2.4, label="Diabetes and \n Complication Management",  
 color = "#BC3C29FF", size=7, fontface=2) +  
  
 annotate("text", x=-0.5, y=-0.5, label="Diabetes Management",  
 color = "#0072B5FF", size=7, fontface=2) +  
  
 annotate("text", x=-1.8, y=0.6, label="T1D \n in Youth",  
 color = "#E18727FF", size=7, fontface=2) +  
  
 annotate("text", x=-1.6, y=2.2, label="Decision \n Support",  
 color = "#7876B1FF", size=7, fontface=2) +  
  
 annotate("text", x=1, y=0.1, label="T2D \n Prevention",  
 color = "#20854EFF", size=7, fontface=2)

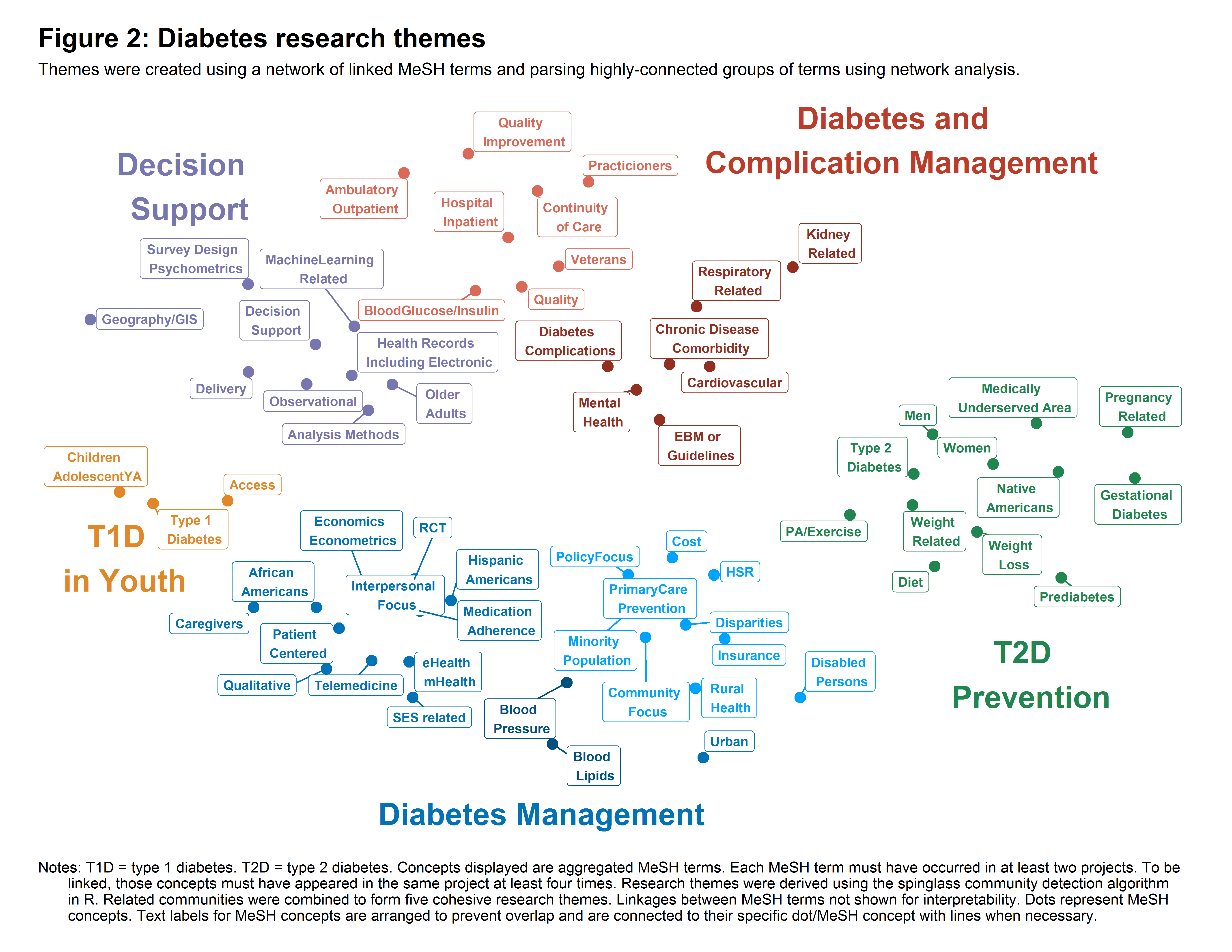
# note we are not graphing the edges because they look awful  
   
