Joint protocol, Rare Disease Research (RDR)

Definition	All types of research funded by MRC or NIHR with relevance to a rare disease, defined as a disease with prevalence of less than 1 in 2000 individuals (definition commonly used in UK and EU). This definition includes diseases of childhood or adult onset, and all causes, as well as ultra-rare diseases and 'syndromes without a name' (SWANs). Note that there is a US definition of rare diseases based on an absolute number (i.e. a disease affecting fewer than 200k people in the US), but a definition based on prevalence is widely adopted in UK and EU, and has several advantages. Orphanet will be used as the starting point for a listing of individual rare diseases, with some correction for synonyms and alternative spellings (see below and detail in annexes). Rare disease research includes: ● Awards on specific rare diseases, which may or may not also identify the disease as 'rare' ● Awards on rare disease *per se*, for example infrastructure that serves all or multiple rare diseases. These awards may or may not mention any individual rare diseases. All types of research (in the broadest sense), including (non-exclusive list): ● basic and discovery science; cell and animal models; experimental medicine in rare diseases (e.g. mechanistic studies in human participants); functional genomics ● preventative and treatment interventions, identification and screening programmes, diagnostic testing, identification of clinical thresholds and care pathways for rare diseases, data science and bioinformatics ● methodology, research design ● Health and care services, organization and workforce issues, including leadership and training ● Infrastructure <u>Disease groupings</u> • All types of diseases are included, e.g. those with genetic or non-genetic origins, and those arising from e.g. infectious, allergic or environmental causes

	• The analysis may need to consider specific groups separately but should not exclude them entirely from the start. For consideration of disease groupings, Orphanet provides a classification of rare diseases into 30 disease groupings (see **Annex 1**). If required, this can be used as a pragmatic way to sub-group e.g. rare neoplastic or rare infectious diseases.
Search strategy	Purpose The primary aim for this protocol is to provide a snapshot of MRC and NIHR funding in this research area for a fixed time period (allowing comparisons to other portfolios with a standard e.g. five-year timeframe). Strategy Due to the number of individual rare diseases, MRC and NIHR will first identify all potentially relevant awards in the selected time period and combine into a joint list (not restricted to rare disease research at this stage). MRC will then run a script to tag each award with 1) any individual rare disease name that is included in the title, abstract or lay summary and 2) any of the search terms for rare disease *per se*. Data fields[1] *Input, list of funded awards (i.e. directly funded research awards) or supported studies (i.e. infrastructure support) for a selected time period* The main data fields will be: ● Funding body i.e. MRC or NIHR ● Funder reference ● Funding programme/NIHR Infrastructure centre/scheme etc. ● Award/Study Title (note, for projects supported by NIHR infrastructure managed by CCF i.e. BRCs, ARCs etc., only titles will be searched, and results will be reported separately) ● Abstract/Technical summary ● Research Summary (MRC: Summary) ● Call text (for NIHR funded awards managed by NETSCC only) ● Research organisation/Contractor ● Award start and end dates ● Awarded amounts (total awarded) ● HRCS coding, separate columns for health category and research activity, multiple values delimited by semicolons *Output (generated by algorithm that MRC will run), list of awards annotated with disease names or search terms* As above plus: ● Additional column listing disease names matched to each award, delimited

[1] Please note, there may be differences in the available data fields and terminology used when looking at data managed by more than one funder and where funders may support more than one type of research activity.

	● Additional column listing search terms matched to each award, delimited ● Columns with all terms matching the record and counts (number of terms matching) Due to the differences in funding process, and therefore interpretation, separate output lists are likely for 1) funded awards (MRC research projects etc., NIHR programmes), 2) projects supported by NIHR infrastructure (e.g. CCF) and 3) projects supported by NIHR CRN.

Data Sources	<u>MRC</u>
	Will include:
	● Research awards, partnerships, fellowships, unit programmes etc.
	● Includes all awards listed on:
	  ○ Siebel
	  ○ LIS – Large Investments System
	
	Will not include:
	● PhD studentships as these records have variable levels of information available on MRC systems (e.g. abstracts are not always available and titles may be nonspecific). Scientific plans for studentships are not centrally peer reviewed and are often awarded through Doctoral Training Programme awards with peer review devolved to the research organisation.
	
