PL, NGP och Seqera

Halfdan Rydbeck

2024-10-24

Table of contents

Första sliden

Alterative titel: Bioinformatik på PL?

Nu skall vi prata om arbetsflödes hanteringsprogram och om hur de kan komma till nytta för oss på Precisions medicinskt laboratorium

Den här boken är skriven med "Quarto boo"k. This is a Quarto book. För att lära mer om Quarto books kolla på https://quarto.org/docs/books.

Blir många bilder

Jag har ritat många diagram här fritt tagna ut fantasin. Eftersom det är så många (nya) begrepp som presenteras så tror jag det underlättar om man kan knyta ihop dem i mentala modellr. Mina diagram och bilder är ett försök att underllätta den processen.

Svenska

jag försöker göra presenationen på svenska. Det fungera bra när man diskuterar organisationer etc men ibland finns mej veterligen inga bra motsvarande svenska or. Då får vi slå över.

Useful links

https://alexd106.github.io/intro2R/Github_intro.html#Option_2_-_RStudio_first

Reg ostergotland Logos

https://www.regionostergotland.se/ro/press/grafisk-profil/ladda-ner-logotyp

Ploting Hex stickers

https://github.com/GuangchuangYu/hexSticker

Is there a package to plot multiple stickers??

1 Vad är PL?

Precisions medicinsk laboratorium (PL) i Linköping utför laboratorie tjänster och analys för de **tre verksamhetsområdena** Klinisk patologi, Klinisk genetik och Klinisk mikrobiologi.

- För Pat och klin_gen så handlar det om att PL blir till-sänt prov och medföljande remiss och i bästa fall sänder tillbaka bakomliggande genetiska varianter.
- För Mikro så handlar det om att man får bakterieprov och sänder tillbaka information om potentiella resistensgener och MLST varainter.

PL-Linköping utgör en av sju Genomic Medicine Centers (GMCs) i Genome Medicine Sweden (GMS) som är ett samarbete mellan regioner med universtietessjukvård som finansieras ungeför likvärdigt av: - Vinnova (innovasionsmyndigheten) - universitetsjukhus regionerna

1.1 PL-NGS

Här ses PL isolerat utan de servade verskamhetsområdena.PL består av tre enheter som vardera servar de tre versamhetsområdena. Hexagonen i kärnan representerar det som de har gemensamt:

- Ledning
- Sekvenseringsmaskiner
  - (Presentationen kommer att begränsa sig till att se på PL utifrån ett sekvenserings perspektiv, eftersm det är det som jag jobbar med.).
- Datornätverk

2 PL ingår i det nationella nätverket Genome Medicine Sweden (GMS)

Nätverket har sju noder, eller Genomisk Medicin Centrum (GMCs) som utgörs ave regionerna universitetssjukhus.

På universitetssidan så har de ett systernätverk som organiseras av SciLifeLab där noderna, eller Clinical Genomics (CG) key services, ligger på sju av landets universitet. Det finns en vision om lokal synergi mellan Clinical Genomcs och Genomiks Medicin Centra, där meningen är att CGs skall stötta GMCs med kompetens.

GMS finansierar till hälften av projekt medlen som ansökts om hos Vinnova och till hälfden av de involverade regionerna.

Det är framför allt två projekt som berör PL:

- Swelife
- Systemdemonstrator **(NGP)**

2.1 Data delning och central analys via NGP är del i ett större EU sammanhang

Motsvarigheten till NGP på EU-nivå är förmodligen European Health Data Space (EHDS). Drivande organistino bakom är [European Healt Union](https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/promoting-our-european-way-life/european-health-union_en)

EHDS är viktigt för de är drivande i tolkning och tilläggslagstiftning kring General Data Protection Regulation (GDPR)

2.1.1

3 Vad är NGP?

NGP är GMS's sätt bana väg för transformera dataflöden inom svensk/nationell klinisk diagnostik och forskning.

- data strukturering
- data standardisering
- applikations utveckling
- fördjupas samarbete med forsknigs infrastrukturen
- Tillgängliggörande av data för akademin, myndigheter och näringsliv

Diagrammet visar att NGP utgörs av tree functionlaiteter som benämns NGP repository (NGPr), NGP commander (NGPc) och NGP indexing (NGPi)

3.1 Tre funktionaliteter

3.1.1 NGPr

Det är hit det är tänkt att GMC-nodena skall kunna lasta upp sin data. NGPc är ett fleranvändar system. Varje GMS nod är en användare eller Tenant som det kallas i moln tjänst användbar språk. NGPC har ett data lagrings segment som är privat för varja Tenant. I tillägg så har det ett segment som är gemensamt.