ggsave(filename = "output/Fig-2-5comms.png", plot=last\_plot(), device="png",   
 units="in", width=11, height=8.5, dpi=500)   
  
  
  
# 8 communities for some nuance -----------------------------------------------------------  
  
nejm\_not\_LAZY2 = c("Diabetes management in veterans" = "#da6958",  
 "Complications and comorbidity management" = "#922f20",  
   
   
 "T2D management 1" = "#0072B5FF",  
 "T2D management 2" = "#00a2ff",  
 "Lipids & pressure" = "#005180",  
   
 "Decision Support" = "#7876B1FF",  
 "T2D prevention" = "#20854EFF",  
 "T1D management" = "#E18727FF"  
   
 )  
  
base +   
   
 # and now add the network points  
 geom\_node\_point(aes(colour=commF), size=3) +   
   
 # label those points with the concept name  
 geom\_node\_label(aes(label=name,   
 colour=commF),   
 size=3,   
 repel=TRUE,   
 fontface="bold") +   
   
 # specify that we want to use the reordered colors specified above  
 scale\_color\_manual(values=nejm\_not\_LAZY2) +  
 scale\_fill\_manual(values=nejm\_not\_LAZY2) +   
   
   
 annotate("text", x=0.6, y=2.4, label="Diabetes and \n Complication Management",  
 color = "#BC3C29FF", size=7, fontface=2) +  
   
 annotate("text", x=-0.5, y=-0.5, label="Diabetes Management",  
 color = "#0072B5FF", size=7, fontface=2) +  
   
 annotate("text", x=-1.8, y=0.6, label="T1D \n in Youth",  
 color = "#E18727FF", size=7, fontface=2) +  
   
 annotate("text", x=-1.6, y=2.2, label="Decision \n Support",  
 color = "#7876B1FF", size=7, fontface=2) +  
   
 annotate("text", x=1, y=0.1, label="T2D \n Prevention",  
 color = "#20854EFF", size=7, fontface=2)

# note we are not graphing the edges because they look awful  
  
ggsave(filename = "output/Fig-2-nuance.png", plot=last\_plot(), device="png",   
 units="in", width=11, height=8.5, dpi=500)   
  
  
  
# remove everything from memory   
rm(list=objects())

include\_graphics("output/Fig-2-5comms.png")



include\_graphics("output/Fig-2-nuance.png")