	<u>NIHR</u>
	Will include:
	● NIHR Programmes i.e., directly funded research including career development awards
	● NIHR Infrastructure (i.e., supported research managed by CCF)
	● NIHR Clinical Research Network (CRN)
	
	Will not include:
	  ○ ESP Infrastructure (Cochrane Groups)
	  ○ NIHR Programmes Global Health Research i.e., ODA funded research
	  ▪ Global Health Policy and Systems Research (GHPSR) - managed by NETSCC
	  ▪ Global Health Research Units and Groups (GHRUG) - managed by NETSCC
	  ▪ Research and Innovation for Global Health Transformation (RIGHT) - managed by CCF
	  ▪ NIHR Global Research Professorship - managed by NIHR Academy
	  ● Research Professorships (where funder recorded as NIHR (ODA) - managed by NIHR Academy)

Inclusions	<u>Time period</u>

	● The proposed range is all awards active on or after 1st April 2016, and five years after that date. ● This will provide a five year 'snapshot' comparable to other reports that are made on a five year basis.
Exclusions	Orphanet was adopted as the default list of rare diseases/conditions, and therefore no disease/condition that is listed in Orphanet was excluded.
Keywords	The search will use three lists of search terms: 1. Individual rare disease names, based on Orphanet (**Annex 1**, and csv file listing in **Annex 2**) 2. Search terms for rare disease(s) *per se* (csv file listing in **Annex 3**) 3. Limited set of manually identified search terms (**Annex 4**).
Comments	<u>Approach, comparability to wider 'scoping' searches</u> Due to the number of rare diseases the use of a broad scope to these searches will provide an indication of diseases where research is not being funded as well as identifying research that has been funded. The search can be fully documented (including any exceptions or customizations of Orphanet) and therefore will be able to be re-run in the future and by other stakeholders <u>Data Sharing</u> Much, but not all, of the MRC and NIHR internal data is in the public domain (project titles, abstracts etc.). MRC award data is published on the UKRI Gateway to Research website. Orphanet data is licensed as CC BY 4.0. <u>Implications for wider project</u> The practical difficulties in defining a portfolio could be identified as an outcome, e.g. it would potentially benefit the field if better methodologies, lists of diseases and approaches to identifying rare disease research could be developed. In addition, rare disease research is hard to define because of a lack of mechanistic and molecular definitions and classification schemes, as a result clustering or grouping of diseases is usually by somewhat arbitrary criteria (main phenotype, organ system affected, clinical specialty etc.)
References	● UK Rare Disease Framework 2021 ● England Action Plan 2022 ● MRC PSMB webpage and rare disease board opportunity definition ● Orphanet ○ Website: https://www.orpha.net/consor/cgi-bin/index.php ○ Data portal: http://www.orphadata.org/cgi-bin/index.php

	○ Orphanet documentation: https://www.orpha.net/orphacom/cahiers/docs/GB/eproc_disease_inventory_R1_Nom_Dis_EP_04.pdf ● Gateway to Research (GtR): https://gtr.ukri.org/ ● NIHR Open Data: ○ Dashboard: https://nihr.opendatasoft.com/pages/nihr-awards-filters/#-value-of-awards ○ Funded portfolio: https://nihr.opendatasoft.com/explore/dataset/infonihr-open-dataset/table/?disjunctive.funder&disjunctive.project_status&disjunctive.programme&disjunctive.programme_type&disjunctive.acronym
Subject experts consulted	● DHSC Rare Disease Steering Group, ● DHSC Rare Disease Advisory Group

ANNEXES

Annex 1: Individual rare disease names, based on Orphanet

Deriving a list of disease names from Orphanet

Orphanet is the de facto standard for rare diseases, provides data in usable and reusable formats/licenses, and provides links to external data sources including International Classification of Diseases 10th revision (ICD-10), Medical Subject Headings (MeSH, the controlled vocabulary thesaurus of the US National Library of Medicine) and Online Mendelian Inheritance in Man (OMIM, an online compendium of human genes and genetic phenotypes). However, Orphanet is not a keywords list, rather a list of disease names, with inconsistent usage and definitions (ranging from precisely understood monogenic disease through to syndromes)

It will not be feasible to cover all synonyms of diseases or variations more commonly used by researchers, the aim is a consistent, reproducible approach that can be fully described. Limited customizations will be identified manually and only used if essential to avoid clearly erroneous results (see **Annex 4**).

