

NGI: RNAseq

Processing RNA-seq data at the
National Genomics Infrastructure

SciLifeLab



NGI stockholm

Phil Ewels
phil.ewels@scilifelab.se
NBIS RNA-seq tutorial
Umeå, 2018-11-14

— SciLifeLab NGI



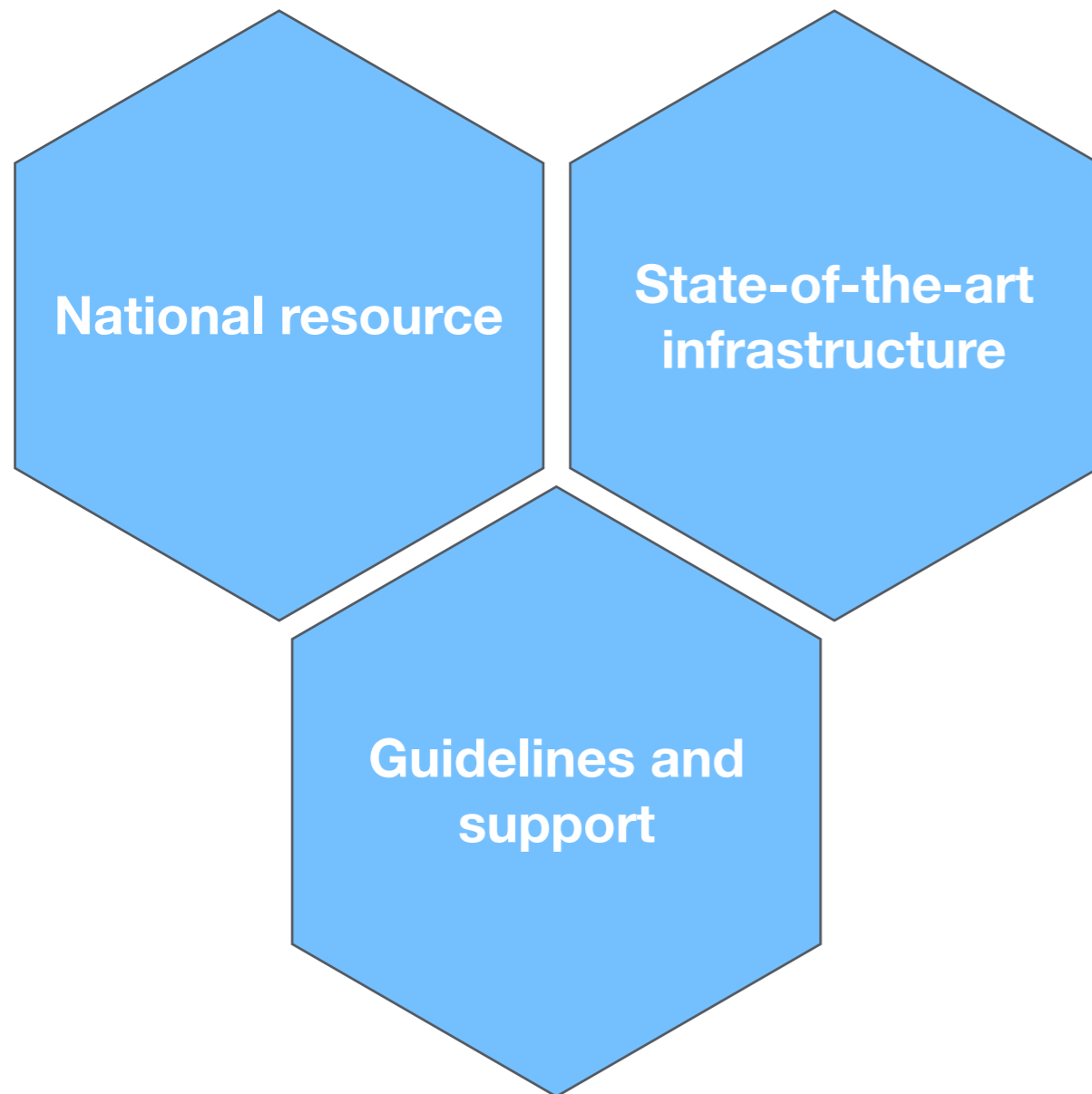
 **NATIONAL CTAC**
ATC GENOMICSGT
INFRASTRUCTURE

Our mission is to offer a **state-of-the-art infrastructure** for massively parallel DNA sequencing and SNP genotyping, available to researchers all over Sweden