# Appendix: function code

## Classify funders

FUNDERS = function(INPUT\_DS){  
   
 OUTPUT <-   
 INPUT\_DS %>%  
   
 mutate(funder = str\_extract\_all(SAList, "(?<=\\().+?(?=\\))"),   
 numFunders = lengths(funder),   
 NIH = map\_int(map(funder, str\_detect, pattern = "NIH"), any),   
 AHRQ = map\_int(map(funder, str\_detect, pattern = "AHRQ"), any),   
 RWJF = map\_int(map(funder, str\_detect, pattern = "RWJF"), any),   
 CDC = map\_int(map(funder, str\_detect, pattern = "CDC"), any),   
 PCORI = map\_int(map(funder, str\_detect, pattern = "PCORI"), any),   
 VA = map\_int(map(funder, str\_detect, pattern = "VA"), any),  
 HRSA = map\_int(map(funder, str\_detect, pattern = "HRSA"), any),  
 NIDILRR = map\_int(map(funder, str\_detect, pattern = "NIDILRR"), any),  
 RRF = map\_int(map(funder, str\_detect, pattern = "RRF"), any),  
 Commonwealth = map\_int(str\_detect(SAList, pattern="Commonwealth Fund"), any),  
 PharmAlliance = map\_int(str\_detect(SAList, pattern="PharmAlliance"), any),  
 Pfizer = map\_int(str\_detect(SAList, pattern="Pfizer"), any),  
 Gordon = map\_int(str\_detect(SAList, pattern="Gordon and Betty Moore Foundation"), any),  
 Aetna = map\_int(str\_detect(SAList, pattern="Aetna"), any),  
 ADA = map\_int(str\_detect(SAList, pattern="American Diabetes Association"), any),  
 CMS = map\_int(str\_detect(SAList, pattern="Centers for Medicare and Medicaid Services"), any),  
 USDA = map\_int(map(funder, str\_detect, pattern="USDA"), any),  
 DHHS = map\_int(map(funder, str\_detect, pattern="DHHS"), any),  
 NSF = map\_int(map(funder, str\_detect, pattern="NSF"), any),  
   
   
 # just leaving these as others  
 # NCDHHS = map\_int(str\_detect(SAList, pattern="North Carolina Department"), any),  
 # HSPHARM = map\_int(str\_detect(SAList, pattern="American Society of Health-System Pharmacists"), any),  
 # SAList = InHealth  
 # SAList=Florida Agency for Health Care  
 # SAList=Massachusetts Division of Health Care Finance  
 # Cincinnati Children's Hospital  
 # Academic Medicine and Managed Care Forum  
 # Parke Davis  
 # WTGrant  
 # PA Department of Health  
 # BCBS  
 # Office of Minority Health  
 # New Mexico Dept of Health  
 # United Hospital Fund  
 # AARP  
 # US Army  
 # California Health Care Foundation  
 # Center for Health Care Strategies  
 # GlaxoSmithKline  
 # Indian Health Service  
 # National Multiple Sclerosis Society  
 # Pew Charitable Trusts  
 # Kellogg Foundation   
   
 primaryFunder = case\_when(  
 NIH==1 ~ "NIH",  
 AHRQ==1 ~ "AHRQ",  
 RWJF==1 ~ "RWJF",  
 CDC==1 ~ "CDC",  
 PCORI==1 ~ "PCORI",  
 VA==1 ~ "VA",   
 HRSA==1 ~ "HRSA",   
 NIDILRR==1 ~ "NIDILRR",   
 RRF==1 ~ "RRF",   
 Commonwealth==1 ~ "CFUND",  
 PharmAlliance==1 ~ "PHARM",  
 Pfizer==1 ~ "PFIZER",   
 Gordon==1 ~ "GBMOORE",   
 Aetna==1 ~ "AETNA",  
 ADA==1 ~ "ADA",   
 CMS==1 ~ "CMS",  
 USDA==1 ~ "USDA",  
 DHHS==1 ~ "DHHS",  
 NSF==1 ~ "NSF",  
 TRUE ~ "OTHER"  
 ),   
   
 NIHspecific = case\_when(  
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "^NIMH$"), any) ~ "NIMH",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "^NIMHD$"), any) ~ "NIMHD",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NHLBI"), any) ~ "NHLBI",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NIDDK"), any) ~ "NIDDK",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NIA"), any) ~ "NIA",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NCI"), any) ~ "NCI",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NINR"), any) ~ "NINR",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NIEHS"), any) ~ "NIEHS",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NEI"), any) ~ "NEI",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NIDA"), any) ~ "NIDA",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NICHD"), any) ~ "NICHD",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NIDCR"), any) ~ "NIDCR",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NLM"), any) ~ "NLM",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NCATS"), any) ~ "NCATS",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "OD"), any) ~ "NIH-OD",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NHGRI"), any) ~ "NHGRI",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NCCIH"), any) ~ "NCCIH",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NINDS"), any) ~ "NINDS",   
 primaryFunder=="NIH" & map\_int(map(funder, str\_detect, pattern = "NCRR"), any) ~ "NCRR",   
 TRUE ~ "NotSpecifiedNotNIH"  
 )  
 ) %>%  
 select(-funder)  
   
 return(OUTPUT)  
   