Starting from the Orphanet data file (see below, data file is updated monthly):

- Will select main disease names (ca. 7-8000)
- Then add synonyms (similar number to main names), but map these to the main disease names (so that the number of apparent diseases is not inflated)
- Handle following cases by treating the alternative form as an additional synonym:
    - UK-US spellings.
        - Define set of common alternative spellings and create synonyms.
        - Implement by copy, then search and replace? Adding a UK spelling that is not in fact used inflates the list of search terms but won't affect the results.
        - Cases to handle, often US□UK spelling. *Note that phrases can be parts of words or whole words*
            - aging □ ageing
            - celiac □ coeliac
            - edema □ oedema
            - emia, emic □ aemia, aemic
            - enia □ aenia, e.g. cypotenia
            - esophagus/esophageal/esophagitis □ oesphagus/oesphageal/oesophagitis
            - fetal/fetus □ foetal/foetus

- Fiber fibre
- Hemo haemo
- Leuko leuco
- Pediatric paediatric
- Seborrheic seborrheic [NORD, not in ORPHA]
- Tumor tumour

- Hyphens i.e. “Ehlers-Danlos” or “Ehlers Danlos” – 1) remove hyphen, 2) replace hyphen with space
- ‘Catalogue format’ or use of commas i.e. “Acrofacial dysostosis, Catania type” or “Catania type Acrofacial dysostosis”
  - Search for commas and convert “main, subtype” “subtype main” - use as potential synonym
  - Note, some commas could be a legitimate part of a disease name
- Use of slash (/) i.e. “Ehlers-Danlos/osteogenesis imperfecta syndrome” “Ehlers-Danlos” and “osteogenesis imperfecta syndrome”
- Use of ‘and’, e.g. “Erythropoietic Protoporphyria and X-Linked Protoporphyria”
  - Slash (/) and ‘and’ could imply linked disease, not either/or?
- Special characters like in “Behçet’s Syndrome”, “Björnstad Syndrome”, “Ménière’s Disease”
- Possessive apostrophe, Huntington and Huntington’s, also Huntingtons? - undecided
- Abbreviations - search as is, as the abbreviation may well be the common usage.

Data files

- Orphanet: Rare diseases and alignment with terminologies and databases (updated monthly).
- Latest version, **01 April 2022**: http://www.orphadata.org/data/xml/en_product1.xml
  - Excel import: using the Excel import wizard for XML files, and Excel-generated schema: 67 columns, 72717 rows
- *Note for linking: “The Rare diseases and alignment with terminologies and databases files includes all Orphanet clinical entities and their alignments with external terminologies or resources (ICD-10, OMIM, UMLS, MeSH, MedDRA and GARD). The alignments specify the comparability between terminologies by defining if the concepts are perfectly equivalent (exact mapping) or not. The dataset includes the ORPHAcode, preferred name and synonyms, a definition, the code or reference within the other terminologies or databases, and the qualified relationships. These files are updated once a month”.* (Orphanet website)

R search algorithm

- Read in 1) a list of awards and 2) a list of disease names and synonyms
- Join all target fields (title, summary, technical summary) in the list of awards
- For each main disease name/synonym:
    - search all target fields (with R grep command – Get Regular Expression Pattern) for an **exact match, case-insensitive, whole phrase only** (use word boundaries at start and end to avoid matching parts of longer phrases).
    - Append the terms causing the match to new columns, for 1) disease names, 2) search terms for rare disease per se and 3) any custom searches, 4) all terms i.e. 1+2+3.
    - Delimit multiple values to make secondary analysis easier.
    - If reporting counts (additional column) then increment count when a term is added.

See example script in **Annex 5**

Orphanet disease groupings

Notes

Orphanet provides 30 groupings based on medical specialties, listed in the table. Note that the groups are defined in different ways and vary in size.