3.1.2 NGPi

Detta är en funktinalitet som skall göra det lättare och snabbare att hitta relevant data för olika användare av systemet.

3.1.3 NGPc

Har kommer analys verktyg att installeras som i huvudsak kommer att utgöras av **pipelines eller workflows** eller bioinformatiska arbetsflöden. I sin allra enklaste definitin så är ett sådan arbetsflöde en antal program som kopplats ihop för att till tillsammans utföra uppgifter vars komplexitet gör att det inte finns enskilda program som kan utföra dem.

3.2 Nytta för GMCerna

- Tolknignsverktyg

3.3 Övriga benefaktorer

- Universitet - DDLS
- Företag
- Myndigheter, Fohm

4 Vad är ett bioinformatiskt arbetsflöde?

4.1 Exempel från Sällsynta diagnoser - Del i ett större arbetsflöde

Här visas en schematisering av ett arbetsflöde på PL-rd. Schematiskeringen är förenklad och framför allt i det bioinformatiska flöden så har saker som tex identifiering av kopietalsförändringar utelämnats.

4.1.1 Manuell/fysisk del

1. Det startar med att prov kommer in
2. DNA extraheras och bibliotek prepareras och lastas på en sekvenseringsmaksin

4.1.2 Bioinformatisk del

1. Sekvenseringsmaskinen skriver sekvens och kvalitetsdata data, ursprungligen ofta i nått företagsspecifikt format.
2. Textfiler, sk fastq-filer, en för varje sekvenserad fragment, med sekvens och kvalitetsdata tillverkas.
3. Fragment-sekvens-datan filtreras och och alignas (radas upp) mot ett referensgenome
4. Varianter i förhållande till referens genomet identifieras och sparas i en Variant Call File VCF.
5. Varianterna annoteras med information som möjliggör efterkommande tolkning

4.1.3 Samarbete människa maskin

1. Tolkning

4.2 Liknande arbetsflöden finns ju också för Patologi och Mikrogrupperna

Det visade arbetsflödet skall klara helgenoms analyser vilket gör bioinformatik delen mer resurskrävande än de övriga flödena.

4.3

5 Förutom alla fördelar, som att stå på varandras axlar - så finns det många utmaningar med bioinformatiska arbetsflöden

5.1 Bioinformatiska arbetsflöden behöver nån form av beräknings kraft fär att köras.

- I huvudsak av effektiviseringssjäl/ekonomiska så sker det en stor omändring av av ekonomiska själ hur resurskrävande programmatiska uppgifter utförs. Konceptet dator och server håller på att bli förlegat ock ersätts av, HPC, molnberäkning och diverse andra begrepp. Bland annat hårdvaruutvecklingen ställer nya krav på mjukvara, och också på det området så föds kontinuerligt nya begrepp.

5.2 Konkurenskraftig utveckling

- För att förbli konkurrenskraftigt så måste det finnas resurser för intensiv utveckling av flödet
- Av ingående program och av flödet som sådant
- Kräver ett system för att hantera versioner både av flödet och av de ingående komponenterna

5.3 Skall kunna köras var som helst

- Från tex forsknings- och hälso-perspektiv är det oftast önskvärt att analysresultaten är reproducerbara mellan labb.
- Det kräver bland annat att flödet skall vara så flexibelt som möjligt med tanke på vilka hårdvaru-konfigurationer som det kan köras på.
- resultaten skall bli desamma oavsett å vilket system som flödet körs på.