 }

## Classify if project is research

TYPES = function(INPUT\_DS){  
   
 OUTPUT <-   
 INPUT\_DS %>%  
 mutate(grantFirst = str\_sub(GrantID, 1, 1),   
 grantType = str\_sub(GrantID, 1, 3),   
 isResearch = if\_else(# These are the combinations of funders and grant numbers that qualify as research  
 # R and K level grants from federal agencies  
 primaryFunder %in% c("NIH", "AHRQ", "HRSA") & grantFirst %in% c("R", "K") |   
   
 # U level grants from the CDC  
 primaryFunder=="CDC" & grantFirst %in% c("U") |  
   
 # U level grants from the NIH  
 # I DON'T UNDERSTAND U LEVEL GRANTS?   
 primaryFunder=="NIH" & grantType %in% c("U01", "UH3", "U54", "UC4",   
 "F30", "F31", "F32") |   
   
 # Comparative-effectiveness grants from PCORI  
 primaryFunder=="PCORI" & grantType %in% c("CE-", "CER", "IH-",   
 "IHS", "ME-", "AD-") |   
   
 # VA intramural research grants  
 primaryFunder=="VA" & grantType %in% c("I01", "IIR", "IK2",   
 "PPO", "RRP", "SDP", "SDR", "VCA") |   
   
 # Some other AHRQ grants  
 primaryFunder=="AHRQ" & grantType %in% c("IM1", "CRD") |   
   
 # Adding from manual review  
 # all of the titles/abstracts look like some sort of research to me   
 primaryFunder %in% c("AETNA", "GBMOORE", "NIDILRR", "PFIZER", "RRF")   
   
 # Things that are not research:   
 # commonwealth fund Israeli grant thing  
 # the HRSA center grants  
 # PCORI center grants - CDRN - clinical data research networks. Seems to fund infrastructure  
 # NIH UL1  
 # PHARM grant -- alliance of pharmacy schools  
 # RWJF grants all look like training  
 ,   
   
   
 # if a grant meets any of those conditions is a research grant  
 true = 1,   
   
 # and if not, then it is not research   
 false = 0),  
   
   
 ## fixing mis-classifications as a result of programmatically assigning research status.  
 # these were found during the manual review of the abstracts for SEM classification  
 # 20154193 is an R13 project funded by AHRQ for a conference  
 # 20162180 is an R13 project funded by NIH/NHLBI for SBM annual meeting  
 # 20174271 is a KL2 center project   
 isResearch = if\_else(ProjectID %in% c("20154193", "20162180", "20174271"),   
 true=0,   
 false=isResearch))  
   
   
  
 return(OUTPUT)  
   
}  
  
  
#write.csv(tmp1, file="output/isGrantResearch.csv", row.names = FALSE)

## Clean funding amount

FUNDINGAMT = function(INPUT\_DS){  
   
  
 OUTPUT\_DS <- INPUT\_DS %>%  
 #select(ProjectID, TotalFundingAmount, Fundings, primaryFunder, lengthProj) %>%  
   
 mutate(Fundings = if\_else(!is.na(TotalFundingAmount), NA\_character\_, Fundings)) %>%  
   