SciLifeLab

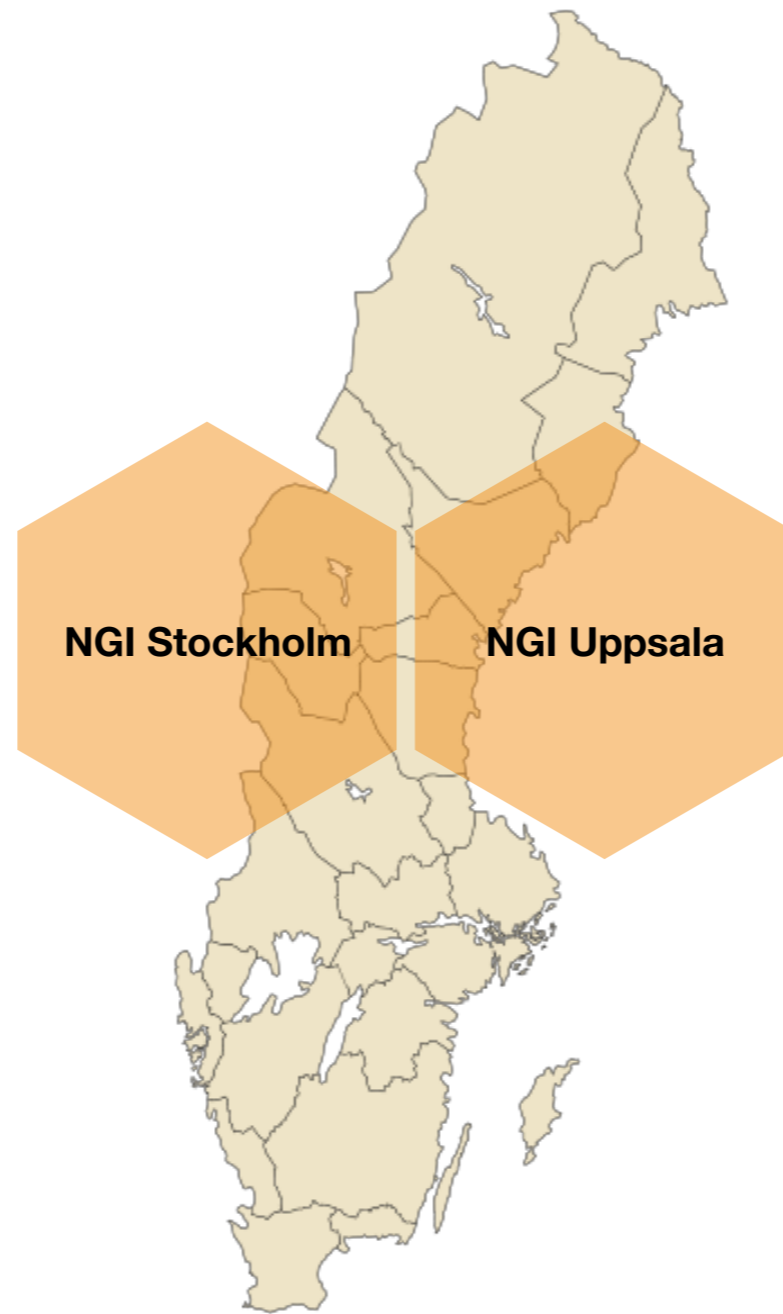
 **NGI** stockholm

SciLifeLab NGI



We provide
guidelines and support
for sample collection, study
design, protocol selection and
bioinformatics analysis