5.4 Undvika att köra om delar där riktiga resultat redan genererats - tar lång tid att köra

- Därför svårt att felsöka
- Tar lång tid att köra om vid stop

5.5 Skall följa lagar och regler för den institution som använder dem

- IVDR

5.6 Variablelt behov av resurser från beräknings platformen (runtime management); Kräver ett skalerbart system

- RAM
- Hårddiskutrymme
- Parallell processering (kräver många processorer/cores)

5.7 Olika flöden skall kunna köras samtidigt av flera olika användare

Här står alternativet mellan:

- En skalerbar fleranvändarmiljö
- flera enskilda småskaliga miljöer, en för varje användare

5.7.1 Mjukvarutveckling baserad på storskaligt samarbete och system för versionskontrollering

6 Vad är Seqera?

Based on Knowledge of high-throughput analysis and modern software engineeering gained from building Nextflow the same people have

created a platform to make data-intensive research scalable, flexible, and collaborative

Why do I mention a company the first thing after talking about bioinformatics workflows? Because it has been growing out of free stuff initaiteives, becasue it offers a way to deal with sensitive data and because it offers a GUI interface to handling, running pipelines an their output. I think its my best bet to make this presentaiton interesting for you to show you (Emedgene endorsers) a a pipelein can be run in seqera cloud.

- Seqera's affärs ide är att underlätta att arbeta med arbetsflöden
- Grundarna ligger bakom två icke-kommersiella initiativ
  - Ett workflow hanterings verktyg som bygger på ett eget dedikerat programmeringsspråk, Nextflow
  - Ett regel-set och socialt forum för uniform utvecklande av workflows - nf-core

Seqera presents itself as an open science company. For me that is hard to comprehend what it is. What dous that mean? They also say they want to be a centrla hub at open science? Does open science want/need a hub that is comercail? I dont knwo. In any case what

Seqera would not be the only compay that open science relies on. Github, Docker and Singluarity are other examples. Undrstand their business model is beyond my understande. What I can grasp though is theat they provide interesting useful products for free.

There is also of course the micro array sequencing platfors tha tare pillars of open genomics science. Many of the ENxtfow pipleline are in direct competintion with Illumina products like Emedgene. For me the most open poduct would win form any perspective,be it IVDR or usability.

6.1 Many develpers of groupleaders, Nextflow and nf-core are employess at Seqera

6.2 Secrets, secure handling of sensitive data

https://seqera.io/blog/pipeline-secrets-secure-handling-of-sensitive-information-in-tower/

6.2.1 Examples

https://aws.amazon.com/blogs/hpc/leveraging-seqera-platform-on-aws-batch-for-machine-learning-workflows-part-1-of-2/

6.3 Nextflow

solve problems with reproducible workflows

7 Hur kan liknande arbetsflöde(n) köras(på bästa sätt)?

I dagsläget så verkar det som att det i flesta fall blir mest effektivt att använda sig av en skalerbar fleranvändarmiljö, dvs ett cluster eller en moln miljö.

I vårt fall när det skall tas hänsyn till säkerhetsklassad/känslig data så krävs antingen lokalt system eller ett system med säker inloggning.

7.1 Nya lösningar inom både hardvaru- och mjukvaru-design gör det allt lättare att köra men också utvecka och vidarutveckla arbetsföden

7.1.1 Hårdvara +Operativesystem/hårdvaruhateringssystem

7.1.1.1 HPC

- NGP (altair grid engine)
- PDC-Dardel (slurm)

7.1.1.2 Moln

- AWS
- Azure

7.1.1.3 Lokal server

- Blir svårhanterligt...

7.1.2 Beräkningsmiljöer

- Altair Grid engine
- Slurm

7.1.3 Mjukvara - hanteringsprogram

- Version
- container
- arbetsflöde

8 Hårdvara och beräkningsmiljöer

Vi ser på detta utfrån arbetsflödeshanterings program perspektiv. Dessa program behöver ha gränssnitt kan samarbeta med de olika typeran av hårdvara

Viktigt kriteria är att att det skall gå att lagra och analysera sensitiv klinisk patient data. Det finns ett EU belsut om att det skall vara lova att lagara och analysera den typen av data på molnlösningar (med hårdvra i Europa?).

Det gör att vi i nuläget inte kan dra igång med molntjänster.

8.1 HPC

Ett HPC består av en grupp datorer/servrar(noder) som kan samarbeta för att utföra en gemensam beräknings uppgift. Varje nod i tar emot och processerar beräknings uppgifter oberoende av varandra. Noderna koordinerar och synkroniserar uppgifterna för att till slut producerar ett sammanslaget resultat.

Även om arkitekturen hos ett super computing cluster eller HPC kan vara no så komplicerad och skilja sig mycket från en generation till nästa så är det inte nödvändigt för den generella användaren av systemet. Det viktiga är veta processen för hur man skickar iväg ett jobb som ansöker om beräkningsnoder (fysiska grupperingar av processorer) som gör de beräkningar som man är intresserade av. Ett HPC är byggt för att ha flera användare så därför kommer det alltid med en mjukavara som kan sätta sätat upp köer för användare.