Data files:

- Single linearized file, maps each disease to one grouping:
    - Web page: http://www.orphadata.org/cgi-bin/rare_free.html#linearmodal, updated monthly.
    - Version used: 01 April 2022: http://www.orphadata.org/data/xml/en_product7.xml
        - Excel import: using the Excel import wizard for XML files, and Excel-generated schema: 25 columns, 7232 rows
    - *Note: "The polyparental structure of the Orphanet classification of rare diseases implies that a clinical entity is included in as many classification hierarchies as necessary depending on its clinical presentation and the medical specialties to which it is relevant. In order to enable the sorting out of all rare diseases by medical specialty and avoid multiple counting of multiclassified entities in statistical analysis, a linearisation process is applied in the Orphanet database to attribute one medical specialty to each clinical entity. In order to ensure the consistency of this attribution, a set of rules have been established (see below procedure on Linearisation rules for Orphanet classifications). The dataset provided in this section includes all rare diseases present in the Orphanet nomenclature (ORPHAcode and preferred term in English), each with the medical specialty attributed as the preferential parent by Orphanet. It is updated once a month."* (Orphanet website)

Annex 2: csv file listing search phrases used for individual rare diseases

See csv file

Annex 3: csv file listing search phrases used for rare disease constructs

See csv file

Annex 4: Limited set of manually identified search terms

Selected search terms will be added manually only if the office teams or expert groups identify known gaps in the results due to the lists of search terms used. This list will only be used when essential so that the core basis for the search on Orphanet) is not compromised, and the search is fully documented, repeatable (by MRC, NIHR and others) and standardized as possible, without omitting important results.

Manual terms that were added are listed in the following table, and included within the individual disease search (Annex 1):

OrphaCode	MainDisease	SearchPhrase	Name/Synonym
60	Alpha-1-antitrypsin deficiency	alpha-1 antitrypsin deficiency	Synonym_CustomAddition
65	Leber congenital amaurosis	lebers congenital amaurosis	Synonym_CustomAddition
95	Friedreich ataxia	friedreich's ataxia	Synonym_CustomAddition
100	Ataxia-telangiectasia	ataxia telangiectasia	Synonym_CustomAddition
191	Cockayne syndrome	cockayne's syndrome	Synonym_CustomAddition
388	Hirschsprung disease	hirschsprung's disease	Synonym_CustomAddition
399	Huntington disease	huntington's disease	Synonym_CustomAddition
543	Burkitt lymphoma	burkitt's lymphoma	Synonym_CustomAddition
547	Non-Hodgkin lymphoma	non-hodgkin's lymphoma	Synonym_CustomAddition
549	Legionnaires disease	legionnaire's disease	Synonym_CustomAddition
846	Alpha-thalassemia	thalassaemia	Synonym_CustomAddition
846	Alpha-thalassemia	thalassemia	Synonym_CustomAddition
1942	Myoclonic-astatic epilepsy	myoclonic astatic epilepsy	Synonym_CustomAddition
2134	Atypical hemolytic uremic syndrome	atypical haemolytic uraemic syndrome	Synonym_CustomAddition
2134	Atypical hemolytic uremic syndrome	atypical haemolytic uremic syndrome	Synonym_CustomAddition
2134	Atypical hemolytic uremic syndrome	atypical hemolytic uraemic syndrome	Synonym_CustomAddition
2978	Chronic intestinal pseudoobstruction	chronic intestinal pseudo-obstruction	Synonym_CustomAddition
79278	Autosomal erythropoietic protoporphyria	erythropoietic protoporphyria	Synonym_CustomAddition
85138	Addison disease	addison's disease	Synonym_CustomAddition
85451	ATTRV122I amyloidosis	transthyretin amyloid cardiomyopathy	Synonym_CustomAddition
90038	Shiga toxin-associated hemolytic uremic syndrome	stec hus	Synonym_CustomAddition
96253	Cushing disease	cushing's disease	Synonym_CustomAddition
96253	Cushing disease	cushing's syndrome	Synonym_CustomAddition
98293	Hodgkin lymphoma	hodgkin's lymphoma	Synonym_CustomAddition

139417	Acute transverse myelitis	transverse myelitis	Synonym_CustomAddition
275555	Preeclampsia	pre eclampsia	Synonym_CustomAddition
275555	Preeclampsia	pre-eclampsia	Synonym_CustomAddition
275752	Sickle cell disease and related diseases	sickle cell	Synonym_CustomAddition
275752	Sickle cell disease and related diseases	sickle-cell	Synonym_CustomAddition
391673	Necrotizing enterocolitis	necrotising enterocolitis	Synonym_CustomAddition
UKRDRL-1	spinal muscular atrophy	spinal muscular atrophy	NameOfMainDisease_CustomAddition
UKRDRL-2	epidermolysis bullosa	epidermolysis bullosa	NameOfMainDisease_CustomAddition
UKRDRL-3	Charcot-Marie-Tooth	charcot marie tooth	Synonym_CustomAddition
UKRDRL-3	Charcot-Marie-Tooth	charcot-marie tooth	Synonym_CustomAddition
UKRDRL-3	Charcot-Marie-Tooth	charcot-marie-tooth	NameOfMainDisease_CustomAddition