 # mean if the total funding is given directly  
 mutate(yearlyFunding1 = as.numeric(TotalFundingAmount) / lengthProj,  
   
   
 # for those who don't have a total funding amount listed, we need to extract the yearly amounts  
 # remove numbers we don't care about - the years  
 # and commas, which will make numbers>999 look like multiple numbers to R  
 # then take the mean to get average yearly funding   
 yearlyFunding2 = str\_extract\_all(str\_remove\_all(Fundings,   
 paste(c("\\,", "2012", "2013",   
 "2014", "2015", "2016",   
 "2017", "2018", "2019"),   
 collapse = "|")),   
 "\\d+"),   
 avgFunding = map\_dbl(yearlyFunding2, function(x) mean(as.numeric(x))),   
   
 avgYearlyFunding = case\_when(  
 !is.na(yearlyFunding1) ~ yearlyFunding1,   
 !is.na(avgFunding) ~ avgFunding,  
 ProjectID==20182042 ~ -1,   
 ProjectID==20141381 ~ 749473,   
 TRUE ~ as.numeric(NA)  
 )  
 ) %>%  
 select(-yearlyFunding1, -yearlyFunding2, -avgFunding)  
  
 return(OUTPUT\_DS)  
  
}

## Classify MeSH terms into broad concepts

# Complications -----------------------------------------------------------  
  
RESPIRATORY <- c("Asthma",   
 "Pulmonary Disease, Chronic Obstructive",   
 "Respiration Disorders",   
 "Smoking",   
 "Pneumonia")  
  
KIDNEY <- c("Renal Insufficiency, Chronic",   
 "Glomerular Filtration Rate",   
 "Kidney Failure, Chronic",   
 "Chronic Kidney Failure")  
  
  
  
# Chronic Diseases ---------------------------------------------------------------  
  
CHRONCOMORBID <- c("Comorbidity",   
 "Chronic Disease",   
 "Disease Management",   
 "Disease Progression")  
  
CARDIOV <- c("Myocardial Infarction",   
 "Heart Failure",   
 "Coronary Artery Disease",   
 "Coronary Disease",   
 "Stroke",   
 "Cardiovascular Diseases",   
 "Cardiology",   
 "Ischemia")  
  
BP <- c("Blood Pressure",   
 "Hypertension")  
  
BLOODGLUCOSE <- c("Metformin",   
 "Hyperglycemia",   
 "Hypoglycemia",   
 "Hypoglycemic Agents",   
 "Glycated Hemoglobin A",   
 "Hemoglobins",   
 "Thiazolidinediones",   
 "Blood Glucose",   
 "Blood Glucose Self-Monitoring",   
 "Glucose Tolerance Test",   
 "Insulin")  
  
BLOODLIPIDS <- c("Dyslipidemias",   
 "Lipids",   
 "Hyperlipidemias",   
 "Hypercholesterolemia")  
  
  
  
HIV <- c("HIV Infections")  
SLEEP <- c("Sleep")  
OSTEO <- c("Osteoarthritis")  
  
ALZ <- c("Alzheimers Disease",   
 "Alzheimer's Disease",   
 "Cognitive Dysfunction",   
 "Alzheimer Disease")  
  
  
# Risk Factors ------------------------------------------------------------  
  
WEIGHT <- c("Body Mass Index",   
 "Body Composition",   
 "Body Weight",   
 "Obesity",   
 "Overweight",   
 "Waist Circumference",   
 "Anthropometry")  
  
WEIGHTLOSS <- c("Weight Reduction Programs",   
 "Weight Loss")  
  
DIET <- c("Diet")  
  
EXERCISE <- c("Acceleration",   
 "Exercise",   
 "Exercise Therapy",   
 "Sedentary Behavior",   
 "Accelerometry")  
  
MENTALHEALTH <- c("Depression",   
 "Mental Disorders",   
 "Mental Health Services",   
 "Depressive Disorder, Major",   
 "Stress, Psychological",   
 "Antipsychotic Agents")  
  
  
  
  
# Populations -------------------------------------------------------------  
  
VETERANS <- c("Veterans")  
  
FEMALE <- c("Female", "Mothers")  
  
MALE <- c("Male")  
  
CARE <- c("Caregivers")  
  
DISABLE <- c("Disabled Persons")  
  
AGED <- c("Aged",   
 "Aging",   
 "Aged, 80 and over",   
 "Age Factors",   
 "Middle Aged",   
 "Geriatrics",   
 "Health Services for the Aged",   
 "Adult")  
  
URBAN <- c("Urban Population")  
  