Det finns en handfull mjuk-varor för kö-hantering som dominerar användarmarknaden. Några är gratis och några kostar pengar.

Det som är genomgående för mjukvaran är att de erbjuder möjligheten att formulera skript som definierar:

- Hur mycket datorkraft man vill ta från HPCn

8.1.1 Komponenter

8.1.1.1 Beräkning

- Head eller login node
- Vanliga beräknings noder
  - CPU, GPU

8.1.1.2 Lagring

- Fysisk lagring (on premise). Kan ofta vara överlägser ur hsitghets/performacne perspektiv. Tillåter parallella filsystem oc hlow latancy access.
- Moln baserad lagring. Skalerbart. Tillåter ofta hög hastighet(numera).
- Hybrid

8.1.1.3 Nätverk

Noderna måste kunna kommunicera med varann. Viktigt är att upnå högst möjlgia hastighet.

8.1.1.4 HPC Jobbschemaläggare och resurshanterare

- Vikitg komponent hos HPC

8.1.2 Vanliga typer av arkitekturer

Paralllel, cluster och grid beräkning

En HPC desing can kombinera Paralllel, cluster och grid design alltså innefatta alal tre.

8.1.2.1 Parallell beräkning

Förmåga att distribuera en beräknings uppgift eller data på flera noder/processorer

8.1.2.2 Cluster beräkning

Koppla ihop flera datorer till en enhet

8.1.2.3 Grid och distribuerar beräkning

Handlar om att koppla ihop geografiskt spridda beräkngns resurser till en virtuell enhet. Så til skillnad från ett cluster så involverar en grid enheter från flera olika platser och organisationer.

8.1.3 NGP (äger vår data själv)

Det finns inte så mycket dokumentation för NGP från ett användar perspektiv/ Ingan användamanual. Där för kan vi ta ett annat svensk HPC upsätt för icke sensitive data.

8.1.3.1 Lokalt HPC, grid computing

Det kallas grid computing för att det kommersiella köhantinrings programmet heter Altaier GRID engine.

Plan på att installera ett lokalt HPC. De kommer att likna NGP i sin uppsättning

8.1.4 PDC-Dardel

8.2 Moln/internet beräkning

Tillgång till servrar, lagring, database, nätverk och mjukvara, analysverktyg och intelligens.

8.2.1 Intelligence

- Elasticitet i tillgång på beräkningsresurser. Förmågan att dynamiskt dra in nya data resurser
- Datorstödd affärsanalys (analytiska processer som undersöker data och presetnerar anvädnbar information baserade på till exemple rapporter)
- Ofta/alltid affärsdrivande och kräver en betalnings modell. Enskilda användarkonton där var och en betalar för sig.
- Oerhört centraliserad resurs med stor kompentens (AWS/Azure vs. NAISS/GMS)

8.2.2 Wikipedia

Moln beräkning innefattar så många olika saker att en definiton riskear att bli vag.

- On demand, självbetjäning
- Tillgänglig för alla sortes enheter,. Mobi, surfplattor och arbetsstationer.
- Snabb elasticitet
- Övervakad resursanvändning #### Påstådda fördelar
- Kostnade. Betalar bara för när resurserna används
- Webgränssnitt gär att man kopal upp sig med vad som helst som har en web browser
- Inget on-premise underhåll

8.2.2.1 Tänkbara nackdelar

- data säkerhet. Moln användare anförtror sin data till tredjeparts leverantörer.
- Reducerad transparans. kan sakan full översikt/insikt i hur resurser övervakas och rapporterass
- Fullstädigt översikt över hur systement fungara kan bli omöjligt.Något som kan utläsas av metaforen “moln”.
- Migratin från moln kan vara komplicerat
- Impementering av mjukvara och arbetsflöden kan drabbas av problem som har att göra med bl.a. distriburad beräkningskapacitet.
- Om man inte håller bra koll på vilak resurser som körs och vad de kostar så kan man få en överaskning i form av kostander.