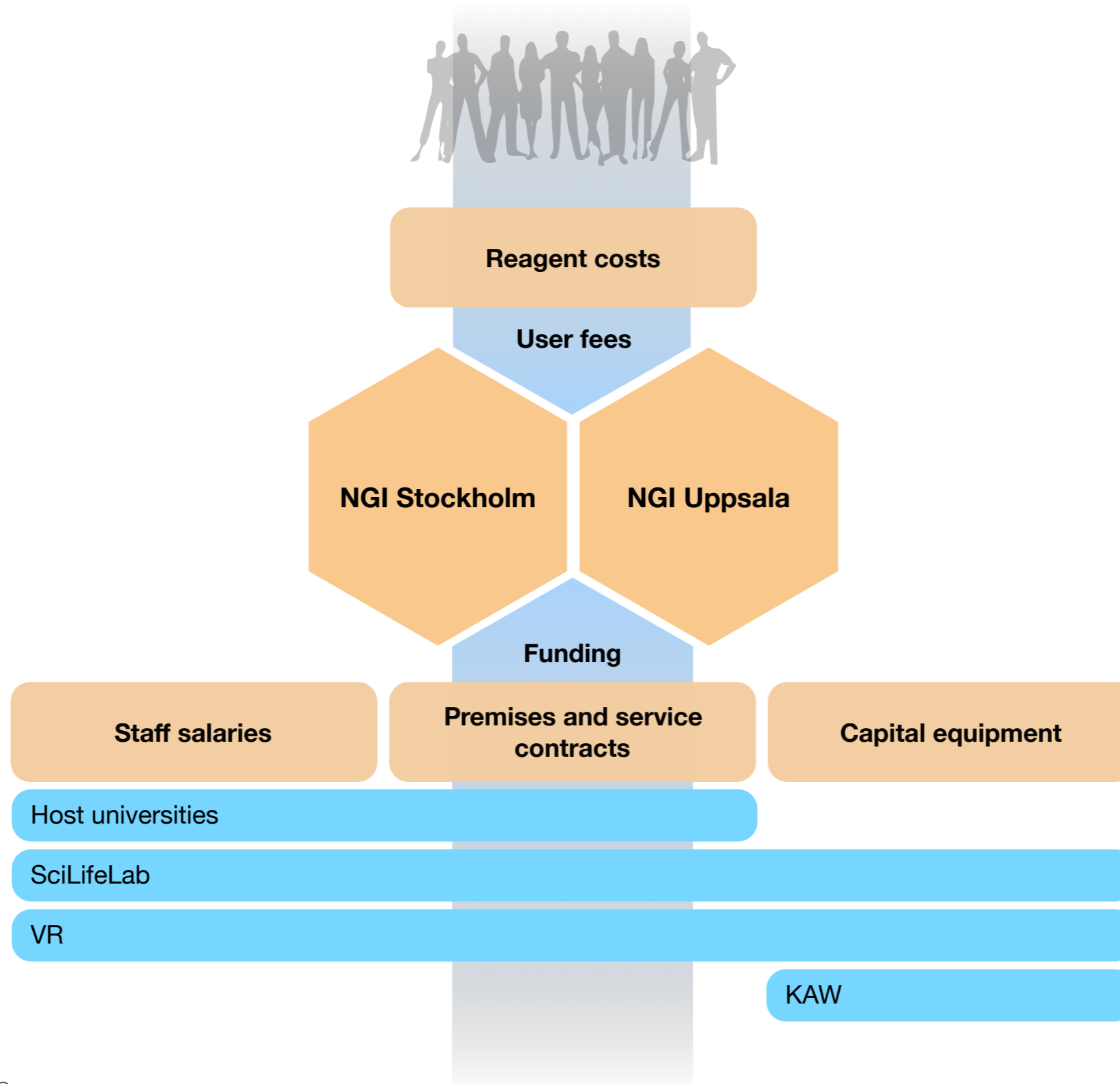
— NGI Organisation



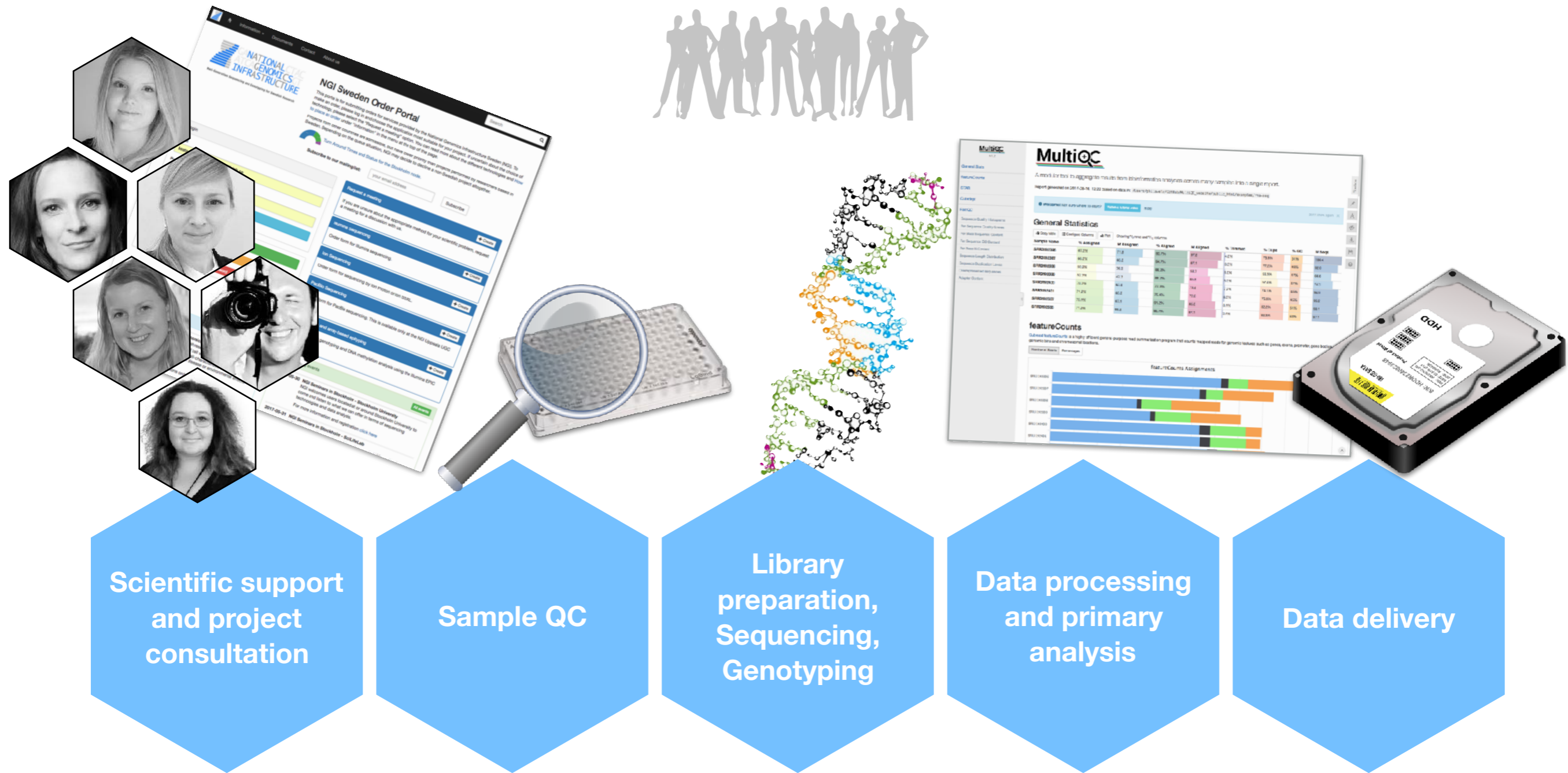
SciLifeLab

 NGI stockholm

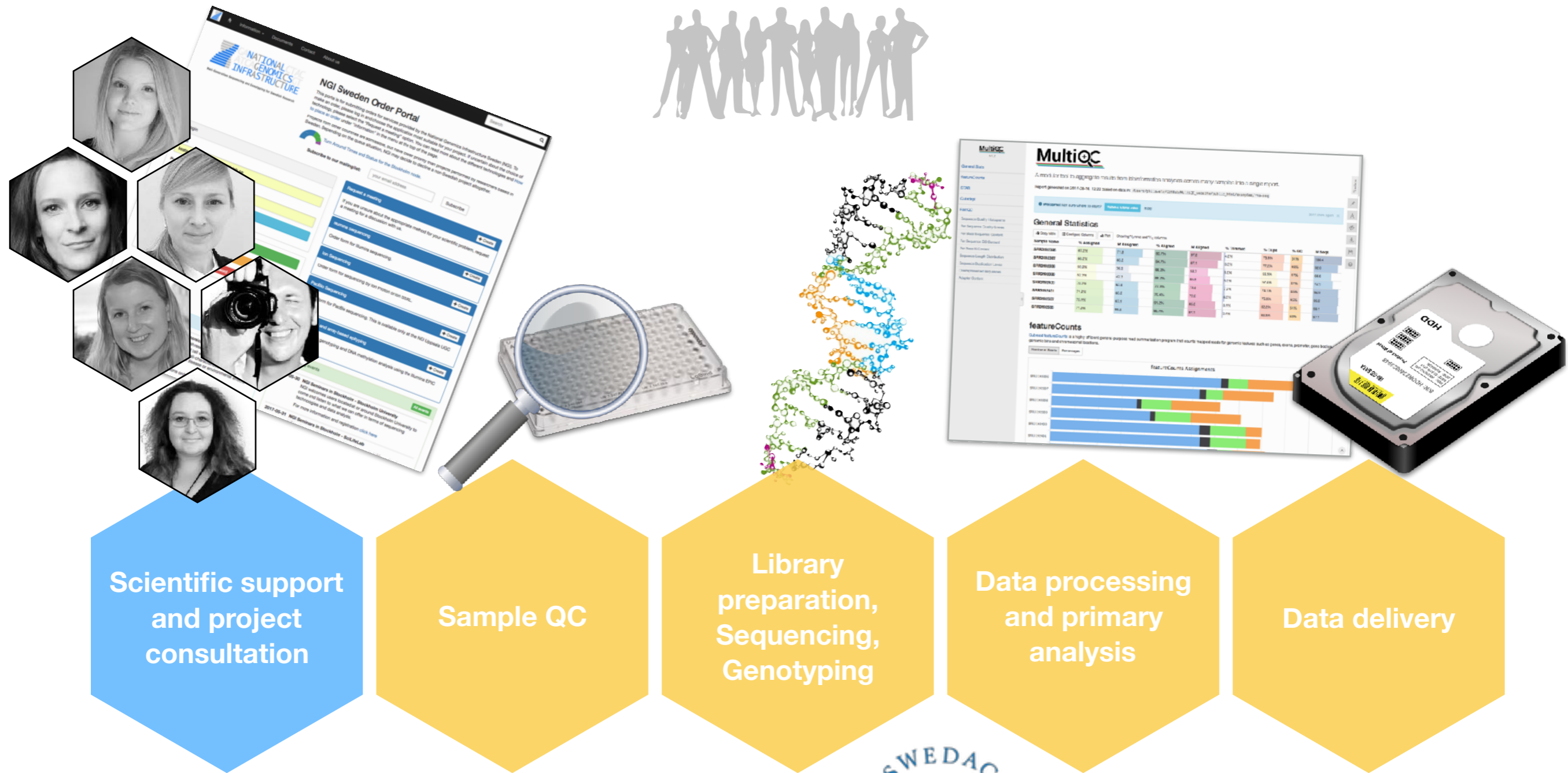
NGI Organisation



Project timeline



Project timeline

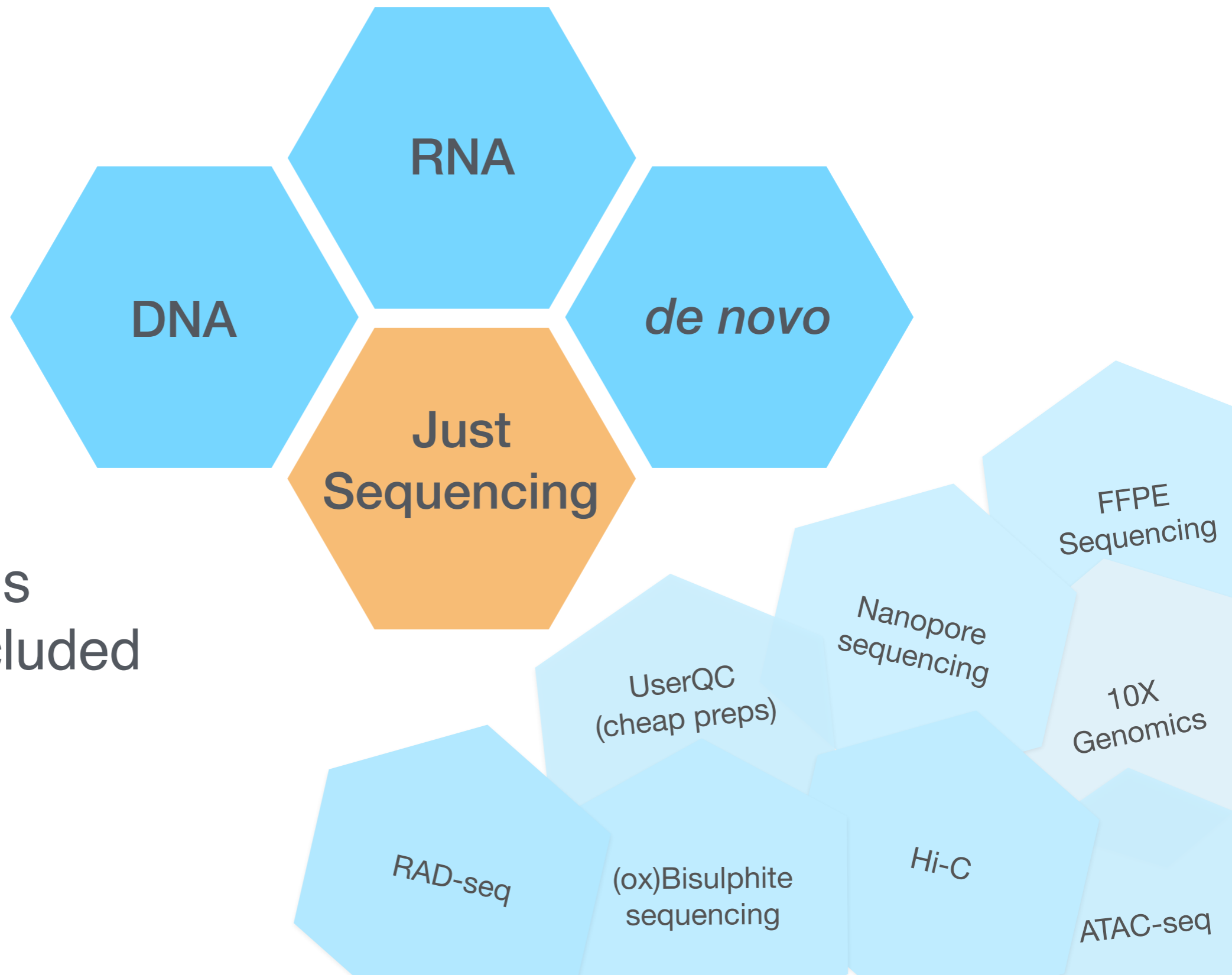


SciLifeLab

NGI stockholm

ACKREDITERING.SWEDAC.SWEDEN
Ackred. nr 1850
Provning
ISO/IEC 17025

Methods offered at NGI



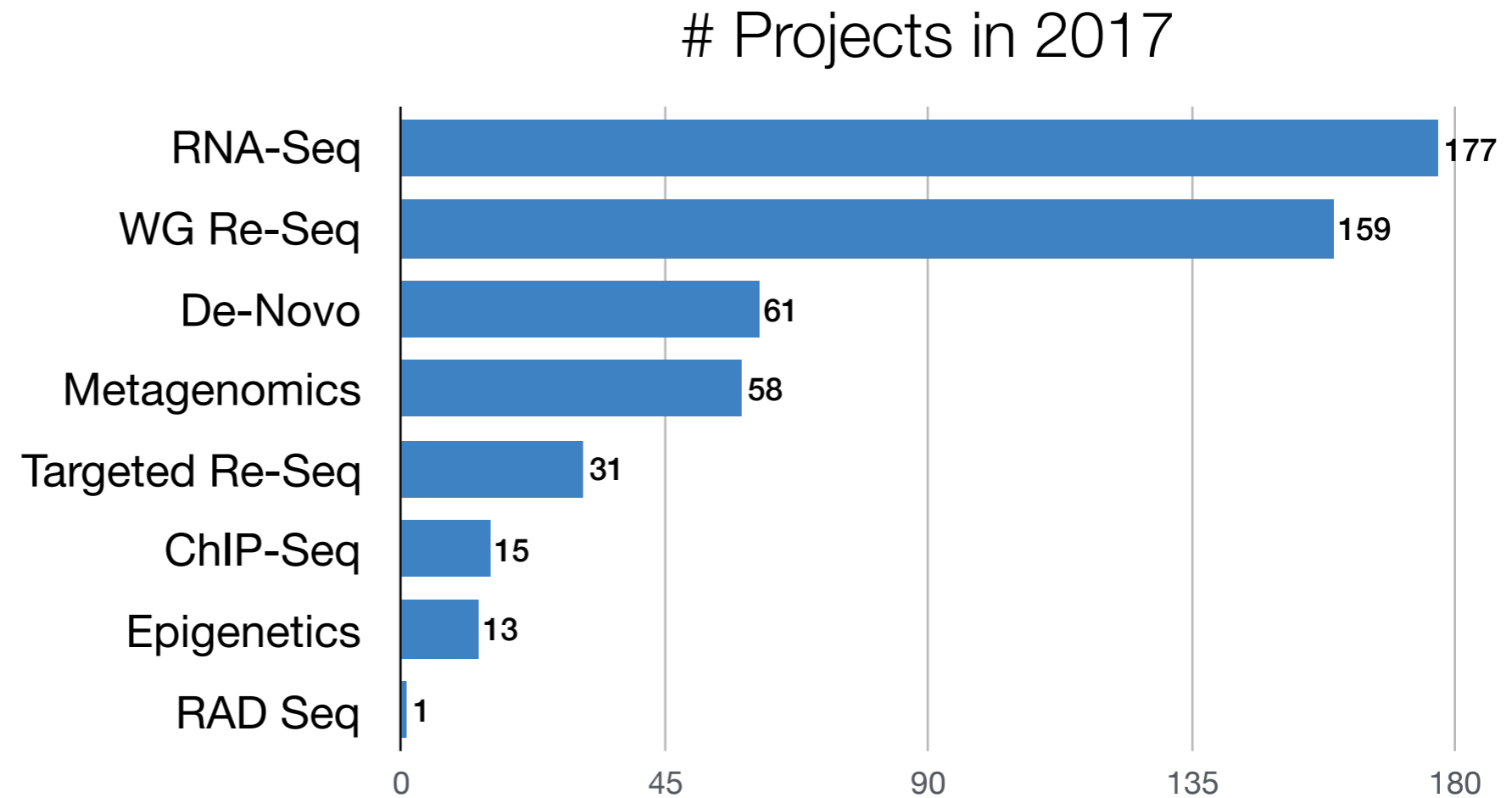
Data analysis
pipelines included

SciLifeLab

NGI stockholm

RNA-Seq: NGI Stockholm

- RNA-seq is the most common project type



RNA-Seq: NGI Stockholm

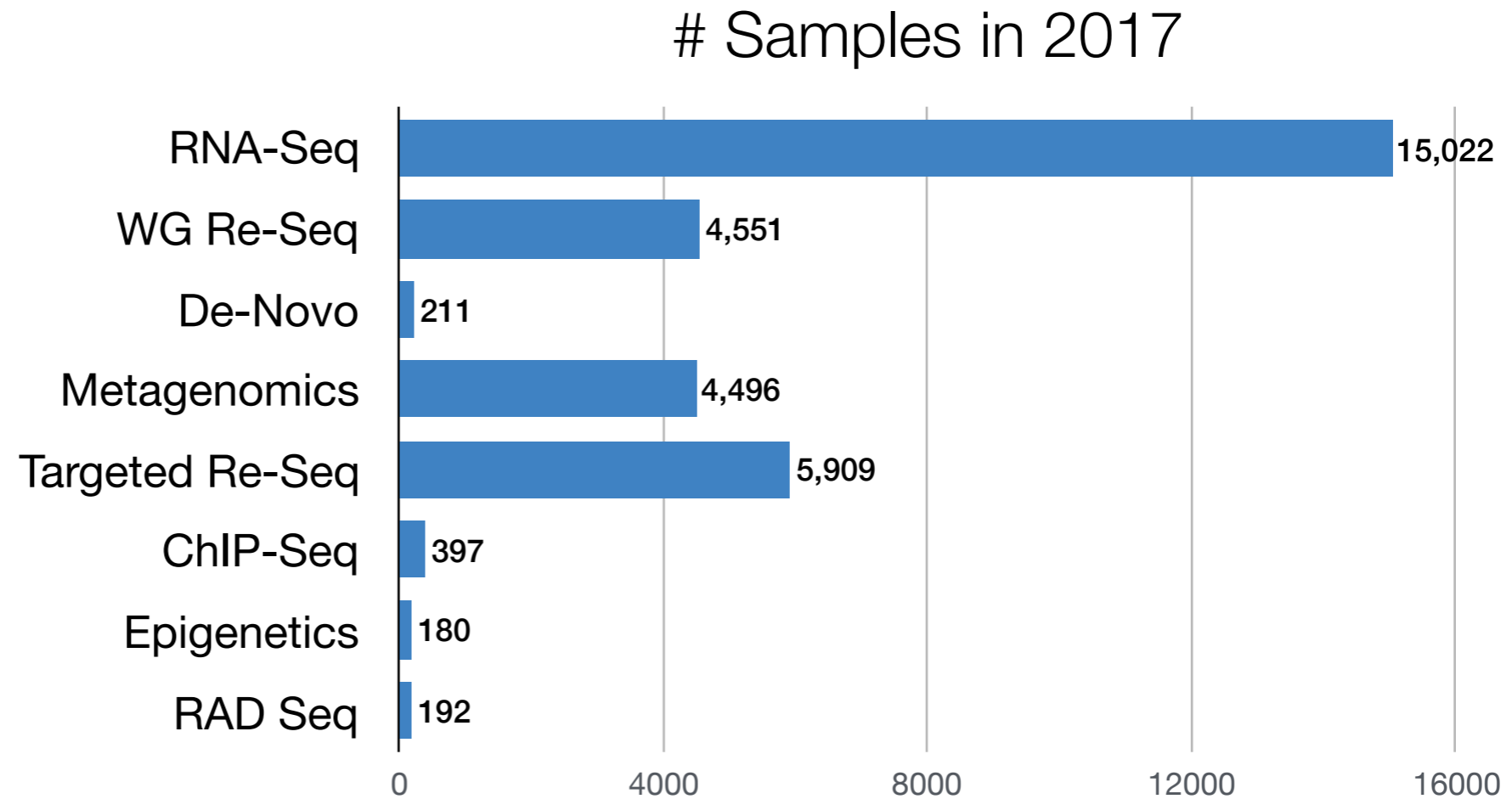