RURAL <- c("Rural Population",   
 "Rural Health Services")  
  
YOUNG <- c("Adolescent Medicine",   
 "Adolescent",   
 "Young Adult",   
 "Child, Preschool",   
 "Child",   
 "Pediatrics")  
  
  
MINORITY <- c("Vulnerable Populations",   
 "Minority Groups",   
 "Ethnic Groups",  
 "Emigrants and Immigrants")  
  
HISPANIC <- c("Hispanic Americans",   
 "Mexican Americans")  
  
AA <- c("African Americans")  
  
AIAN <- c("Indians, North American")  
  
ASIAN <- c("Asian Americans")  
  
  
  
  
# Topics ------------------------------------------------------------------  
  
HSR <- c("Health Services Research")  
  
QI <- c("Quality Improvement")  
  
PROGRAMS <- c("Program Development",   
 "Pilot Projects",   
 "Program Evaluation")  
  
EBM <- c("Evidence-Based Medicine",   
 "Guidelines as Topic")  
  
ACCESS <- c("Health Services Accessibility",   
 "Access to Information",   
 "Access to Health Care")  
  
QUALITY <- c("Quality of Life",   
 "Quality of Health Care",  
 "Quality Assurance, Health Care")  
  
DELIVERY <- c("Delivery of Health Care, Integrated",   
 "Delivery of Health Care")  
  
COST <- c("Health Care Costs",   
 "Cost Sharing",   
 "Drug Costs")  
  
PATIENT <- c("Patient-Centered Care",   
 "Patient Participation",  
 "Professional-Patient Relations",  
 "Patient Satisfaction",   
 "Patient Preference",  
 "Patient Education as Topic",   
 "Patient Navigation")  
  
CONTINUITY <- c("Continuity of Patient Care",  
 "Patient Care Team")  
  
MEDADHERE <- c("Medication Adherence",   
 "Patient Compliance")  
  
DISPARITIES <- c("Healthcare Disparities",   
 "Health Status Disparities",   
 "Health Equity",   
 "Health Care Disparities")  
  
DECISION <- c("Decision Making",   
 "Decision Support Systems, Clinical",   
 "Decision Support Techniques")  
  
HEALTHCOMMPROM <- c("Communication",   
 "Health Communication",   
 "Health Promotion",   
 "Public Health",   
 "Health Education")  
  
PCAREPREVENTION <- c("Primary Health Care",   
 "Preventive Medicine",   
 "Primary Prevention")  
  
INSURANCE <- c("Insurance, Health",   
 "Medicare",   
 "Medicaid",   
 "Patient Protection and Affordable Care Act",   
 "Accountable Care Organizations",   
 "Deductibles and Coinsurance",   
 "Health Insurance",   
 "Affordable Care Act")  
  
EHR <- c("Electronic Health Records",   
 "Medical Records Systems, Computerized",   
 "Health Records, Personal",   
 "Medical Records")  
  
ELECTRONIC <- c("Cell Phone",   
 "Internet",   
 "Text Messaging",   
 "Reminder Systems",   
 "Wireless Technology")  
  
OUTCOMES <- c("Outcome Assessment, Health Care",   
 "Outcome and Process Assessment, Health Care",   
 "Treatment Outcome")  
  
SES <- c("Poverty",   
 "Social Class",   
 "Unemployment")  
  
SDOH <- c("Social Determinants of Health")  
  
HOSPITALS <- c("Hospitalization",   
 "Emergency Service, Hospital",   
 "Hospitals",   
 "Length of Stay",   
 "Inpatients",   
 "Hospitals, Veterans",   
 "Patient Admission",   
 "Patient Discharge",   
 "Patient Safety",   
 "Patient Readmission")  
  
AMBULATORY <- c("Ambulatory Care",   
 "Monitoring, Ambulatory",   
 "Outpatients")  
  
GIS <- c("Geography",   
 "Geographic Information Systems")  
  