8.2.2.2 FIgur text

8.2.2.2.1 Infrastructure as a service (IaaS)

Tex svervrar, Lagringsdiskar och nätverk

EC2 is and IaaS

8.2.2.2.2 Platform as a service (PaaS)

Tex operativ system, databaser, säkerhetsprogram

8.2.2.2.3 Software as a service (SaaS)

Enskilda program. I assume this is when Netflix, AirBnB are examples of Saas on AWS

Famous Saas companies:

Adobe Zoom Microsoft

8.2.3 AWS

On demand molnberäkningslösning som inkluderar 200+ tjänster, platformar och APIs som används av företag, myndigheter och privatpersonen som alla betalar efter de resurser de använder.

Elasitic Comput (EC2), Amazon's virtualla beräknings service, Glacier, en lågpris moln lagrings service, och S3, Amazon's lagrings system, är tre grund componenter i AWS.

8.3 Lokal server

9 Jobbschemaläggare och resurshanterare

9.1 Gratis

9.1.1 Slurm

SLURMs primära funktion är att allokera resurser inom klustret till dess användare . Resurshantering kan innefatta hantering av noder, sockets, kärnor och hypertrådar. Dessutom kan resursallokering baserad på topologi, mjukvarulicenser och generiska resurser som GPU:er hanteras av SLURM

https://slurm.schedmd.com/documentation.html

https://blogs.oracle.com/research/post/a-beginners-guide-to-slurm?fireglass_rsn=true#fireglass_params&tabid=a133ec835f7b2014&start_with_session_counter=2&application_server_address=sg-integration2-europe-west3.prod.fire.glass

9.1.1.1 Submitta ett jobb

You can submit a job script to the Slurm queue system from the login node with:

```
sbatch mitt_slurm_jobb.sh
```

mitt_slurm_jobb.sh

```
#!/bin/bash -l
# The -l above is required to get the full environment with modules

# Set the allocation to be charged for this job
# not required if you have set a default allocation
#SBATCH -A naissYYYY-X-XX

# The name of the script is myjob
#SBATCH -J myjob
```

```
# The partition
#SBATCH -p main

# 10 hours wall-clock time will be given to this job
#SBATCH -t 10:00:00

# Number of nodes
#SBATCH --nodes=4

# Number of MPI processes per node
#SBATCH --ntasks-per-node=128

# Run the executable named myexe
# and write the output into my_output_file
srun ./myexe > my_output_file
```

Running Bowtie

```
#!/bin/bash
#SBATCH --job-name=bowtie2_example
#SBATCH --cpus-per-task=8
#SBATCH --time=00:10:00
#SBATCH -o Bowtie_test.o%j
#SBATCH --partition=standard
#SBATCH --account=<YOUR_ALLOCATION>

#Load the Bowtie Module
module load gcc
module load bowtie2

# Change to temp working directory with example files
cd /scratch/$USER/bowtie_temp

# Indexing a reference genome
bowtie2-build ./example/reference/lambda_virus.fa lambda_virus

# Aligning example reads, standard example
bowtie2 -p $SLURM_CPUS_PER_TASK -x lambda_virus -U ./example/reads/reads_1.fq -S align.sam

# Paired-end example
bowtie2 -p $SLURM_CPUS_PER_TASK -x lambda_virus -1 ./example/reads/reads_1.fq -2 ./example/re
```

```
# Local alignment example
bowtie2 -p $SLURM_CPUS_PER_TASK --local -x lambda_virus -U ./example/reads/longreads.fq -S al
```

9.2 Kommersiell

9.2.1 Altair Gridengine

9.2.1.1 Submitta ett jobb

```
qsub -V -b n -cwd mitt_gridengine_jobb.sh
```

mitt_gridengine_jobb.sh

```
#!/bin/bash
                    #$ -N run_bowtie2
                    #$ -cwd
                    #$ -pe smp 6
                    #$ -l h_vmem=6G


                    infile=/data/bioinfo/READS2/R1_001.fastq.gz
                    outfile=/data/bioinfo/READS2/aln/R1_001.sam
                    btindex=/data/bioinfo/genome_data/Caenorhabditis_elegans/UCSC/ce10/Sequer

                    gzip -dc $infile | bowtie  --chunkmbs 300 --best -m 1 -p 6 --phred33 -q S
```

10 Vad kan versionskontrolleras?

Varje nf-core arbetsflöde görs tillgängligt via Github en website med det huvudsakliga syftet att dela versionskontrollerade project över internet. Versionkontrollering är användbart för nästan allt skapande som kan utföras på en dator. Här pratar vi om det utifrån perspekivet att arbetsflödeshanterings program

- En fil
- En folder med filer och subfoldrar
- Den här presentationen
- Ett arbetsfödesschema

10.1 Det allra enkaste exemplet - en fil

Som en textfil, eller en program fil.