- RNA-seq is the most common project type

- Production protocols:

- TruSeq (poly-A)
- RiboZero

- In development:

- SMARTer Pico
- RNA Exome



RNA-Seq: NGI Stockholm

- RNA-seq is the most common project type
- Production protocols:
 - TruSeq (poly-A)
 - RiboZero
- In development:
 - SMARTer Pico
 - RNA Exome



RNA-Seq Pipeline

- Takes raw FastQ sequencing data as input
- Provides range of results
 - Alignments (BAM)
 - Gene counts (Counts, FPKM)
 - Quality Control
- First RNA Pipeline running since 2012
- Second RNA Pipeline in use since April 2017

RNA-Seq Pipeline

- Takes raw FastQ sequencing data as input
- Provides range of results
 - Alignments (BAM)
 - Gene counts (Counts, FPKM)
 - Quality Control
- First RNA Pipeline running since 2012
- Second RNA Pipeline in use since April 2017

RNA-Seq Pipeline

nf-core/rnaseq

FastQC	<i>Sequence QC</i>
TrimGalore!	<i>Read trimming</i>
STAR	<i>Alignment</i>
dupRadar	<i>Duplication QC</i>
featureCounts	<i>Gene counts</i>
StringTie	<i>Normalised FPKM</i>
RSeQC	<i>Alignments QC</i>
Preseq	<i>Library complexity</i>
edgeR	<i>Heatmap, clustering</i>
MultiQC	<i>Reporting</i>

RNA-Seq Pipeline

nf-core/rnaseq

FastQ

BAM

TSV

HTML

FastQC

TrimGalore!