Annex 5: sample R script

```
### USERS SHOULD VERIFY THAT THE CODE IS FIT FOR PURPOSE AND OPERATING CORRECTLY ON THE DATA SOURCE BEING USED BEFORE USING FOR DATA PRODUCTION

### R script for tagging portfolios against rare disease constructs ("CON") and individual rare disease names from Orphanet ("INDI").

### Code below expects separate data fields for award titles (TI) and combined title and abstracts (TIAB)

### Code: VERSION 2, started on 13-08-2022. This version runs searches against TI then TIAB, and does both INDI and CON; see loops 1-4 below.

### Paths etc are for use on a standard Win 10 laptop, with R installed. In the code below users will need to set the paths and variables
indicated by '---' before and after the descriptions given

### Data: Read in a table of applications('awards') and a list of searchPhrases for a) CON, CONstructs for rare diseases per se and b) INDI,
names/synonyms of INDIvidual rare disease


### create folder manually then set the working directory here for code output

setwd("---path to output directory---")

awards=read.csv("---filename and path for a csv file containing the awards data to be searched---", stringsAsFactors=FALSE) #list of awards

awardsName ="---set to a short but descriptive name to identify the results---" #match this to the data file for AWARDS specified in the line
above. it is used in the file name for the results

indi=read.csv("---file name and path for a csv file containing a list of search phrases for INDIvidual rare diseases---", stringsAsFactors=FALSE)
#list of rare disease names and synonyms, csv format

con=read.csv("---file name and path for a csv file containing a list of search phrases for rare disease CONstruct terms---",
stringsAsFactors=FALSE) #list of rare disease constructs, csv format


#the awards file should have columns header TI and TIAB, in order for the code below to work. If not, these columns must be generated here

#awards$TIAB = paste(awards$TI, awards$Research.Summary, sep=". ") # uncomment this line and tailor as needed


### FULL SEARCH OPTION,

# test on a subset of records if the full set is large. Depending on the computer used, each sub-search may take multiple seconds, and large
lists of awards/search phrases will take multiple hours to run through.

# iterate through awards titles or combined titles/abstracts with list of search phrases (either con or indi), and tag relevant rows for
downstream analysis (pivot tables etc)


# 1. CON TI

awards$CON_TI_RDcount =0

awards$CON_TI_RDlist=""
```

```
for (i in 1:length(con$SearchPhrase)){

        print (paste("CON TI: ", i, Sys.time(), con$SearchPhrase[i], sep=" "))

        grepResult = grep(paste("\\b", con$SearchPhrase[i], "\\b", sep=""), awards$TI, ignore.case=TRUE, perl=TRUE)

        val = length(grepResult)

        con$TI_COUNT[i] = val

        if (val > 0) {

                print (paste("CON TI: ", Sys.time(), " ", i, " of ", length(con$SearchPhrase), " >>> ", con$SearchPhrase[i], " COUNT: ", val,
sep=""))

                for (ii in 1:val) {

                        awards$CON_TI_RDcount[grepResult[ii]] = awards$CON_TI_RDcount[grepResult[ii]] + 1 # increment

                        if (awards$CON_TI_RDcount[grepResult[ii]]==1) {awards$CON_TI_RDlist[grepResult[ii]] = con$SearchPhrase[i]} #first time, so
initialise to this phrase

                        if (awards$CON_TI_RDcount[grepResult[ii]]>1) {awards$CON_TI_RDlist[grepResult[ii]] =
paste(awards$CON_TI_RDlist[grepResult[ii]], con$SearchPhrase[i], sep="|")} # not first RD for this app, so append RD to list

                }

        }

}


# 2. CON TIAB

awards$CON_TIAB_RDcount =0

awards$CON_TIAB_RDlist=""


for (i in 1:length(con$SearchPhrase)){

        print (paste("CON TIAB", i, Sys.time(), con$SearchPhrase[i], sep=" "))

        grepResult = grep(paste("\\b", con$SearchPhrase[i], "\\b", sep=""), awards$TIAB, ignore.case=TRUE, perl=TRUE)

        val = length(grepResult)

        con$TIAB_COUNT[i] = val

        if (val > 0) {

                print (paste("CON TIAB: ", Sys.time(), " ", i, " of ", length(con$SearchPhrase), " >>> ", con$SearchPhrase[i], " COUNT: ", val,
sep=""))
```

```
		for (ii in 1:val) {

			awards$CON_TIAB_RDcount[grepResult[ii]] = awards$CON_TIAB_RDcount[grepResult[ii]] + 1 # increment

			if (awards$CON_TIAB_RDcount[grepResult[ii]]==1) {awards$CON_TIAB_RDlist[grepResult[ii]] = con$SearchPhrase[i]} #first time, so initialise to this phrase

			if (awards$CON_TIAB_RDcount[grepResult[ii]]>1) {awards$CON_TIAB_RDlist[grepResult[ii]] = paste(awards$CON_TIAB_RDlist[grepResult[ii]], con$SearchPhrase[i], sep="|")} # not first RD for this app, so append RD to list

		}

	}

}
```