Istället för att spara filen som synliga odokumenterade versioner.Tex Version_1,version2,final_version_3
Så använder man ett versions hanterings program som låter oss spara versioner i en dold databas/repository döljer och ber oss som att dokumentera ändringarna i varje version. Med verktyget såkan vi och vilja vilka ändringar som gjort som skall följa med in i ett version shot.

10.2 En folder

Smam sak kan göras simultatnt för flera filer i en foldr

10.3 Den här presentationen

den här presentationen har utveckalts med versins hanterigns progam

10.4 Ett arbetsflödes schema

Programmatiska aarbetsflöden utveckal under versins kontroll

11 git, ett system för versionskontrollering

11.1 Vad kan versionkontrolleras?

- en fil
- En folder (-strukture) med flera filer

11.2 git

Git bygger på att man har en folder (workspace, working directory, project folder) som innehåller ett ett flera dokument som man vet att man kommer att vidareutveckla över tid med viss osäkerhet, där man kan ångra sig och därför behöva återgå till tidigare versioner. Genom att gömma alla versioner utom den som man jobbar på för tillfället så minimerar programmet den distraherande påverkan som det kan ha att se flera versioer av sitt/sina dokument samtidigt.

Eftersom vidarutvecklingen sker över tid så samarbetar man alltid åtminstone med sitt framtida jag, tex genom att dokumentera/sammanfatta ändringarna som gjorts i varje version. Det underlättar att vid ett senare tillfälle senare bestämma vilken version man vill återgå till. Git underlättar också samarbeta med andra tex genom att detektera när två filändringar överlappar/är motstridigar och erbjuda verktyg för att editera/slå ihop de motstridiga ändringarna.

Efter att ha installera programmet så börjar arbetet med versionkontollering genom att initiera en s.k. repository i föräldrafoldern för dina dokument.

```
git init
```

Kommandot skapat en gömd folder ".git" i föräldrafolder. I den gömda foldern så kommer dokumenterade ögonblicksbilder av din/dina fil/filer att sparas.

Arbetsgången med git innnhåller tre centrala moment som motsvars av tre filtillstånd:

1. Gör filändringar; filen/filerna är ändrad(e)

2. Välj ändringar som skall förevigas i en ögonblicksbild (med commandot git add); Filens/filernas ändringar har valts ut till nästa ögonblicksbild; staged
3. Spara utvallda ändringar i som en ögnblickbild i .git-databsen/repositoryn tilllsammans med dokumentation (med commandot git commit -m); Filens ändringar har sparats i en ögonblicksbild; commited

Överkurs är sedan att lära sig hur man kan gå tillbaka till (checka ut) tidigare versioner.

11.3 Några egenskaper hos git

11.3.1 Tillåter icke linjärt skapande

Dvs att man kan jobba på olika delar samtidigt. Du jobbar på en fil som berör ett visst samnahang av det du vill säga med ditt projekt. Så får du en idee om nått du vill säga om ett helt annat samanhang. Då kan du spara det du höll på med börja jobab med en nya ideen och sedan problem fritt återgå til det du höll på med.

11.3.2 Simultan/paralllel utveckling

Flera kan samarbeta på ett dokument utan att riskera att skiva över varandras bidrag.

11.3.3 Gör det väldigt svårt att förlora material som en gång sparats in en version.

11.4 Github

Github är en website med den huvudsakliga funktionen att dela versionskontrollerade projekt över internet.

12 Kollektivt versionskontrollerad mjukvaruutveckling A

Molnbaserat Git kodförråd (repository). Det gör det lättare för personer och grupper att använda GIt för versionskontrollering och samarbete.

Github gränssnittet är användarvänligt nog så att tom nyblivna programmerare (ni) kan använda det. Github är så användarvänligt att en del personer tom använder det för att skriva böcker eller PhD uppsatser.

Vem som helst kan skapa ett konto, logga in och lasta upp versions kontrollerade kodförråd/dokumet foldrar.