STAR

dupRadar

featureCounts

StringTie

RSeQC

Preseq

edgeR

MultiQC

Sequence QC

Read trimming

Alignment

Duplication QC

Gene counts

Normalised FPKM

Alignments QC

Library complexity

Heatmap, clustering

Reporting

Nextflow

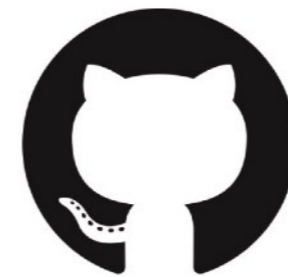
nextflow

- Tool to manage computational pipelines
- Handles interaction with compute infrastructure
- Easy to learn how to run, minimal oversight required

Nextflow

nextflow run <workflow>

CODE



GitHub

SOFTWARE



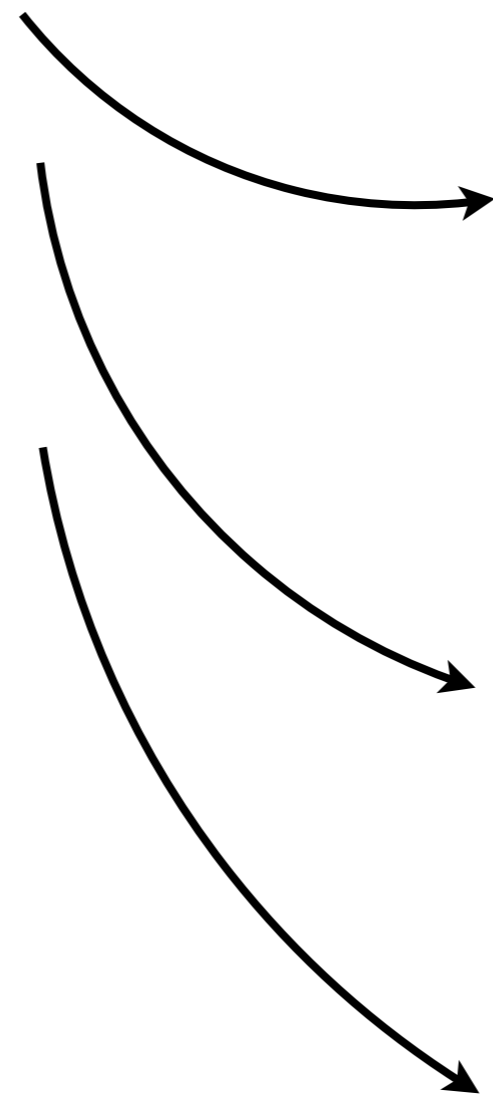
docker
Hub

COMPUTE

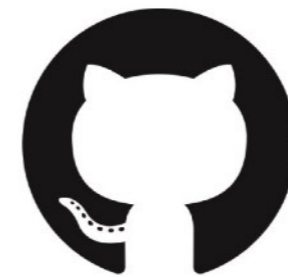


Nextflow

nextflow run <workflow>



CODE



GitHub

SOFTWARE



Singularity

COMPUTE

UPPNE



Nextflow

nextflow

```
#!/usr/bin/env nextflow

cheers=Channel.from "Bonjour","Ciao","Hello","Hola"

process sayHello {
  input:
  val x from cheers

  """
  echo $x world!
  """
}
```

Nextflow

nextflow

```
#!/usr/bin/env nextflow

input = Channel.fromFilePairs( params.reads )
process fastqc {
    input:
    file reads from input

    output:
    file "*_fastqc.{zip,html}" into results

    script:
    """
    fastqc -q $reads
    """
}
```

Nextflow

```
#!/usr/bin/env nextflow

input = Channel.fromFilePairs( params.reads )
process fastqc {
  input:
  file reads from input

  output:
  file "*_fastqc.{zip,html}" into results

  script:
  """
  fastqc -q $reads
  """
}
```

Default: Run locally, assume software is installed

```
docker {
  enabled = true
}

process {
  container = 'biocontainers/fastqc'
}
```



Run locally, use docker container for software dependencies

Nextflow

```
#!/usr/bin/env nextflow

input = Channel.fromFilePairs( params.reads )
process fastqc {
  input:
  file reads from input

  output:
  file "*_fastqc.{zip,html}" into results

  script:
  """
  fastqc -q $reads
  """
}
```

```
singularity {
  enabled = true
}

process {
  container = 'biocontainers/fastqc'
  executor = 'slurm'
  clusterOptions = { "-A b2017123" }
}
```



```
docker {
  enabled = true
}

process {
  container = 'biocontainers/fastqc'
}
```

Submit jobs to SLURM queue
Use Singularity for software

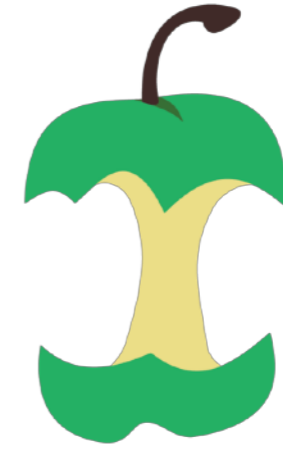


SciLifeLab

NGI stockholm



nf-core



CANCER RESEARCH UK

BEATSON INSTITUTE



SciLifeLab



Genome Institute of Singapore



International Agency for Research on Cancer



wellcome sanger institute

https://nf-co.re/

[Home](#)[Pipelines](#)[Usage](#)[Developers](#)[Tools](#)[About](#)

A community effort to collect a curated set of analysis pipelines built using Nextflow.

[VIEW PIPELINES](#)

For facilities

Highly optimised pipelines with excellent reporting. Validated releases ensure reproducibility.

For users

Portable, documented and easy to use workflows. Pipelines that you can trust.

For developers

Companion templates and tools help to validate your code and simplify common tasks.

Nextflow is an incredibly powerful and flexible workflow language.

nf-core pipelines adhere to strict guidelines - if one works, they all will.

Nextflow is an incredibly powerful and flexible workflow language.

nf-core pipelines adhere to strict guidelines - if one works, they all will.

Documentation

Extensive documentation covering installation, usage and description of output files ensures that you won't be left in the dark.



CI Testing

Every time a change is made to the pipeline code, nf-core pipelines use continuous-integration testing to ensure that nothing has broken.



Travis CI

Stable Releases

nf-core pipelines use GitHub releases to tag stable versions of the code and software, making pipeline runs totally reproducible.



Docker

Software dependencies are always available in a bundled docker container, which Nextflow can automatically download from dockerhub.