 # 3. INDI TI

```
awards$INDI_TI_RDcount =0

awards$INDI_TI_RDlist=""


for (i in 1:length(indi$SearchPhrase)){

	print (paste("INDI TI: ", i, Sys.time(), indi$SearchPhrase[i], sep=" "))

	grepResult = grep(paste("\\b", indi$SearchPhrase[i], "\\b", sep=""), awards$TI, ignore.case=TRUE, perl=TRUE)

	val = length(grepResult)

	indi$TI_COUNT[i] = val

	if (val > 0) {

		print (paste("INDI TI: ", Sys.time(), " ", i, " of ", length(indi$SearchPhrase), " >>> ", indi$SearchPhrase[i], " COUNT: ", val, sep=""))

		for (ii in 1:val) {

			awards$INDI_TI_RDcount[grepResult[ii]] = awards$INDI_TI_RDcount[grepResult[ii]] + 1 # increment

			if (awards$INDI_TI_RDcount[grepResult[ii]]==1) {awards$INDI_TI_RDlist[grepResult[ii]] = indi$SearchPhrase[i]} #first time, so initialise to this phrase

			if (awards$INDI_TI_RDcount[grepResult[ii]]>1) {awards$INDI_TI_RDlist[grepResult[ii]] = paste(awards$INDI_TI_RDlist[grepResult[ii]], indi$SearchPhrase[i], sep="|")} # not first RD for this app, so append RD to list

		}

	}

}
```

```
# 4. INDI TIAB

awards$INDI_TIAB_RDcount =0

awards$INDI_TIAB_RDlist=""


for (i in 1:length(indi$SearchPhrase)){

        print (paste("INDI TIAB: ", i, Sys.time(), indi$SearchPhrase[i], sep=" "))

        grepResult = grep(paste("\\b", indi$SearchPhrase[i], "\\b", sep=""), awards$TIAB, ignore.case=TRUE, perl=TRUE)

        val = length(grepResult)

        indi$TIAB_COUNT[i] = val

        if (val > 0) {

                print (paste("INDI TIAB: ", Sys.time(), " ", i, " of ", length(indi$SearchPhrase), " >>> ", indi$SearchPhrase[i], " COUNT: ", val,
sep=""))

                for (ii in 1:val) {

                        awards$INDI_TIAB_RDcount[grepResult[ii]] = awards$INDI_TIAB_RDcount[grepResult[ii]] + 1 # increment

                        if (awards$INDI_TIAB_RDcount[grepResult[ii]]==1) {awards$INDI_TIAB_RDlist[grepResult[ii]] = indi$SearchPhrase[i]} #first
time, so initialise to this phrase

                        if (awards$INDI_TIAB_RDcount[grepResult[ii]]>1) {awards$INDI_TIAB_RDlist[grepResult[ii]] =
paste(awards$INDI_TIAB_RDlist[grepResult[ii]], indi$SearchPhrase[i], sep="|")} # not first RD for this app, so append RD to list

                }

        }

}
```