PRACTICIONERS <- c("Nurse Practitioners",   
 "Nurses",   
 "Occupational Health Services",   
 "Physician Assistants",   
 "Physicians",   
 "Pharmacy")  
  
  
PREGNANCY <- c("Postpartum Period",   
 "Pregnancy")  
  
  
  
MUA <- c("Medically Underserved Area")  
  
  
  
  
  
  
# Methods Related ---------------------------------------------------------  
  
RCT <- c("Clinical Trials as Topic",   
 "Randomized Controlled Trials as Topic",   
 "Multicenter Studies as Topic")  
  
OBSERVATIONAL <- c("Observational Studies as Topic",   
 "Cohort Studies",   
 "Longitudinal Studies",   
 "Retrospective Studies",   
 "Prospective Studies",   
 "Cross-Sectional Studies",   
 "Registries",   
 "Insurance Claim Review",   
 "Medical Informatics",   
 "Databases, Factual")  
  
ECONOMICSMETRICS <- c("Models, Econometric",   
 "Models, Economic",   
 "Comparative Effectiveness Research",   
 "Economics, Medical",   
 "Cost-Benefit Analysis")  
  
MACHINELEARNINGRELATED <- c("Algorithms",   
 "Software",   
 "Natural Language Processing",   
 "Precision Medicine",   
 "Machine Learning")  
  
QUALITATIVE <- c("Motivational Interviewing",   
 "Focus Groups")  
  
SURVEYPSYCHO <- c("Psychometrics",   
 "Surveys and Questionnaires",   
 "Health Surveys",   
 "Reproducibility of Results",   
 "Validation Studies as Topic")  
  
ANALYSIS <- c("Computer Simulation",   
 "Data Interpretation, Statistical",   
 "Linear Models",   
 "Regression Analysis",   
 "Models, Statistical",   
 "Proportional Hazards Models",   
 "Multivariate Analysis",   
 "Cluster Analysis",   
 "Data Collection")  
  
  
  
INDIVIDUAL <- c("Self Care",   
 "Motivation",   
 "Attitude to Health",   
 "Self Care",   
 "Health Literacy",   
 "Health Knowledge, Attitudes, Practice",   
 "Self Efficacy",   
 "Health Behavior",   
 "Life Style")  
  
INTERPERSONAL <- c("Cultural Characteristics",   
 "Social Support",   
 "Family Health",   
 "Peer Group",   
 "Group Processes",   
 "Physician-Patient Relations",   
 "Family")  
  
COMMUNITY <- c("Models, Organizational",   
 "Community Health Workers",   
 "Community Health Services",   
 "Community-Institutional Relations",   
 "Community-Based Participatory Research")  
  
POLICY <- c("Health Policy",   
 "Health Care Reform")  
  
  
  
  
  
# Funders -----------------------------------------------------------------  
  
NIDDK <- c("National Institute of Diabetes and Digestive and Kidney Diseases (U.S.)")  
  
AHRQ <- c("United States Agency for Healthcare Research and Quality")  
  
CMS <- c("Centers for Medicare and Medicaid Services, U.S.")  
  
CDC <- c("Centers for Disease Control and Prevention, U.S.")  
  
WHO <- c("World Health Organization")  
  
VA <- c("United States Department of Veterans Affairs")  
  
  
  
# Physical Locations ------------------------------------------------------  
  
# will be dropped in the network, just putting all the locations in here for now   
GEOGRAPHY <- c("United States", "California", "Texas", "New York City", "Massachusetts", "Michigan", "New Mexico",   
 "District of Columbia", "North Carolina", "Washington", "Wisconsin", "Arizona", "Arkansas", "Alabama",   
 "Alaska", "Baltimore", "Cambodia", "California /epidemiology", "Pennsylvania", "Chile", "China", "Caribbean Region",   
 "United States Virgin Islands", "Virginia", "Chicago", "Louisiana", "Maryland", "Mexico", "Micronesia",   
 "Midwestern United States", "Minnesota", "Missouri", "Vietnam", "Connecticut", "India")