Singularity

If you're not able to use Docker, built-in support for Singularity can solve your HPC container problems. These are built from the docker containers.



Bioconda

Where possible, pipelines come with a bioconda environment file, allowing you to set up a new environment for the pipeline in a single command.



Get started in minutes

Nextflow lets you run nf-core pipelines on virtually any computing environment.

nf-core pipelines come with built-in support for [AWS iGenomes](#) with common species.

The nf-core companion tool makes it easy to list all available nf-core pipelines and shows which are available locally. Local versions are checked against the latest available release.

```
# Install nextflow
curl -s https://get.nextflow.io | bash
mv nextflow ~/bin
```

```
# Launch the RNAseq pipeline
nextflow run nf-core/RNAseq \
  -profile standard,docker \
  --genome GRCh37 \
  --reads "data/*_{R1,R2}.fastq.gz"
```

```
# Install nf-core tools
pip install nf-core
```

```
# List all nf-core pipelines and show available updates
nf-core list
```



Pipelines

Browse the **16** pipelines that are currently available as part of nf-core.

Available Pipelines

Can you think of another pipeline that would fit in well? [Let us know!](#)

Filter:

Released **4**Under development **12**

Sort:

Last Release

Alphabetical

Status

Stars

nf-core/eager ✓

★ 8

adna ancientdna pathogen-genomics population-genetics

A fully reproducible and state of the art ancient DNA analysis pipeline.

Version **2.0.2**

Published 3 days ago

nf-core/rnaseq ✓

★ 48

rna rna-seq

RNA sequencing analysis pipeline using STAR or HISAT2, with gene counts and quality control

Version **1.1**

Published 1 month ago

nf-core/hlatyping ✓

dna hla hla-typing immunology optitype personalized-medicine rna

Precision HLA typing from next-generation sequencing data

Version **1.1.1**

Published 3 months ago

nf-core/methylseq ✓

★ 15

bisulfite-sequencing dna-methylation methyl-seq

Methylation (Bisulfite-Sequencing) analysis pipeline using Bismark or bwa-meth + MethylDackel

Version **1.1**

Published 3 months ago

nf-core/rnafusion ⚠

nf-core/rrna-ampliseq ⚠

★ 9

nf-core/rnaseq

The screenshot shows the GitHub repository page for `nf-core/rnaseq`. At the top, there is a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. The repository name `nf-core/rnaseq` is displayed, along with its parent repository `ewels/nf-core-rnaseq`. Statistics show 26 Unwatch, 51 Star, and 82 Fork. The repository description is "RNA sequencing analysis pipeline using STAR or HISAT2, with gene counts and quality control" with a link to `http://nf-co.re`. Below the description are topic tags: `nf-core`, `nextflow`, `workflow`, `rna-seq`, `rna`, and `pipeline`. A progress bar shows 987 commits, 5 branches, 2 releases, 21 contributors, and MIT license. Action buttons include "Branch: master", "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download". A commit history table is visible below.

Commit	Message	Time
ewels	Merge pull request #101 from nf-core/dev	Latest commit 1cd5ab7 on 6 Oct
	assets	Remove branding, lowercase logo. 5 months ago
	bin	Use quotes for feature counts in MultiQC 2 months ago
	conf	Fix withName syntax for hebbe profile 2 months ago

— nf-core/rnaseq

README.md

nfcore/rnaseq Documentation

The nfcore/rnaseq documentation is split into the following files:

1. [Installation](#)
2. Pipeline configuration
 - [Local installation](#)
 - [Amazon Web Services \(aws\)](#)
 - [Swedish UPPMAX clusters](#)
 - [Swedish cs3e Hebbe cluster](#)
 - [Tübingen QBiC](#)
 - [CCGA Kiel](#)
 - [Adding your own system](#)
3. [Running the pipeline](#)
4. [Output and how to interpret the results](#)
5. [Troubleshooting](#)

- Running nextflow

Step 1: Install Nextflow

- Uppmax - load the Nextflow module
`module load nextflow`
- Anywhere (including Uppmax) - install Nextflow
`curl -s https://get.nextflow.io | bash`



Step 2: Try running NGI-RNAseq pipeline

```
nextflow run SciLifeLab/NGI-RNAseq --help
```

- Running nextflow

Step 3: Choose your reference

- Common organism - use iGenomes
`--genome GRCh37`
- Custom genome - Fasta + GTF (minimum)
`--fasta genome.fa --gtf genes.gtf`

Step 4: Organise your data

- One (if single-end) or two (if paired-end) FastQ per sample
- Everything in one directory, simple filenames help!

– Running nextflow

Step 5: Run the pipeline on your data

- Remember to run detached from your terminal
screen / tmux / nohup

Step 6: Check your results

- Read the Nextflow & MultiQC reports

Step 7: Delete temporary files

- Delete the `./work` directory, which holds all intermediates

Using Docker

```
nextflow run nf-core/rnaseq
  -profile docker
  --fasta genome.fa --gtf genes.gtf
  --reads "data/*_R{1,2}.fastq.gz"
```



- Can run anywhere with Docker
 - Downloads required software and runs in a container
 - Portable and reproducible.

Using UPPMAX

```
nextflow run nf-core/rnaseq
  -profile uppmax
  --project b2017123
  --genome GRCh37
  --reads "data/*_R{1,2}.fastq.gz"
```

UPPNE 



- Profile for UPPMAX
 - Knows about central iGenomes references
 - Uses centrally installed software

Using other clusters

```
nextflow run nf-core/rnaseq
  -profile hebbe
  --fasta genome.fa --gtf genes.gtf
  --reads "data/*_R{1,2}.fastq.gz"
```



- Can run just about anywhere
 - Supports local, SGE, LSF, SLURM, PBS/Torque, HTCondor, DRMAA, DNAnexus, Ignite, Kubernetes

Using AWS

```
nextflow run nf-core/rnaseq
  -profile aws
  --genome GRCh37
  --reads "s3://my-bucket/*_{1,2}.fq.gz"
  --outdir "s3://my-bucket/results/"
```



- Runs on the AWS cloud with Docker
 - Pay-as-you go, flexible computing
 - Can launch from anywhere with minimal configuration

Input data

```
ERROR ~ Cannot find any reads matching: XXXX
NB: Path needs to be enclosed in quotes!
NB: Path requires at least one * wildcard!
If this is single-end data, please specify
--singleEnd on the command line.
```

`--reads '*_R{1,2}.fastq.gz'`

`--reads '*.fastq.gz' --singleEnd`



`--reads sample.fastq.gz`

`--reads *_R{1,2}.fastq.gz`

`--reads '*.fastq.gz'`

– Read trimming

- Pipeline runs TrimGalore! to remove adapter contamination and low quality bases automatically
- Some library preps also include additional adapters
 - Will get poor alignment rates without additional trimming

```
--clip_r1 [int]
```

```
--clip_r2 [int]
```

```
--three_prime_clip_r1 [int]
```

```
--three_prime_clip_r2 [int]
```

Library strandedness

- Most RNA-seq data is strand-specific now
- Can be "forward-stranded" (same as transcript) or "reverse-stranded" (opposite to transcript)
- If wrong, QC will say most reads don't fall within genes
 - forward_stranded
 - reverse_stranded
 - unstranded

— Lib-prep presets

- There are some presets for common kits
- Clontech SMARTer PICO
 - Forward stranded, needs R1 5' 3bp and R2 3' 3bp trimming

`--pico`

- Please suggest others!