12.1 Arbetsfolder

Dett är platsen där du jobbabr för tillfället, där dina filer håller till. Den platsen kallas också "untracked" område hos/av git. Filändringar kommer att markeras och bli sedda i arbetsträdet/arbetsfoldern. Om du gör filändringar här utan vidare tilltag och sedan raderar/skriver över ändringarna så kommer det att vara förlorade. Dette eftersom ändringarna ännu inte sagt att git skall bry sig om ändringaran. Om man gr ändirngar där såk ommer git att se dem, men inte förren git blir tillsgt att "Hej, föj de här filerna ändringar", kommer git att spara nått som sker med dem.

12.2 Staging area/index/förberdelse område/fil

The staging area är en fil i Git foldern som sparar information om vad som tas med i nästa "commit"/ögonblicksbild.

12.3 Lokalt kodförråd/.git foldern/ Git foldern

Det är här sm GIt spara metadatan och objectdatabasen för projektet. Git foldern är den som lasta upp till github och som sedna kan kopieras när man klanr från en annan dataor.

12.4 Github/fjärr kodförråd

Det här är . git foldern som lastats upp till Github.

13 Kollektivt versionskontrollerad mjukvaruutveckling B

14 Ikoner som kommer användas för versionskontrollerande enheterna

15 Program utvecklade i en specifik hårdvarukonfiguration skall kunna köras i vilken som helst annan.

15.1 Vad är software dependancies

En program fil som skrivs fungerar alltid i en kontext av andra program filer och för att fungera som tänkt. Programeringsspråk baserar sig ofta på kodbibliotek, "libraires" som kan refereras i en programfil för att användas för sin dedikerade uppgift. Det finns tex kodbibliotek för att hämta dta ur en databas. Då kan en mjukvara beroende av det kodbilboteket för att kommunicera med databaser för att fungera som tänkt.

Ett software dependancy är en kodbibliotek eller paket som återanvänds i ett mjukvara. Tex så kan ett maskinlärningsprojekt anropa en pythn bibliotek för att bygga modeller.

15.1.1 Conda packages (paket)

15.2 Container hanterings program

15.2.1 Singularity

15.2.2 Docker

16 Container hanteringsprogram och container register

17 Arbetsflödehanteringsprogram, Nextflow, ett av många

17.1

Nextflow är en gratis och open-source programvara, som utvecklas av företaget Seqera labs.

Nextflow har/är ett eget dedikerat programmeringsspråk, ett s.k. Domain Specific Language (DSL) som tillvärkningen av pipelines baserar sig på.

Språket är tillverkat baserat på samma idee som Linux baserar sig på. Anvädn små kraftfulla kommondolinjebaserade program som när de länkas samma underlättar utförandet av komplexa datahanterings uppgifter.

17.2 Funktionskraven hos ett arbetsflödes hanteringsprogram omfattar mer än bara att knyta ihop ett antal program till ett pärlband.

det omfattar dessutom

- skalerbarhet
- reproducerbarhet
- förmåga att integrerar mjukvarupaket, programmiljö hanteringsprogram som Docker, Singluarity och Conda för att möjligöra att sammankoppla scriptspråk såsom BASH, R och Python.
- Förenkla att köra pipelinen på olik platformar som tex cloud eller HPC-baserade infrastrukturer
- hantering mha tex social media av en friviligbaserade intressegrupp för utveckling av standarder för och arbetfödena i sig själva som är av gemensamt intresse. Användar och utvecklare socialisear i ett stort virrvarr av olika intressen. Slack, github; Seqera??

Om ni frågar mig så är det förvirrande att Nextflow används som ett paraply begrepp för flera funktioner.

17.3 Nextflow är

17.3.1 Ett scriptspråk dedikerat för att bygga programmatikska arbetsföden

17.3.2 Ett vertyg för att interagera med

17.3.2.1 container hanteringsprogram

17.3.2.2 versions hanteringsprogram

17.3.2.3 Executors/platforms

Kan kommunicera med dessa och specificera vilka resurser varje enskil modul kräver.

17.3.2.4

18 Arbetsflödet och programmen som det omfattar är versionskontrollerade

19 Skalbarhet

20 Nextflow arbetsflöden som följer och inte följer nf-core

21 Behov av kontinuerlig utveckling av algoritmer

22 Konfigurera och köra på från kommando linjen

23 Om Seqera igen

24 Konfigurerar och köra på Seqera