– Saving intermediates

- By default, the pipeline doesn't save some intermediate files to your final results directory
 - Reference genome indices that have been built
 - FastQ files from TrimGalore!
 - BAM files from STAR (we have BAMs from Picard)
- `--saveReference`
- `--saveTrimmed`
- `--saveAlignedIntermediates`

— Resuming pipelines

- If something goes wrong, you can resume a stopped pipeline
 - Will use cached versions of completed processes
 - NB: Only one hyphen! **-resume**
- Can resume specific past runs
 - Use **nextflow log** to find job names

```
nextflow run -resume job_name
```

— Customising output

`-name`

Give a name to your run. Used in logs and reports

`--outdir`

Specify the directory for saved results

`--aligner hisat2`

Use HiSAT2 instead of STAR for alignment

`--email`

Get e-mailed a summary report when the pipeline finishes

– Nextflow config files

- Can save a config file with defaults
 - Anything with two hyphens is a params

`./nextflow.config`

`~/.nextflow/config`

`-c /path/to/my.config`

```
params {  
  
    email = 'phil.ewels@scilifelab.se'  
    project = "b2017123"  
  
}
```


nf-core/rnaseq config

```
$ nextflow run nf-core/rnaseq -profile test,docker
```

```
N E X T F L O W ~ version 0.32.0
```

```
Launching `/home/travis/build/nf-core/rnaseq/main.nf` [golden_brenner] - revision:  
7c9a828c2b
```

```
=====
```



```
nf-core/rnaseq : RNA-Seq Best Practice v1.1
```

```
=====
```

Run Name	: golden_brenner
Reads	: data/*{1,2}.fastq.gz
Data Type	: Single-End
Genome	: false
Strandedness	: None
Trim R1	: 0
Trim R2	: 0
Trim 3' R1	: 0
Trim 3' R2	: 0
Aligner	: STAR
Fasta Ref	: https://github.com/nf-core/test-datasets/raw/rnaseq/reference/genome.fa
GTF Annotation	: https://github.com/nf-core/test-datasets/raw/rnaseq/reference/genes.gtf
Save Reference	: No
Save Trimmed	: No
Save Intermeds	: No
Max Memory	: 6 GB
Max CPUs	: 2
Max Time	: 2d
Output dir	: ./results
Working dir	: /home/travis/build/nf-core/rnaseq/test/work

Version control

The image shows two overlapping screenshots. The background screenshot is a GitHub release page for 'nf-core/rnaseq version 1.1'. It features a 'Releases' tab, a 'Draft a new release' button, and a 'Latest release' badge. The release is by 'ewels' and dated '6 Oct'. It lists two assets: 'Source code (zip)' and 'Source code (tar.gz)'. Below the assets is a 'Pipeline updates' section with a bulleted list of changes. The foreground screenshot is a Docker Cloud interface for the 'nfcore / rnaseq' repository, showing a list of tags: 'dev', '1.1', 'latest', and '1.0', each with a size of 1 GB and a timestamp indicating when it was last updated.

GitHub Release Page:

- Releases | Tags
- Draft a new release
- Latest release
- 1.1
- 1cd5ab7
- Verified
- ewels released this on 6 Oct
- Assets 2
 - Source code (zip)
 - Source code (tar.gz)
- Pipeline updates
 - Wrote docs and made minor tweaks to the
 - Removed the deprecated `uppmx-modules`
 - Updated the `hebbe` config profile to use t
 - Use new `workflow.manifest` variables in t
 - Updated minimum nextflow version to `0.`

Docker Cloud Tags Page:

- Showing 1-4 of 4 Tags
- dev 1 GB (Last updated 9 days ago)
- 1.1 1 GB (Last updated a month ago)
- latest 1 GB (Last updated a month ago)
- 1.0 1 GB (Last updated 3 months ago)

— Version control

- Pipeline is always released under a stable version tag
- Software versions and code reproducible
- For full reproducibility, specify version revision when running the pipeline

```
nextflow run nf-core/rnaseq -r v1.1
```

– Software Dependencies

- Already specified in most config profiles!



```
-profile standard,docker
```



```
-profile standard,docker
```

BIOCONDA[®]

```
-profile standard,conda
```

SciLifeLab

 NGI stockholm

<https://github.com/nf-core/rnaseq>

Running Offline

- If running offline, need to transfer the required files manually
- Pipeline files
- Singularity image

```
$ wget https://github.com/nf-core/rnaseq/archive/1.1.zip

$ singularity pull
  --name nf-core-rnaseq-1.1.simg
  docker://nfcore/rnaseq:1.1

$ ## transfer files to offline cluster,
  ## eg. ~/pipelines/
```

```
$ cd ~/pipelines/

$ unzip 1.1.zip -d .

$ cd ~/my_data/

$ nextflow run ~/pipelines/nf-core/rnaseq-v1.1/
  -with-singularity ~/pipelines/nf-core-rnaseq-1.1.simg
  --reads "data/*{1,2}.fq.gz"
  ## other normal parameters
```


Conclusion

- Use nf-core/rnaseq to prepare your data if you want:
 - To not have to remember every parameter for STAR
 - Extreme reproducibility
 - Ability to run on virtually any environment
- Now running for all RNA projects at NGI-Stockholm

nf-core/rnaseq 🍏

Conclusion

Phil Ewels

✉ phil.ewels@scilifelab.se

🐙 [ewels](#)

🐦 [tallphil](#)

Acknowledgements

Rickard Hammarén

Max Käller

Denis Moreno

NGI Stockholm Genomics

Applications Development Group

<https://nf-co.re>

🐙 [nf-core](#)

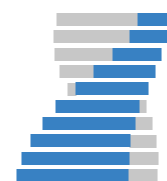
🐦 [nf_core](#)

support@ngisweden.se

<https://opensource.scilifelab.se>

🐦 [ngisweden](#)

SciLifeLab



NGI stockholm